applied sciences

MDPI

Article

A Group Resident Daily Load Forecasting Method Fusing Self-Attention Mechanism Based on Load Clustering

Jie Cao [1], Ru-Xuan Zhang [1], Chao-Qiang Liu [1], Yuan-Bo Yang [1] and Chin-Ling Chen [2,3,*]

[1] School of Computer Science, Northeast Electric Power University, Jilin City 132012, China
[2] School of Information Engineering, Chuangchun Sci-Tech University, Changchun 130012, China
[3] Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan
* Correspondence: clc@mail.cyut.edu.tw

Abstract: Daily load forecasting is the basis of the economic and safe operation of a power grid. Accurate prediction results can improve the matching of microgrid energy storage capacity allocation. With the popularization of smart meters, the interaction between residential electricity demand and sources and networks is increasing, and massive data are generated at the same time. Previous forecasting methods suffer from poor targeting and high noise. They cannot make full use of the important information of the load data. This paper proposes a new framework for daily load forecasting of group residents. Firstly, we use the singular value decomposition to address the problem of high dimensions of residential electricity data. Meanwhile, we apply a K-Shape-based group residential load clustering method to obtain the typical residential load data. Secondly, we introduce an empirical mode decomposition method to address the problem of high noise of residential load data. Finally, we propose a Bi-LSTM-Attention model for residential daily load forecasting. This method can make full use of the contