

Article

A Building Heat Load Prediction Method Driven by a Multi-Component Fusion LSTM Ridge Regression Ensemble Model

Yu Zhang ^{1,2}  and Guangshu Chen ^{1,*} 

¹ School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China; zhangyuscholar@bucea.edu.cn

² State Key Laboratory in China for GeoMechanics and Deep Underground Engineering, China University of Mining & Technology, Beijing 100083, China

* Correspondence: chenguangshu2021@163.com; Tel.: +86-188-1040-2720

Abstract: Under the background of “double carbon”, building carbon emission reduction is urgent, and improving energy efficiency through short-term building heat load forecasting is an efficient means of building carbon emission reduction. Aiming at the characteristics of the decomposed short-term building heat load data, such as complex trend changes, significant seasonal changes, and randomness, a single-step short-term building heat load prediction method driven by the multi-component fusion LSTM Ridge Regression Ensemble Model (ST-LSTM-RR) is designed and implemented. First, the trend and seasonal components of the heat load are decomposed by the STL seasonal decomposition algorithm, which are fused into the original data to construct three diversified datasets; second, three basic models, namely, the trend LSTM, the seasonal LSTM, and the original LSTM, are trained; and then, the ridge regression model is trained to fuse the predicted values of the three basic models to obtain the final predicted values. Finally, the method of this paper is applied to the heat load prediction of eight groups in a large mountain hotel park, and the root mean square error (RMSE) and mean absolute error (MAE) are used as the evaluation indexes. The experimental results show that the average RMSE and average MAE of the prediction results of the proposed method in this paper are minimized on the eight groups.

Keywords: building heat load prediction; ensemble deep learning; seasonal and trend decomposition using LOESS; long short-term memory neural network; ridge regression



Citation: Zhang, Y.; Chen, G. A Building Heat Load Prediction Method Driven by a Multi-Component Fusion LSTM Ridge Regression Ensemble Model. *Appl. Sci.* **2024**, *14*, 3810. <https://doi.org/10.3390/app14093810>

Academic Editor: Rodolfo Dufo-López

Received: 31 March 2024

Revised: 23 April 2024

Accepted: 24 April 2024

Published: 29 April 2024



Copyright: © 2024 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

On 22 September 2020, President Xi clearly stated at the United Nations General Assembly that “China’s carbon dioxide emissions should peak by 2030 and strive to complete carbon neutrality by 2060”. The building industry, as a major pillar of the economy and industry, is one of the main sources of carbon emissions [1]. The total annual carbon emissions of the building industry in 2022 will be 5.08 billion tons of CO₂, accounting for 50.9% of the national carbon emissions [2]. Carbon reduction in buildings is imminent. In the field of building heating, to reduce carbon emissions, we can not only use renewable energy sources such as solar photovoltaics and wind energy to generate electricity and utilize this electricity to power electric heating systems like ground source heat pumps, but we can also enhance the energy utilization efficiency during the heating process to reduce energy consumption and further decrease carbon emissions. Integrating Internet of Things-enabled systems with big data and artificial intelligence technologies significantly enhances the dynamic response capabilities of heating equipment, improves energy efficiency, implements on-demand heating, and reduces overall energy consumption. For example, in the study by Kent et al. [3], IoT-enabled smart environmental sensors rapidly collect high-resolution temporal and spatial big data (such as temperature and

lighting). This data are then applied to machine learning models for predictive analysis, which in turn reduces energy consumption and enhances thermal comfort. The building heat load can guide the operation and regulation of the heating system, which is an important data basis for ensuring the stable and low-consumption operation of the heating system. If the heat load of a building can be predicted in advance, accurate heating can be realized. Building heat load forecasts can be categorized into ultra-short-term, short-term, medium-term, and long-term load forecasts according to the time granularity. Short-term building heat load forecasting is to forecast the heat load of a building for the next 1 h to 24 h or several days. According to the number of prediction steps, the short-term building heat load prediction can be further divided into single-step prediction and multi-step prediction, where single-step prediction refers to the prediction of the heat load in the next 1 h only, and multi-step prediction refers to the prediction of the heat load of the next multiple hours at one time by the model. In this paper, we study the single-step prediction method for short-term building heat loads. Short-term heat load data are characterized by seasonality, trends, randomness, volatility, and non-stationarity, which leads to an extremely complex pattern of change and a high difficulty in prediction. Therefore, the establishment of an accurate and stable prediction model becomes the key to solving the problem. With the continuous development of various Internet of Things (IoT), big data, and cloud computing technologies, it has become more convenient to obtain and store heating system operation data. This provides rich historical data for the establishment of data-driven models and satisfies the database required for data analysis and model training. Therefore, data-driven models are highly favored in the field of building heat load prediction at a national and international level and are gaining more research and application opportunities.

Data-driven models fall into four main categories: statistical models, machine learning models, deep learning models, and hybrid models [4,5]. Statistical models utilize mathematical and statistical knowledge to construct a model and produce predictions directly from the input data and the model. Because of its simple structure, it has a significant advantage in terms of time overhead [6]. Common statistical models include Autoregressive Moving Averages (ARMA) [7], Autoregressive Composite Moving Average (ARIMA) [8], Exponential smoothing [9], Multiple Nonlinear Regression (MNR) [10], Recursive Least Squares (RLS) [11], and the classic Box–Jenkins methodology models [12]. Although these statistical models can consider the temporal relationship of the data, it is difficult to capture the nonlinear pattern of heat load data and the nonlinear relationship with other relevant influencing factors, which leads to the prediction accuracy being insufficient to meet the actual work requirements. With the progress of computer hardware technology and artificial intelligence technology, artificial intelligence models based on machine learning and deep learning are gradually popularized and applied [13].

The most common machine learning models are Decision Trees (DT) [14], Random Forest (RF) [15], Support Vector Machine (SVM) [16], Extreme Gradient Boost (XGBoost) [17] and so on. These machine learning models can be used for regression prediction and are capable of effectively forecasting the patterns of change in building heat load data. However, it is difficult to take into account the time dependence of data, and feature engineering is more complicated. The deep learning model is an end-to-end model and does not need to carry out complex feature engineering. It can not only predict the nonlinear pattern of heat load data but also consider the time dependence of the data and capture the relationship between different influencing factors and the heat load so as to reduce the impact of the randomness of the heat load.

Among the deep learning models suitable for time-series prediction are Deep Neural Networks (DNNs) [18], Recurrent Neural Networks (RNNs) [19], Long Short-Term Memory Neural Networks (LSTMs) [20,21], Gated Recurrent Units (GRUs) [22], Convolutional Neural Networks (CNNs) [23,24], Transfer Learning [25,26], Reinforcement Learning [27], and so on. LSTM is a variant of RNN that optimizes the issues of vanishing and exploding gradients in RNNs, making it the preferred model for sequence prediction. Li et al. [28]

developed a Bayesian optimization-based LSTM model for short-term heat load forecasting. This model is trained and tested using real-time data from the heat exchange stations in Changchun City, and its predictive performance is verified through multiple evaluation metrics. The results indicate that this model surpasses other methods in terms of prediction accuracy, especially in the 72 h forecast step. Jung et al. [29] proposed a multi-layer GRU model based on an attention mechanism, which performs excellently in multi-step short-term load forecasting, particularly in dealing with long input sequences. It can effectively focus on important variables, thereby improving the model's predictive performance.

However, single deep learning models have certain limitations in terms of prediction accuracy and robustness. Hybrid models, which combine multiple base models, aim to leverage their strengths comprehensively to enhance predictive performance. Hybrid modeling has gradually been widely studied and applied by scholars. Jiawang Sun et al. [30] proposed a novel hybrid deep reinforcement learning integrated optimization model, aimed at effectively predicting the heat load in District Heating Systems (DHS). This model integrates a similar sample selection method, a short-term prediction model pool, and a deep reinforcement learning integration strategy, and it has been validated using a dataset from a heat exchange station in Tianjin. The experimental results show that the model can accurately predict heat load variations, achieving energy savings of 5.33%, 5.31%, and 5.07% across different prediction periods. Pachauri et al. [31] developed a novel regression tree ensemble model in their study, which combines decision trees with the LS-boosting algorithm and is optimized by the SFLA algorithm. This model effectively predicted the heating and cooling loads of residential buildings. The experimental results indicate that the model surpasses other methods in terms of accuracy and efficiency. Moradzadeh et al. [32] proposed a novel hybrid machine learning model called GSVR (Group Support Vector Regression) for predicting the heating and cooling loads of residential buildings. This model combines the GMDH (Group Method of Data Handling) and SVR (Support Vector Regression) models. The study validated the model using datasets of 12 different building shapes simulated in Ecotect software. The experimental results show that the GSVR model can accurately predict heating and cooling loads, demonstrating high correlation coefficients and low statistical errors. Zherui Ma et al. [33] proposed a decomposition-integration prediction model combining VMD (Variational Mode Decomposition) and GRU (Gated Recurrent Unit) to accurately predict building thermal loads in the absence of meteorological parameters. The model operates through four steps: data cleaning, modal decomposition, GRU prediction, and result integration, proving its superiority over traditional models. The main advantage of the model is its ability to make effective predictions without meteorological data.

Through analyzing the current trends in building heat load research, it has been observed that an increasing number of scholars are adopting machine learning, deep learning, various decomposition algorithms, and optimization algorithms for hybrid modeling to forecast short-term building heat loads. However, few researchers start from the data itself to analyze the inherent time series characteristics of heat load data, such as trends, periodicity, seasonality, randomness, volatility, and stationarity. These characteristics add to the complexity of the data. Choosing appropriate data processing methods and forecasting models based on the data's characteristics can further enhance the accuracy of predictions. This article conducts a thorough investigation into this subject. By analyzing the characteristics of the heat load data, it is found to exhibit complex trend changes, significant seasonal variations, and randomness. To accommodate these characteristics, this paper designs and implements a multi-component fusion LSTM Ridge Regression Ensemble Model (ST-LSTM-RR) for the short-term single-step prediction of the building heat load. First, the STL seasonal decomposition algorithm is used to extract the trend and seasonal components of the heat load, which are then integrated into the original data to construct three diverse datasets. Second, the original LSTM is trained for basic prediction, the trend LSTM is employed to capture peak changes in the heat load, and the seasonal LSTM is used to capture cyclic variations. Lastly, a Ridge Regression model is trained to integrate

the predictive values of the three basic models, thereby enhancing the overall model’s prediction accuracy and stability and reducing the impact of randomness on the prediction results. The experimental results show that the proposed method achieves the smallest average RMSE and average MAE across eight groups.

2. Methodology

This article designs and implements a single-step prediction method for a short-term building heat load driven by a Multicomponent Fusion LSTM Ridge Regression Ensemble Model (abbreviated as ST-LSTM-RR). The overall system framework consists of two main parts: data analysis and processing and model training and prediction, with the overall system framework illustrated below in Figure 1.

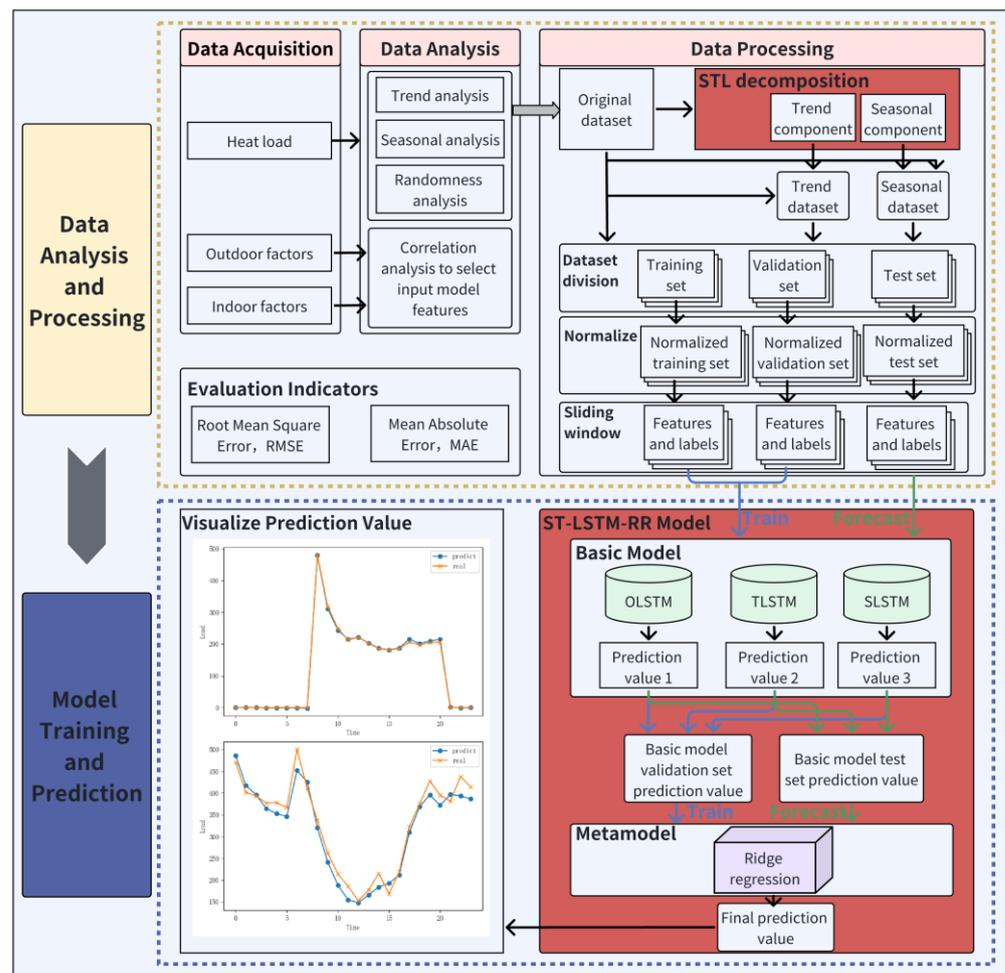


Figure 1. Overall system framework diagram.

The data analysis and processing part includes three major steps: Data Acquisition, Data Analysis, and Data Processing. In Data Acquisition, different features within the data are obtained through various means, including indoor factors, outdoor factors, and the heat load. In Data Analysis, the time series characteristics of the heat load are analyzed, and features for the input model are selected based on Pearson correlation coefficients. Time series analysis allows us to understand the intrinsic patterns of variation in the heat load. Correlation analysis helps identify the correlation coefficients between different features and the heat load, enabling the selection of features that significantly impact heat load variations. In Data Processing, the main task is to process the experimental data into a format suitable for model training and prediction. Notably, using the STL seasonal

decomposition algorithm to decompose the heat load is an important step in constructing the ST-LSTM-RR. This part will be detailed in Section 2.1.

The model training and prediction parts encompass Basic Model Training, Metamodel Selection, and how to use the final ensemble deep learning model for forecasting. The theoretical content of this section will be elaborated on in Sections 2.2 and 2.3, while the experimental part will be described in Section 3.

2.1. Data Analysis and Processing

2.1.1. Data Acquisition and Correlation Analysis

The original data were obtained from eight different building groups within a large mountainous resort hotel complex over a one-year period, with each data point including a time step at 1 h intervals. The basic information of each group is shown in Table 1. The residential group is abbreviated as Rg and the office group is abbreviated as Og. The data features for each group are divided into indoor factors, outdoor factors, and the historical heat load. Outdoor factors include the dry bulb temperature, relative humidity, solar radiation, wind speed, wind direction, atmospheric pressure, and rainfall. Indoor factors consist of the personnel occupancy rate (the ratio of the actual number of people present in a room to the maximum occupancy of that room), housing occupancy rate (the ratio of the number of rented rooms to the total number of rooms in a hotel), indoor set maximum temperature, and indoor set minimum temperature. The historical load is the actual load value for a period prior to the forecasted time step. All feature values of the data were obtained through various means. Outdoor factors were obtained from weather stations; the personnel occupancy rate was obtained through millimeter wave sensors; the room occupancy rate was directly exported from the hotel's business system; the indoor set maximum and minimum temperatures were measured by indoor return air temperature sensors; and the heat load was collected by the heat meter of each group.

Table 1. Basic information of the heating floor.

Group Name	Heating Area (m ²)	Number of Aboveground Floors	Number of Underground Floors
Rg1	11,407	4	/
Rg2	9570	3	1
Rg3	9637	4	/
Rg4	6268	5	1
Rg5	7104	5	1
Rg6	8831	5	2
Og1	5235	2	/
Og2	18,402	4	2

After data acquisition, we need to select indoor and outdoor influencing factors that have a strong correlation with the building heat load as input features for training the model. We use the Pearson correlation coefficient method for the correlation analysis, which measures the strength and direction of the linear relationship between two variables. The degree of correlation between features can be judged based on the absolute value of the coefficient. The formula for calculating the Pearson correlation coefficient is as follows:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

where r represents the Pearson correlation coefficient, x_i and y_i represent the values of the two features for the i -th sample, and \bar{x} and \bar{y} , respectively, represent the means of the two features.

The absolute value of the Pearson correlation coefficient being between 0.8 and 1.0 indicates a very strong correlation, it being between 0.6 and 0.8 indicates a strong correlation, it being between 0.4 and 0.6 indicates a moderate correlation, it being between 0.2 and 0.4 indicates a weak correlation, and it being between 0 and 0.2 indicates a very weak or no correlation. The results of the analysis using the Pearson correlation coefficient method are shown in Figure 2.

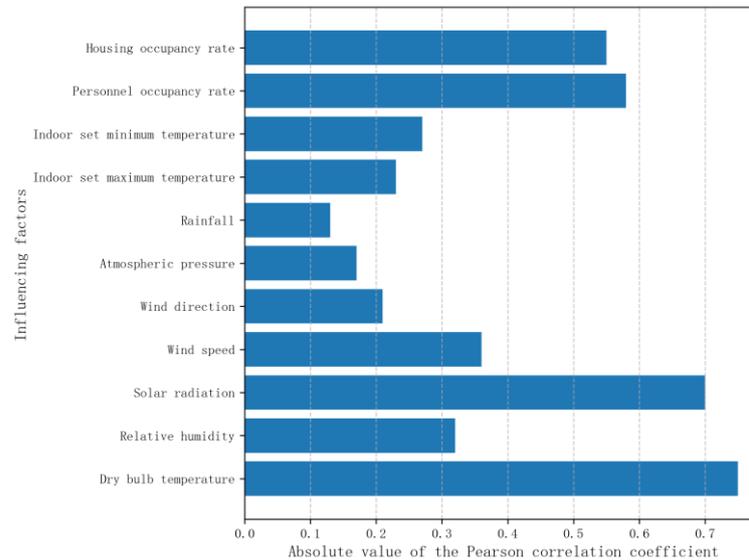


Figure 2. Correlation Analysis Results.

As observed, the absolute values of the correlation coefficients for the dry bulb temperature and solar radiation are the highest, ranging between 0.6 and 0.8, indicating a strong correlation with the heat load. The relative humidity and wind speed have correlation coefficients between 0.3 and 0.4, showing a weak correlation with the heat load. However, since the data are derived from a mountain hotel, these two factors should not be directly disregarded and can be considered as secondary influencing factors. The wind direction, atmospheric pressure, and rainfall have very weak correlations with the heat load and can be ignored. The correlation coefficients for the minimum and maximum indoor set temperatures range from 0.2 to 0.3, indicating a weak correlation. The personnel occupancy rate and housing occupancy rate have correlation coefficients between 0.5 and 0.6, showing a moderate correlation with the heat load.

Ultimately, outdoor weather factors such as the dry bulb temperature and solar radiation are selected as the primary influencing factors, while the relative humidity and wind speed are considered secondary influencing factors. For indoor factors, the hotel's housing occupancy rate and personnel occupancy rate are chosen as the primary influencing factors. These influencing factors, along with the historical heat load data, are then used as input features for the model.

2.1.2. Data Analysis

Short-term building heat load data are among the key datasets reflecting changes in building energy consumption. Essentially, building heat load data are a type of time series data. Observing the heat load distribution changes of the eight groups in Figure 3, it is evident that the patterns of change are extremely complex. Time series analysis can reveal changes in trends, seasonality, randomness, and other characteristics. These analytical results can guide data preprocessing and the construction of predictive model structures, thereby improving model prediction outcomes and providing data support for energy-saving measures. Therefore, analyzing the time series characteristics of short-term building heat load data, such as trends, seasonality, and randomness, plays a crucial role.

1. Trend Analysis

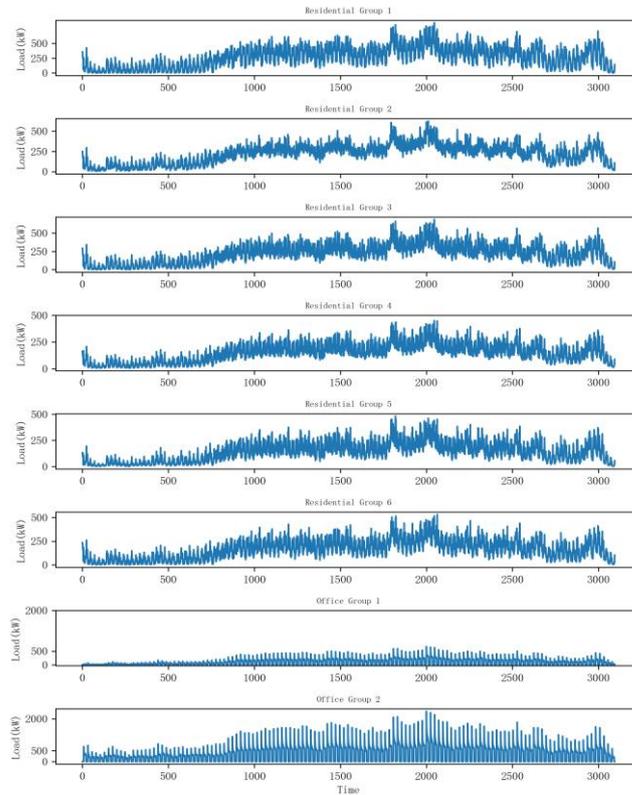


Figure 3. Heat load distribution changes of eight groups.

A trend refers to the direction of change in data over time. Trends can be upward, downward, or stable, and they can be linear or nonlinear, among other characteristics. In this paper, the trends of each group are separated using seasonal decomposition, as shown in Figure 4. Upon observation, it is evident that the trends of the various groups exhibit nonlinear changes, generally rising before falling. This pattern of an increase followed by a decrease is also present in local areas, making the trend change patterns complex.

2. Seasonal Analysis

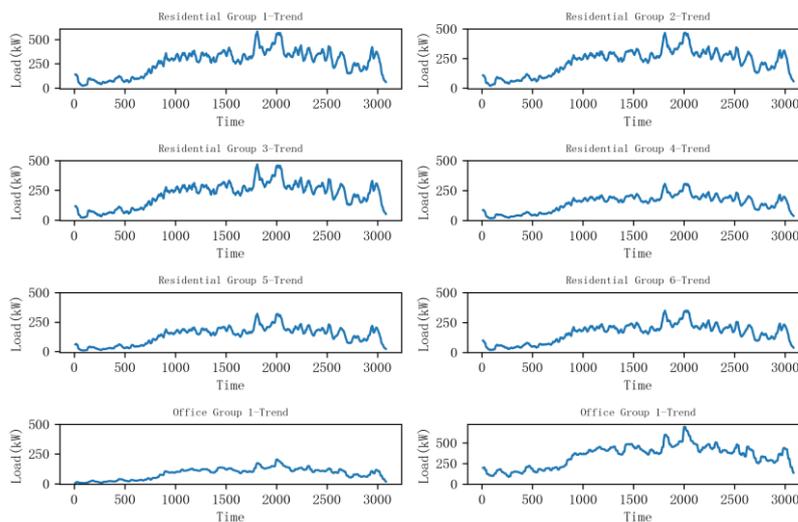


Figure 4. Trend Graphs for Each Group.

Seasonality refers to the fixed changes present in time series data, often related to the weather, holidays, and other seasonal factors. These changes recur annually, monthly, weekly, or daily with relatively stable patterns—for example, an increase in the heating demand during winter, the changing of seasons throughout the year, weekends off, sunrises and sunsets, etc. Methods for identifying seasonality include seasonal decomposition, the seasonal index method, and the time series graphical method, among others. This paper identifies seasonality through seasonal decomposition, with the decomposition results shown in Figure 5. It is evident that each group exhibits a fixed daily heat load change pattern, characterized by a distinct 24 h seasonal variation.

3. Randomness Analysis

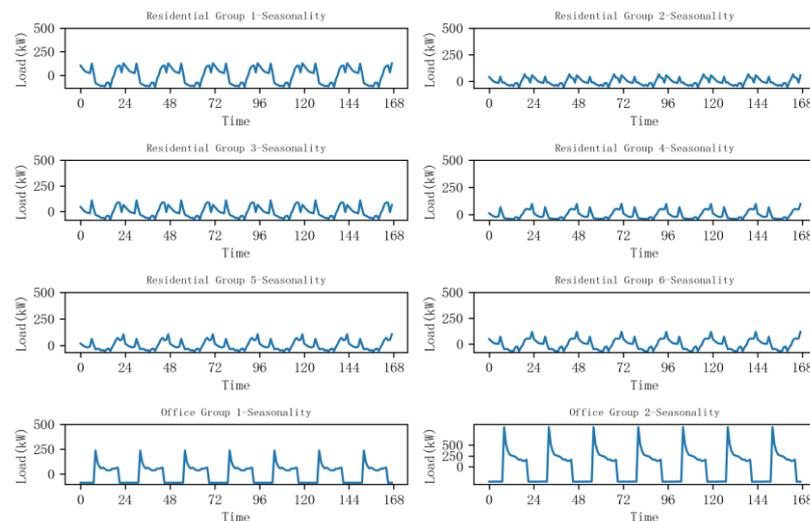


Figure 5. Seasonal Graphs for One Week of the Eight Groups.

Randomness refers to the random fluctuations and uncertainties in the data. The residuals of the heat load data can be decomposed by the seasonal decomposition method, and these residuals are random, which can reflect the randomness of the heat load to a certain extent, as shown in Figure 6. This randomness makes the time series data present irregular and unpredictable characteristics. There are many reasons for the randomness of short-term heat load data, including external factors, internal factors, data acquisition errors, and other unknown factors. We need to analyze the various reasons and find corresponding countermeasures to minimize the influence of randomness.

For external factors, such as weather and climate changes, you can analyze the correlation between the weather data and the heat load, find the weather factors that have a large correlation coefficient with the heat load, and integrate them with the heat load data. For internal factors, such as the structure of the building itself, the use of the building, the activities of the people inside, etc., which also need to be taken into account, some quantifiable factors can be integrated with the heat load data. For the missing or abnormal data caused by data acquisition errors, the data can be supplemented or smoothed by data preprocessing. When choosing to build a prediction model, we should choose to build a model with better stability, and instead of using a single model, we can use a hybrid model to increase the robustness of the model.

In summary, through the analysis of the time series characteristics of heat load data, we found that the building heat load data in this study exhibit complex trend variations, significant 24 h seasonal changes, and random changes, making their patterns of change extremely complex. This requires the models we choose and construct not only to capture the trend and seasonal variations of the heat load data but also to provide stable predictions and mitigate the impacts of random changes, thereby adapting to the complex changes in the heat load data.

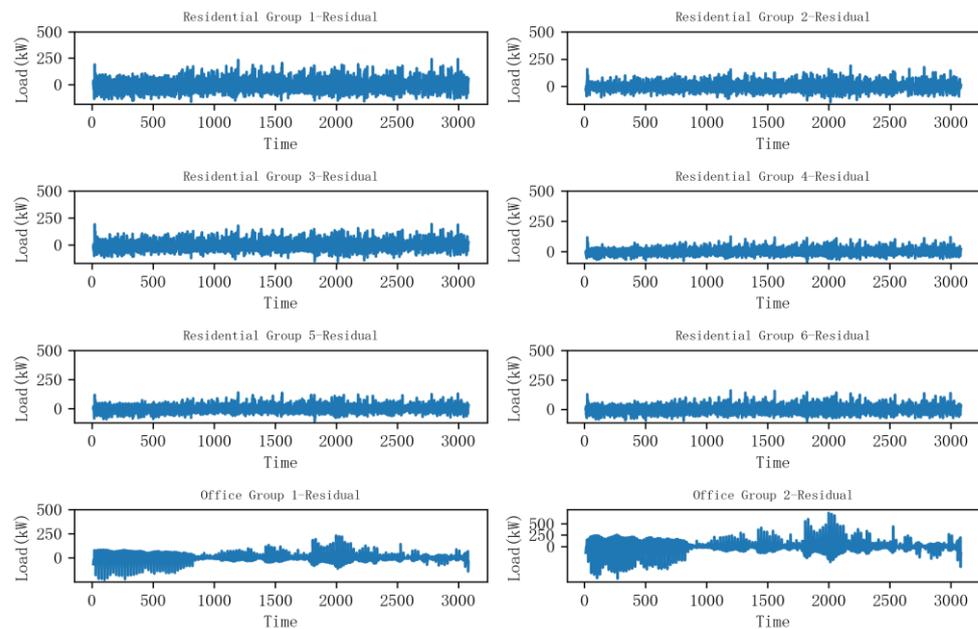


Figure 6. Residuals for each group.

2.1.3. Data Processing

In the data processing section, experimental data need to be subjected to STL seasonal decomposition, dataset division, normalization, and sliding window operations.

1. STL Seasonal Decomposition:

In this step, the original data are decomposed into a trend component, a seasonal component, and a residual component, which is discarded directly because it is difficult to predict the residual component. Then, the trend component and seasonal component are integrated into the original dataset to form the trend dataset and seasonal dataset, respectively.

2. Dataset Division

The data from the eight groups in the park are divided into training, validation, and test sets at a ratio of 6:2:2. The division ratio has a great impact on the training effect and generalization of the model. To ensure the model is evaluated on completely unseen data, thus preventing data leakage, the datasets should be split before data normalization.

3. Normalization

The normalization process stabilizes the scale of the data values, which is particularly beneficial in speeding up the gradient descent during model training. By normalizing the data before prediction, we ensure that the test set and the training set are on a consistent scale. In this paper, we use the maximum–minimum normalization method to scale the original data to between $[0, 1]$, and the calculation formula is as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2)$$

where x is the original data, $\min(x)$ and $\max(x)$ are the minimum and maximum values in each column, and x' is the original data mapped to values in the range $[0, 1]$.

4. Sliding window

The learning method of ST-LSTM-RR is supervised learning; each data sample should consist of input X and output y . The data should be transformed into the form of feature label pairs. Therefore, it is necessary to transform the building heat load data into the form of feature-label pair data for the supervised problem first. In this paper, a sliding window is used to transform the heat load data into feature label pair data.

2.1.4. Evaluation Indicators

Building heat load prediction belongs to regression prediction, and regression prediction cannot be measured by the accuracy rate. The effect of model prediction is generally evaluated by the error assessment. In this paper, the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are used as the error evaluation indexes to quantitatively analyze the prediction effect of the model. The smaller the value of both error indicators, the better the prediction effect is. RMSE is sensitive to outliers and can test the effect of outliers, while MAE is closer to reality. Through the situation of these two error values, the effect of the model prediction can be measured. The specific formula is as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y - y')^2} \quad (3)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y - y'| \quad (4)$$

In the above equation, n denotes the number of predicted samples, y denotes the real value, and y' denotes the predicted value.

2.2. Relevant Basic Theories

2.2.1. Seasonal Decomposition STL Algorithm

The STL algorithm is a locally weighted linear regression (LOESS)-based seasonal decomposition algorithm proposed by Cleveland et al. [34], which uses an additive model to decompose the time series data Y_t into a trend component T_t , a seasonal component S_t , and a residual component R_t .

$$Y_t = T_t + S_t + R_t, \quad t = 1, 2, 3, \dots, n \quad (5)$$

The STL decomposition method is widely used in time series analysis and seasonal adjustment, especially for data with nonlinear trends and seasonal patterns. In the data analysis, this paper identifies that the data from the eight building groups involved in the study exhibit complex trend variations and pronounced 24 h seasonal changes. Therefore, using the STL algorithm will effectively extract the trend and seasonal components of the building heat load. The decomposition principle is divided into two parts, the inner loop and the outer loop; the inner loop is mainly for fitting the trend and calculating the seasonal components, and the outer loop is mainly for calculating and updating the robustness weights of each sample point.

Mark T_t^k, S_t^k as the trend component and seasonal component at the end of the k -1st time of the inner cycle, $T_t^0 = 0$ at the first trial. Denote the number of inner cycles as n_i , the number of outer cycles as n_o , and the period as n_p .

The inner cycle is divided into the following six steps:

1. Detrending. The time series Y_t is de-trended from the last iteration T_t^k to obtain a new $Y_t \leftarrow Y_t - T_t^k$, where, initially, $T_t^0 = 0$.
2. Cyclic subsequence smoothing. Each subsequence in step (1) is processed using LOESS regression, extending one cycle each before and after, with the smoothing parameter n_s , and obtaining a smoothing result denoted as C_t^{k+1} .
3. Low flux filtering of subsequences. The smoothing result C_t^{k+1} in (2) is sequentially subjected to a sliding average of lengths $n_p, n_p, 3$ and then a LOESS regression with parameter n_l is performed to obtain a sequence L_t^{k+1} of length N .
4. Removing seasonal trends. Obtain the seasonal component $S_t^{k+1} = C_t^{k+1} - L_t^{k+1}$.
5. De-seasonalization. The seasonal component of de-seasonalization is $Y_t - S_t^{k+1}$.
6. Trend smoothing. The series obtained in step (5) is subjected to LOESS regression to obtain the trend component T_t^{k+1} . Determine if there is convergence; if there is convergence, output the result; otherwise, return to (1).

The outer loop is mainly used to adjust the robust weight ρ_t in the LOESS regression to minimize the effect of outliers on the regression values.

$$\rho_t = \frac{B(|R_t|)}{h} \tag{6}$$

where h is:

$$h = 6 * \text{median}(|R_t|) \tag{7}$$

$B(x)$ is the weight function, defined as follows:

$$B(x) = \begin{cases} (1 - x^2)^2, & 0 \leq x < 1 \\ 0, & x > 1 \end{cases} \tag{8}$$

2.2.2. Long Short-Term Memory Network

A Long Short-Term Memory Neural Network (LSTM) is a variant of Recurrent Neural Networks (RNNs) designed to process sequential data, addressing the long-term dependency problem found in traditional RNNs. Due to its exceptional ability to capture long-term dependencies within sequential data, it can learn non-linear temporal patterns in data, enabling future data prediction. It introduces three gates: the input gate I_t , forget gate F_t , and output gate O_t , which are defined as follows:

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \tag{9}$$

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f) \tag{10}$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \tag{11}$$

Among them, $W_{xi}, W_{hi}, W_{xf}, W_{hf}, W_{xo}$, and W_{ho} represent the parameter, matrix b_i, b_f, b_o is the bias term, and σ is the Sigmoid activation function. Additionally, a candidate state \tilde{C}_t and a cell state C_t are introduced, defined as follows:

$$\tilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c) \tag{12}$$

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t \tag{13}$$

wherein W_{xc}, W_{hc} represents the parameter matrix, b_c is the bias term, \odot denotes the Hadamard product, indicating element-wise multiplication, and \tanh is the activation function. The cell state C_t stores long-term memory, which can mitigate the vanishing gradient issue. Finally, the hidden state H_t stores short-term memory, defined as follows:

$$H_t = O_t \odot \tanh(C_t) \tag{14}$$

2.2.3. Ridge Regression

Ridge Regression (RR) is an extension of linear regression. The advantage of ridge regression lies in its unbiased estimation, which tends more towards shrinking some coefficients towards zero. This characteristic makes it suitable for dealing with multicollinearity and overfitting problems [35]. In traditional linear regression, a high correlation among independent variables leads to increased variance in parameter estimates. This results in model instability. Ridge regression addresses these issues by adding a regularization term to the loss function, effectively mitigating these problems. The loss function of ridge regression can be expressed as follows:

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \beta_j^2 \\ &= \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \|\beta\|^2 \end{aligned} \tag{15}$$

In the formula, \mathcal{L} denotes the loss function, y_i refers to the actual value of the i th observation, \hat{y}_i indicates the predicted value of the i th observation, β_j represents the regression coefficient for the j th feature, p stands for the number of features, λ is the regularization parameter in ridge regression, which governs the effect of the regularization term on the model, and $\lambda\|\beta\|^2$ signifies the L2 norm of regularization.

In ridge regression, our goal is to minimize the loss function. The regularization term $\lambda\|\beta\|^2$ acts as a penalty on the regression coefficients, constraining their magnitudes and thus reducing model complexity. The choice of the parameter λ is crucial, as it determines the strength of the regularization term. A larger λ results in smaller regression coefficients, thereby lowering model complexity and reducing the risk of overfitting. Ridge regression is typically solved using a variant of the Ordinary Least Squares (OLS) method, known as Ridge Estimation, which has a closed-form solution:

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y \quad (16)$$

In the equation, X represents the design matrix, containing all the independent variables, y is the vector of the target variable, and I is the identity matrix.

2.2.4. Ensemble Deep Learning

Ensemble deep learning aims to better capture complex data distributions and improve model generalization performance by combining the outputs of multiple deep learning models, thereby diminishing the impact of random changes. The success factors of ensemble learning can be categorized into the following aspects [36,37]:

1. Data sample techniques used in training: Enhancing data quality by introducing other data features or using different data sampling techniques, such as bootstrapping and cross-validation, thereby increasing the robustness and generalization performance of the ensemble.
2. Types of basic models: Selecting different types of basic models can increase the diversity of the ensemble and improve overall performance—for example, models like decision trees, support vector machines, and neural networks.
3. Diversity of basic models: Ensuring differences among basic models to avoid overfitting. This can be achieved by adjusting model parameters, selecting different feature subsets, or using different training algorithms to increase model diversity.
4. Methods of combining basic models: This includes voting methods and meta-learning approaches. The main ensemble learning methods include Bagging, Boosting, Stacking, and Ensemble Pruning. This paper employs the Stacking approach, which involves multiple basic models and a meta-model. Through parallel ensemble techniques for generating basic learners, it fully leverages the strengths of different models and obtains the final result through a meta-learning fusion method.

The ensemble deep learning strategy employed in this paper involves training different configurations of the same basic model with a variety of data samples. This approach is unique to ensemble deep learning and allows for the training of the same basic model with different structures on diverse data samples, combining the advantages of different configurations. This method can enhance the predictive performance and robustness of the model.

2.3. Proposed ST-LSTM-RR Ensemble Deep Learning Model

To address the problem of the single-step prediction of a short-term building heat load, this paper proposes and implements a multi-component fusion LSTM Ridge Regression Ensemble Model (hereafter referred to as the ST-LSTM-RR model). This ensemble model is a type of ensemble deep learning model that employs parallel ensemble technology. The integration strategy involves training different structures of the same basic model with datasets containing various features, thereby introducing diversity into the basic model. According to the results of the data analysis, the heat load dataset used in the

study exhibits clear trends, seasonality, and randomness, making the pattern of data changes extremely complex. The proposed ensemble model cleverly incorporates trend and seasonal components to allow the LSTM model to better capture the patterns of heat load changes; by integrating three diverse LSTM models, it further enhances the overall predictive performance and stability of the model, enabling it to counteract the randomness of heat load changes to a certain extent. The ST-LSTM-RR model structure is divided into three main parts, with the specific structure and principles illustrated in Figure 7.

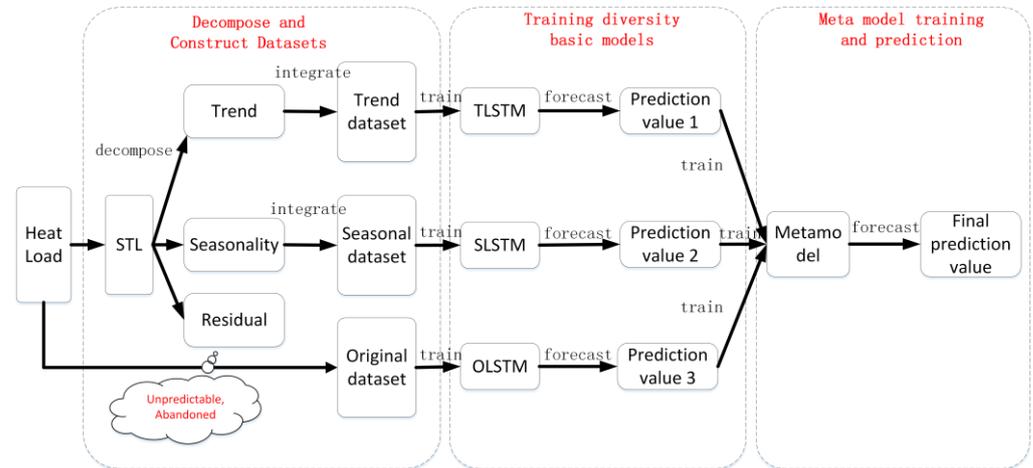


Figure 7. Schematic Diagram of the Ensemble Deep Learning Model Structure.

The first part involves decomposing to construct diverse datasets. Using the additive mode of the STL seasonal decomposition algorithm, the original building heat load values Y are decomposed into three components: the trend T_t , seasonality S_t , and residual R_t . Since the residual component is difficult to predict, it is discarded, and only the seasonal and trend components are added as features to the original data features. This process creates a trend dataset D_T and a seasonal dataset D_S , plus the original dataset D_O , resulting in three diverse datasets. These datasets are independent of each other, with each being used to train a separate basic model.

$$D_T = \{X_1, \dots, X_n, T_t, Y\} \tag{17}$$

$$D_S = \{X_1, \dots, X_n, S_t, Y\} \tag{18}$$

$$D_O = \{X_1, \dots, X_n, Y\} \tag{19}$$

where X_i represents the features, the trend T_t and seasonality S_t are also considered as features, and n is the number of features in the original dataset, excluding the heat load.

The second part involves training diverse basic models. The basic model employed is the LSTM model, which is the preferred model for predicting time series data, and building heat load data also fall into the category of time series data. Parallel training is utilized, employing the trend dataset, seasonal dataset, and original dataset to train the Trend LSTM (abbreviated as TLSTM), Seasonal LSTM (abbreviated as SLSTM), and Original LSTM (abbreviated as OLSTM) models, respectively. These three models are independent of each other, possessing different structures and parameters. The Original LSTM provides the basic prediction, the Trend LSTM is better at predicting changes in peaks, and the Seasonal LSTM excels at capturing cyclical variations. This part ultimately produces three diverse basic models. After training is completed, a validation set (val) is input into the three models to obtain the prediction values P_t , P_s , and P_o , which will be used to train the metamodel.

$$P_S = SLSTM(val) \tag{20}$$

$$P_T = TLSTM(val) \tag{21}$$

$$P_O = OLSTM(val) \quad (22)$$

The third part involves the training and prediction of the meta-model. When training deep learning models, if only a single model is used, it can take a considerable amount of time to adjust parameters to find the optimal model for a given dataset. However, a meta-model, which learns based on the experience of basic models, can integrate the prediction results of multiple basic models, reducing the overall model tuning time and offering better stability.

Since all three basic models are trained based on the LSTM model, their prediction values will exhibit certain multicollinearity issues. The Ridge Regression model (RR) is capable of addressing this issue. Therefore, the ST-LSTM-RR model selects ridge regression as the meta-model to integrate the prediction values of the basic models. The three prediction values are used as features to train the ridge regression meta-model, and the prediction result of the meta-model serves as the final prediction value y' of the ST-LSTM-RR model.

$$y' = RR(P_T, P_S, P_O) \quad (23)$$

3. Results

3.1. Experimental Environment and Parameter Settings

The detailed configuration of the experimental environment for the proposed model is presented in Table 2.

Table 2. Experimental environment configuration.

Experimental Environment		Configuration
Hardware	Operating System	Windows 10
	CPU	Intel(R) Core(TM) i7-9750H
	GPU	NVIDIA GeForce RTX 2060
	Memory	32G
Software	Programming Language	Python 3.7, ipython 7.31.1
	Development Tools	Pycharm 2020, Anaconda 3, Jupyter Notebook 7.3.5
	Deep Learning Framework	Tensorflow 2.1, Keras 1.0.8
	Software Packages	Matplotlib 3.4.3, Numpy 1.19.2, Pandas 1.3.2, Sklearn 0.0, Statsmodels 0.13.0

During the training process of the ensemble model, the STL decomposition algorithm is implemented using the STL function from the Statsmodels package. The basic model uses the ReLU activation function, the Mean Absolute Error (MAE) as the loss function, and Adam as the optimizer, which adaptively adjusts the learning rate. The training involves 300 iterations, with a batch size of 64 for inputting data to the model. The network has a dropout rate of 0.3 and consists of three layers, with neuron counts of 140, 140, and 80, respectively, in each layer. The meta-model employs a ridge regression model, implemented by calling the model from the Sklearn package.

3.2. Basic Model Training

The process of training a model is essentially the search for the optimal hyperparameters, which typically include the learning rate, number of network layers, number of neurons, loss function, optimization algorithm, batch size, activation function, number of iterations, dropout rate, and ratio of dataset division. In the basic LSTM used in this paper, the setting of the time step length is also included. During the process of training the model presented in this paper, it was found that the dataset division ratio and the selection of the time step length have a more significant impact on the performance and generalization ability of the LSTM model. Appropriate choices of the dataset division ratio and time step length can help the model better learn the patterns and rules of the data,

thereby improving the predictive effect and stability of the model. Therefore, detailed experimental explanations for these two hyperparameters are provided.

3.2.1. Dataset Division

Before starting model training, dividing the dataset into training, validation, and test sets is a crucial step. Common division ratios include 8:1:1, 7:1.5:1.5, and 6:2:2. To find the most suitable dataset division ratio for the basic model of the proposed model, experiments were conducted on the datasets of office group one and residential group four. In these experiments, RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) were used as evaluation metrics, and the model's predictive results were recorded, as shown in Table 3. Through observation and analysis of the experimental results, it was found that among all the experimental groups, the smallest error results were obtained when the dataset was divided according to a 6:2:2 ratio. Therefore, the best predictive performance of the model is achieved when the experimental dataset is divided using a 6:2:2 ratio.

Table 3. Experimental results of the dataset division ratio.

Dataset	Division Ratio	RMSE	MAE
Office Group 1	8:1:1	22.108	8.532
	7:1.5:1.5	13.380	8.572
	6:2:2	13.287	6.982
Residential Group 4	8:1:1	34.381	25.224
	7:1.5:1.5	29.730	23.174
	6:2:2	27.232	22.149

3.2.2. Time-Step Selection

In time series forecasting, the time step length for each sample input into the model, that is, how many past timestamps there are whose data are used, is crucial for the model to unearth patterns in the time series. The data from office group one and residential group four are easily predictable and can more clearly demonstrate the impact of the time step length on the model prediction error. Therefore, these two groups' data were selected as benchmarks for conducting time step length experiments. Experiments were carried out with time steps of 72 h, 48 h, 24 h, 12 h, 6 h, 3 h, and 1 h, with specific experimental results presented in Table 4.

Table 4. Experimental results of selecting the time step.

Dataset	Time Step	RMSE	MAE
Office Group 1	1 h	114.624	35.112
	3 h	114.158	35.021
	6 h	106.223	26.599
	12 h	26.273	16.013
	24 h	13.287	6.982
	48 h	6.142	4.105
	72 h	17.809	12.120
Residential Group 4	1 h	49.946	35.363
	3 h	44.805	30.101
	6 h	37.849	27.433
	12 h	33.277	26.455
	24 h	27.232	22.149
	48 h	24.697	20.195
	72 h	35.663	29.567

It is evident that for both office group one and residential group four, the worst predictions occurred with a time step length of 1 h, while the best predictions occurred with a time step length of 48 h. Moreover, as the time step length increased, the error values

of RMSE and MAE progressively decreased, reaching their optimum at 48 h. When the time step length was set to 72 h, the model's prediction results began to decline from their optimum, indicating that although increasing the model's input time step length can reduce the prediction error, too long a time step length can also increase the error. The optimal time step length for this model is 48 h, meaning that when the past two days' data are used as a single input sample for the model, the predictive performance is optimal, effectively utilizing the data's correlations and time dependencies.

3.3. Metamodel Selection

In ensemble deep models, the meta-model is a higher-level model that uses the predictions of basic models as features for training, rather than directly using the original data. By learning from the predictions of the basic models, the meta-model can integrate the strengths of multiple basic models to achieve more accurate and robust predictions, potentially reducing the risk of overfitting and improving the model's generalization ability. Meta models typically choose simpler models, which are easier to understand and interpret, helping to reveal the underlying patterns and mechanisms of the data. To find the most suitable meta-model for the ensemble model proposed in this paper, one to two models were selected from linear models, machine learning models, and neural network models for experimentation. Linear models included the commonly used Ridge Regression and Lasso Regression models; machine learning models included the Support Vector Machine Regression model (SVR) and Random Forest model; and neural network models selected the Deep Neural Network model (DNN). The specific experimental results are presented in Table 5.

Table 5. Metamodel prediction error table.

Dataset	Metamodel	RMSE	MAE
Office Group 1	Ridge Regression	3.809	2.655
	Lasso	6.392	4.905
	Random Forest	6.883	3.012
	SVR	4.317	2.978
	DNN	5.755	3.938
Residential Group 4	Ridge Regression	17.907	14.688
	Lasso	19.940	15.794
	Random Forest	20.775	16.316
	SVR	19.595	14.228
	DNN	19.593	14.981

In office group one, the Ridge Regression model had an RMSE of 3.809 and an MAE of 2.655, performing the best among all meta-models. The performance of SVR was slightly inferior to that of Ridge Regression, while Random Forest and Lasso performed worse. In residential group four, the Ridge Regression model also performed the best among all meta-models, with an RMSE of 17.907 and an MAE of 14.688. SVR and DNN slightly lagged behind Ridge Regression, and Random Forest and Lasso showed poorer performances. In both groups, the Ridge Regression model demonstrated a relatively better performance, with lower RMSE and MAE values, indicating its better generalization ability and stability in the predictive tasks of these two groups. Although Ridge Regression is a linear model, it achieved the best results. This may be because the basic models were all based on LSTM models, which likely presented multicollinearity in their predictions, with high correlations among them. Ridge Regression can effectively handle such multicollinearity, enhancing the predictive performance and stability of the ensemble model.

3.4. Model Comparison Experiment

This section primarily conducts comparative experiments between basic models and the ensemble model, as well as comparisons with other different models, to further demonstrate the superiority of the proposed model.

3.4.1. Comparative Experiments between the Basic Model and Ensemble Model

To demonstrate the effect of incorporating trend and seasonal components for the predictions of the LSTM model, and to show that the three models are diverse yet effective in different ways, we further argue that the ensemble model can integrate the strengths of each basic model. This integration leads to improved prediction accuracy and enhanced stability. A comparison experiment between the basic models and the ensemble model was conducted, with the experimental results shown in Table 6. The best model for each group is indicated in bold. The residential group is abbreviated as Rg and the office group is abbreviated as Og.

Table 6. Comparison between the Basic Model and Ensemble Model.

Error	Model	Rg1	Rg2	Rg3	Rg4	Rg5	Rg6	Og1	Og2
RMSE	OLSTM	53.804	39.272	44.810	24.697	22.045	29.088	6.142	33.878
	TLSTM	40.623	30.734	35.349	18.896	24.564	19.172	6.950	21.329
	SLSTM	50.138	26.680	36.116	22.139	23.056	24.636	4.428	36.639
	Proposed model	40.935	25.533	34.790	17.907	20.843	24.739	3.809	13.397
MAE	OLSTM	42.111	30.632	34.958	20.195	16.919	22.549	4.105	21.672
	TLSTM	29.958	20.310	27.741	15.150	17.173	24.330	4.907	10.883
	SLSTM	41.020	22.047	29.496	17.149	17.497	18.063	3.020	26.009
	Proposed model	30.924	20.638	26.792	14.688	15.058	19.172	2.655	8.328

To highlight the differences between models, the RMSE and MAE values of the model predictions are visualized as bar charts, as seen in Figures 8 and 9.

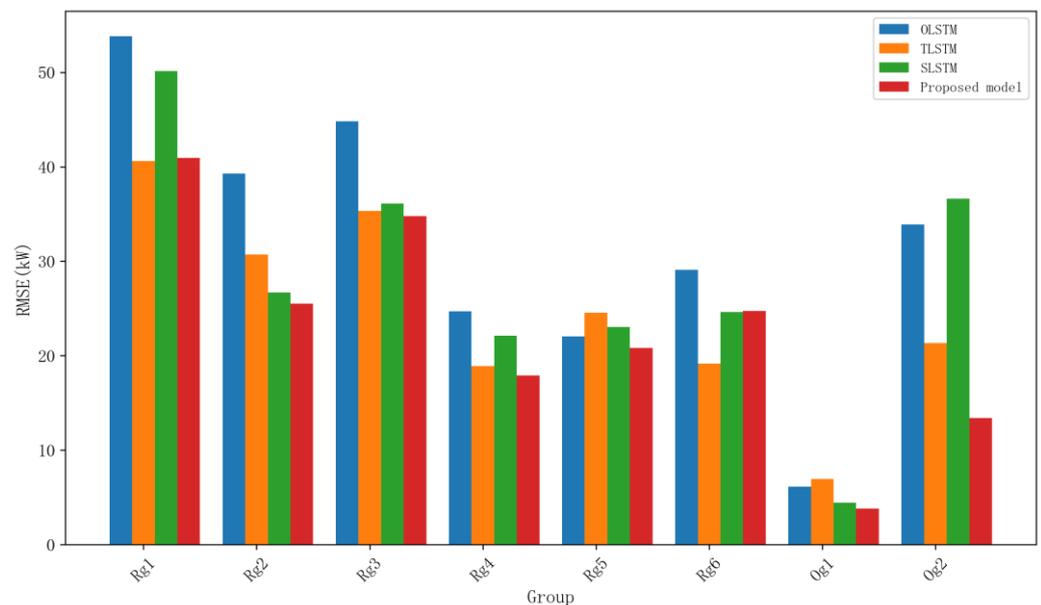


Figure 8. The RMSE of the model on each group.

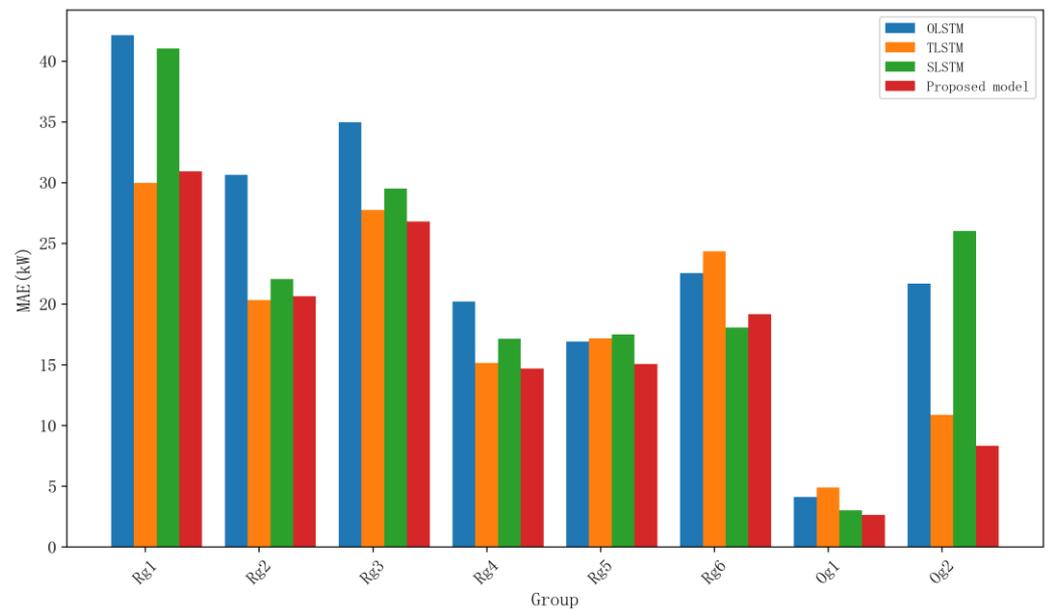


Figure 9. The MAE of the model on each group.

Observing Figures 8 and 9, the Original LSTM model generally performs poorly across most groups, with its predictions exhibiting relatively large errors. However, in residential group five, the Original LSTM's predictions slightly surpass those of LSTM models incorporating trend or seasonal feature components, performing just below the proposed ensemble model. This indicates the necessity of including the Original LSTM model among the basic models.

From the RMSE metric, the Trend LSTM shows the best predictions in residential groups one and six compared to the other models, even outperforming the proposed model in some cases, and it ranks second to the proposed model in residential groups three and four but overall performs better than the Original LSTM model. From the MAE metric, the Trend LSTM achieves the best predictions in residential groups one and two, again outperforming the proposed model in some cases, and it is second only to the proposed model in residential groups three and four and office group two but overall performs better than the Original LSTM model. This demonstrates that incorporating the trend component improves LSTM model predictions, indicating the trend component's influence on LSTM prediction outcomes.

From the RMSE metric, the Seasonal LSTM outperforms the Original LSTM in all groups and exceeds the Trend LSTM in residential groups two and five and office group one, slightly behind the performance of the proposed ensemble model. From the MAE metric, the Seasonal LSTM also surpasses the Original LSTM in all groups, and in residential group six, it even outperforms the proposed ensemble model. This shows that the introduction of seasonal components also impacts LSTM predictions, further reducing the prediction error for the LSTM model.

The proposed ST-LSTM-RR ensemble model exhibits an excellent performance in most groups, particularly in residential groups two, three, four, and five and office groups one and two, achieving the best results. This indicates that the proposed ensemble model can integrate the strengths of each model to achieve more stable predictive outcomes.

In summary, introducing trend or seasonal components can enhance the predictive performance of LSTM models, and the proposed ensemble model offers better predictive outcomes and more stable results.

From Table 6, it is clear that each basic model has its strengths in different groups. To visually verify the diversity of the basic models, line graphs of the predictive results of the three basic models in residential group four and office group one were plotted, as seen in Figures 10 and 11. Figure 10 shows that in the first five days, the predictive results of

the three basic models are quite close and perform very well, but in predicting the turning points on the sixth and seventh days, OLSTM's predictions differ from the actual values by more than 100, indicating a significant error, whereas TLSTM and SLSTM continue to predict the turning points accurately. The data's pattern of change in residential group four is complex. Observing Figure 11, the three basic models demonstrate their strengths at different time points, making better predictions.

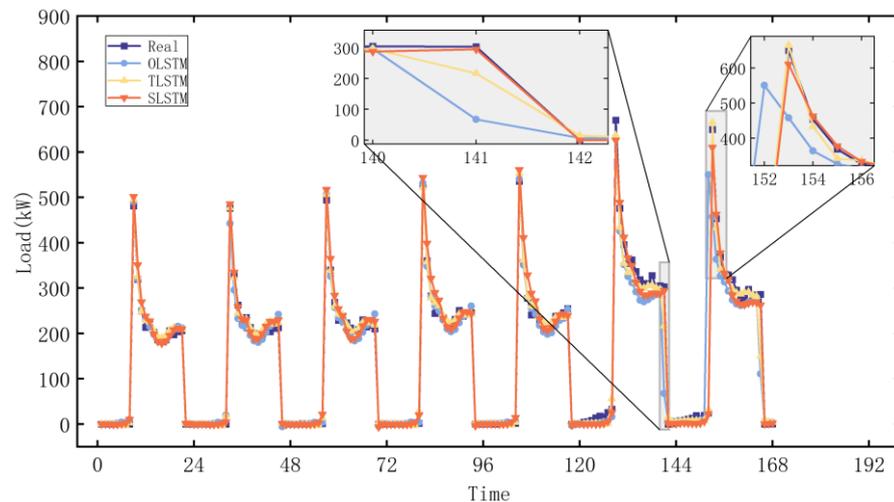


Figure 10. The predictive results of the three basic models in office group one.

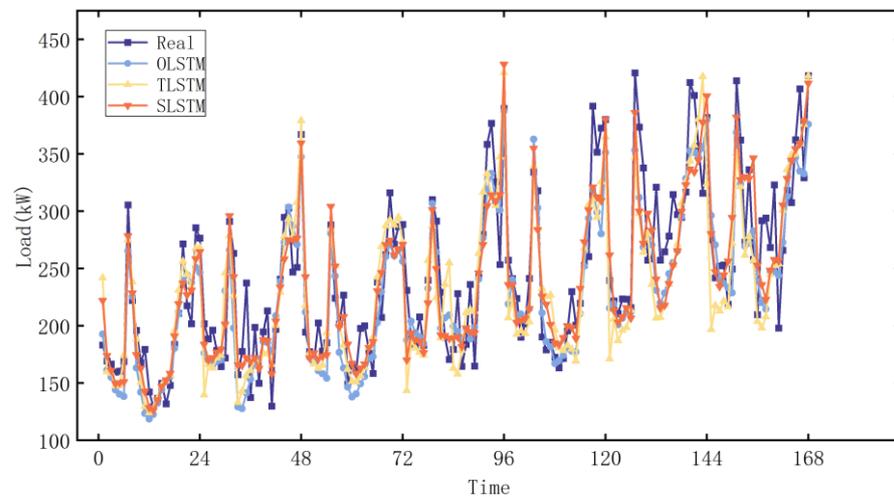


Figure 11. The predictive results of the three basic models in residential group four.

This illustrates the diversity of the three models, capable of making accurate predictions at different times, laying a solid foundation for the success of the ensemble learning model.

3.4.2. Comparative Experiments with Different Models

To verify the superiority of the ensemble model proposed in this paper, it was compared with the TCN (Temporal Convolutional Network) and ConvLSTM (Convolutional LSTM) models through comparative experiments. These two models are extensively used and studied in the field of time series prediction, particularly excelling in modeling sequential data. TCN is renowned for its parallelism and capability in modeling long-term dependencies [38,39]. ConvLSTM combines the strengths of CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory), making it well suited for time series prediction tasks [40,41]. Since the prediction of a building heat load is also a time series

prediction task, these two models can be considered as appropriate benchmark models. The RMSE and MAE values of each model's predictive results across the eight groups are shown in Table 7. The residential group is abbreviated as Rg and the office group is abbreviated as Og.

Table 7. Comparison results of different models.

Error	Model	Rg1	Rg2	Rg3	Rg4	Rg5	Rg6	Og1	Og2
RMSE	ConvLSTM	64.419	48.049	53.009	38.214	35.013	44.749	6.238	21.651
	TCN	58.189	37.044	39.808	27.939	30.347	38.851	15.582	32.418
	Proposed model	40.935	25.533	34.790	17.907	20.843	24.739	3.809	13.397
MAE	ConvLSTM	54.764	41.755	43.825	29.303	26.798	37.101	2.985	9.522
	TCN	45.521	30.418	29.702	23.554	24.415	32.130	12.438	18.090
	Proposed model	30.924	20.638	26.792	14.688	15.058	19.172	2.655	8.328

By observing and analyzing Table 7, it can be seen that the ConvLSTM model performs the worst among all residential groups, but its performance is relatively acceptable in office groups, even better than the TCN model. Particularly in office group one, the ConvLSTM model's performance is slightly inferior to that of the proposed model. In contrast, the TCN model performs better than ConvLSTM in residential groups but still falls short of the proposed model in most groups. The proposed model's predictive results, in terms of RMSE and MAE, are superior to both TCN and ConvLSTM across all eight groups. In residential groups, the model performs best in group four, while in office groups, the best performance is observed in group one.

The average RMSE of the ConvLSTM model is 38.918, with an average MAE of 30.757. For the TCN model, the average RMSE is 35.022, with an average MAE of 27.034. In comparison, the proposed model has an average RMSE of 22.744 and an average MAE of 17.282. This indicates a reduction in the average RMSE and MAE by 35.1% and 36.1%, respectively, compared to TCN, and a reduction by 41.6% and 43.8%, respectively, compared to ConvLSTM. This outcome underscores the significant advantage of the proposed ensemble model in predictive performance. This superiority may stem from the model's ability to effectively integrate the strengths of different models and accurately capture the complex temporal patterns of the heat load.

In summary, the proposed ensemble model demonstrates outstanding performances across various aspects, providing an effective and robust solution for short-term building heat load single-step prediction problems.

4. Conclusions

In this paper, a single-step prediction method for short-term building heat loads, driven by a multi-component fusion LSTM Ridge Regression Ensemble Model (ST-LSTM-RR), is designed and implemented. This method is applied to the single-step prediction of short-term building heat loads for eight groups in a large mountain hotel park. The experimental results show that the introduction of trend and seasonal components has an effect on the LSTM prediction results; the three basic models of the proposed ST-LSTM-RR model are diverse yet effective in different ways; the overall prediction results of the ST-LSTM-RR model are better and more stable than those of the comparison models. The prediction results of the proposed method can be used to guide energy deployment, improve energy efficiency, and then reduce the carbon emissions of buildings.

Author Contributions: Conceptualization, Y.Z. and G.C.; methodology, Y.Z. and G.C.; software, G.C.; validation, Y.Z.; formal analysis, G.C.; investigation, Y.Z. and G.C.; resources, Y.Z.; data curation, Y.Z. and G.C.; writing—original draft preparation, Y.Z. and G.C.; writing—review and editing, Y.Z. and G.C.; visualization, G.C.; supervision, G.C.; project administration, Y.Z. and G.C.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the State Key Laboratory for GeoMechanics and Deep Underground Engineering & Institute for Deep Underground Science and Engineering, grant number XD2021021, and the Postgraduate education and teaching quality improvement project of Beijing University of Civil Engineering and Architecture, grant number J2022003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dasi, H.; Ying, Z.; Ashab, M.F.B. Proposing Hybrid Prediction Approaches with the Integration of Machine Learning Models and Metaheuristic Algorithms to Forecast the Cooling and Heating Load of Buildings. *Energy* **2024**, *291*, 130297. [[CrossRef](#)]
2. Lu, Z.; Wang, P. Research on Carbon Emission Calculation and Emission Reduction Effect of Green Buildings from the Perspective of the Whole Life Cycle. *Environ. Ecol.* **2024**, *6*, 9–16, 25.
3. Kent, M.G.; Huynh, N.K.; Mishra, A.K.; Tartarini, F.; Lipczynska, A.; Li, J.; Sultan, Z.; Goh, E.; Karunagaran, G.; Natarajan, A.; et al. Energy Savings and Thermal Comfort in a Zero Energy Office Building with Fans in Singapore. *Build. Environ.* **2023**, *243*, 110674. [[CrossRef](#)]
4. Barman, M.; Dev Choudhury, N.B.; Sutradhar, S. A Regional Hybrid GOA-SVM Model Based on Similar Day Approach for Short-Term Load Forecasting in Assam, India. *Energy* **2018**, *145*, 710–720. [[CrossRef](#)]
5. Chaganti, R.; Rustam, F.; Daghriri, T.; Díez, I.D.L.T.; Mazón, J.L.V.; Rodríguez, C.L.; Ashraf, I. Building Heating and Cooling Load Prediction Using Ensemble Machine Learning Model. *Sensors* **2022**, *22*, 7692. [[CrossRef](#)] [[PubMed](#)]
6. Yang, R.; Liu, H.; Nikitas, N.; Duan, Z.; Li, Y.; Li, Y. Short-Term Wind Speed Forecasting Using Deep Reinforcement Learning with Improved Multiple Error Correction Approach. *Energy* **2022**, *239*, 122128. [[CrossRef](#)]
7. Yang, Z.; Ce, L.; Lian, L. Electricity Price Forecasting by a Hybrid Model, Combining Wavelet Transform, ARMA and Kernel-Based Extreme Learning Machine Methods. *Appl. Energy* **2017**, *190*, 291–305. [[CrossRef](#)]
8. Jagait, R.K.; Fekri, M.N.; Grolinger, K.; Mir, S. Load Forecasting Under Concept Drift: Online Ensemble Learning with Recurrent Neural Network and ARIMA. *IEEE Access* **2021**, *9*, 98992–99008. [[CrossRef](#)]
9. Gardner, E.S.; Acar, Y. Fitting the Damped Trend Method of Exponential Smoothing. *J. Oper. Res. Soc.* **2019**, *70*, 926–930. [[CrossRef](#)]
10. Fan, C.; Ding, Y. Cooling Load Prediction and Optimal Operation of HVAC Systems Using a Multiple Nonlinear Regression Model. *Energy Build.* **2019**, *197*, 7–17. [[CrossRef](#)]
11. Dawood, N. Short-Term Prediction of Energy Consumption in Demand Response for Blocks of Buildings: DR-BoB Approach. *Buildings* **2019**, *9*, 221. [[CrossRef](#)]
12. Maiti, R.; Biswas, A.; Das, S. Coherent Forecasting for Count Time Series Using Box–Jenkins’s AR(p) Model. *Stat. Neerl.* **2016**, *70*, 123–145. [[CrossRef](#)]
13. Irshad, K.; Zahir, M.H.; Shaik, M.S.; Ali, A. Buildings’ Heating and Cooling Load Prediction for Hot Arid Climates: A Novel Intelligent Data-Driven Approach. *Buildings* **2022**, *12*, 1677. [[CrossRef](#)]
14. Alani, A.Y.; Osunmakinde, I.O. Short-Term Multiple Forecasting of Electric Energy Loads for Sustainable Demand Planning in Smart Grids for Smart Homes. *Sustainability* **2017**, *9*, 1972. [[CrossRef](#)]
15. Jurado, S.; Nebot, À.; Mugica, F.; Avellana, N. Hybrid Methodologies for Electricity Load Forecasting: Entropy-Based Feature Selection with Machine Learning and Soft Computing Techniques. *Energy* **2015**, *86*, 276–291. [[CrossRef](#)]
16. 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](#)]
17. Al-Rakhami, M.; Gumaei, A.; Alsanad, A.; Alamri, A.; Hassan, M.M. An Ensemble Learning Approach for Accurate Energy Load Prediction in Residential Buildings. *IEEE Access* **2019**, *7*, 48328–48338. [[CrossRef](#)]
18. Wang, J.; Chen, X.; Zhang, F.; Chen, F.; Xin, Y. Building Load Forecasting Using Deep Neural Network with Efficient Feature Fusion. *J. Mod. Power Syst. Clean Energy* **2021**, *9*, 160–1691. [[CrossRef](#)]
19. Rahman, A.; Smith, A. Predicting Heating Demand and Sizing a Stratified Thermal Storage Tank Using Deep Learning Algorithms. *Appl. Energy* **2018**, *228*, 108–121. [[CrossRef](#)]
20. Jang, J.; Han, J.; Leigh, S. Prediction of Heating Energy Consumption with Operation Pattern Variables for Non-Residential Buildings Using LSTM Networks. *Energy Build.* **2022**, *255*, 111647. [[CrossRef](#)]
21. Lv, R.; Yuan, Z.; Lei, B.; Zheng, J.; Luo, X. Building Thermal Load Prediction Using Deep Learning Method Considering Time-Shifting Correlation in Feature Variables. *J. Build. Eng.* **2022**, *61*, 105316. [[CrossRef](#)]
22. Xue, G.; Song, J.; Kong, X.; Pan, Y.; Qi, C.; Li, H. Prediction of Natural Gas Consumption for City-Level DHS Based on Attention GRU: A Case Study for a Northern Chinese City. *IEEE Access* **2019**, *7*, 130685–130699. [[CrossRef](#)]

23. Khan, N.; Haq, I.U.; Khan, S.U.; Rho, S.; Lee, M.Y.; Baik, S.W. DB-Net: A Novel Dilated CNN Based Multi-Step Forecasting Model for Power Consumption in Integrated Local Energy Systems. *Int. J. Electr. Power Energy Syst.* **2021**, *133*, 107023. [[CrossRef](#)]
24. Zhao, A.; Mi, L.; Xue, X.; Xi, J.; Jiao, Y. Heating Load Prediction of Residential District Using Hybrid Model Based on CNN. *Energy Build.* **2022**, *266*, 112122. [[CrossRef](#)]
25. Lu, Y.; Tian, Z.; Zhou, R.; Liu, W. A General Transfer Learning-Based Framework for Thermal Load Prediction in Regional Energy System. *Energy* **2021**, *217*, 119322. [[CrossRef](#)]
26. Wang, C.; Yuan, J.; Huang, K.; Zhang, J.; Zheng, L.; Zhou, Z.; Zhang, Y. Research on Thermal Load Prediction of District Heating Station Based on Transfer Learning. *Energy* **2022**, *239*, 122309. [[CrossRef](#)]
27. Liu, T.; Tan, Z.; Xu, C.; Chen, H.; Li, Z. Study on Deep Reinforcement Learning Techniques for Building Energy Consumption Forecasting. *Energy Build.* **2020**, *208*, 109675. [[CrossRef](#)]
28. Li, B.; Shao, Y.; Lian, Y.; Li, P.; Lei, Q. Bayesian Optimization-Based LSTM for Short-Term Heating Load Forecasting. *Energies* **2023**, *16*, 6234. [[CrossRef](#)]
29. Jung, S.; Moon, J.; Park, S.; Hwang, E. An Attention-Based Multilayer GRU Model for Multistep-Ahead Short-Term Load Forecasting. *Sensors* **2021**, *21*, 1639. [[CrossRef](#)]
30. Sun, J.; Gong, M.; Zhao, Y.; Han, C.; Jing, L.; Yang, P. A Hybrid Deep Reinforcement Learning Ensemble Optimization Model for Heat Load Energy-Saving Prediction. *J. Build. Eng.* **2022**, *58*, 105031. [[CrossRef](#)]
31. Pachauri, N.; Ahn, C.W. Regression Tree Ensemble Learning-Based Prediction of the Heating and Cooling Loads of Residential Buildings. *Build. Simul.* **2022**, *15*, 2003–2017. [[CrossRef](#)]
32. Moradzadeh, A.; Mohammadi-Ivatloo, B.; Abapour, M.; Anvari-Moghaddam, A.; Roy, S.S. Heating and Cooling Loads Forecasting for Residential Buildings Based on Hybrid Machine Learning Applications: A Comprehensive Review and Comparative Analysis. *IEEE Access* **2022**, *10*, 2196–22151. [[CrossRef](#)]
33. Ma, Z.; Wang, J.; Dong, F.; Wang, R.; Deng, H.; Feng, Y. A Decomposition-Ensemble Prediction Method of Building Thermal Load with Enhanced Electrical Load Information. *J. Build. Eng.* **2022**, *61*, 105330. [[CrossRef](#)]
34. Cleveland, R.B.; Cleveland, W.S.; McRae, J.E. Terpenning, Irma STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *J. Off. Stat.* **1990**, *6*, 3–73.
35. Fan, G.-F.; Liu, Y.-R.; Wei, H.-Z.; Yu, M.; Li, Y.-H. The New Hybrid Approaches to Forecasting Short-Term Electricity Load. *Electr. Power Syst. Res.* **2022**, *213*, 108759. [[CrossRef](#)]
36. Mohammed, A.; Kora, R. A Comprehensive Review on Ensemble Deep Learning: Opportunities and Challenges. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 757–774. [[CrossRef](#)]
37. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble Deep Learning: A Review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [[CrossRef](#)]
38. Song, J.; Xue, G.; Pan, X.; Ma, Y.; Li, H. Hourly Heat Load Prediction Model Based on Temporal Convolutional Neural Network. *IEEE Access* **2020**, *8*, 16726–16741. [[CrossRef](#)]
39. He, N.; Qian, C.; Liu, L.; Cheng, F. Air Conditioning Load Prediction Based on Hybrid Data Decomposition and Non-Parametric Fusion Model. *J. Build. Eng.* **2023**, *80*, 108095. [[CrossRef](#)]
40. Ullah, F.U.M.; Ullah, A.; Khan, N.; Lee, M.Y.; Rho, S.; Baik, S.W. Deep Learning-Assisted Short-Term Power Load Forecasting Using Deep Convolutional LSTM and Stacked GRU. *Complexity* **2022**, *2022*, 2993184. [[CrossRef](#)]
41. Khan, S.U.; Khan, N.; Ullah, F.U.M.; Kim, M.J.; Lee, M.Y.; Baik, S.W. Towards Intelligent Building Energy Management: AI-Based Framework for Power Consumption and Generation Forecasting. *Energy Build.* **2023**, *279*, 112705. [[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.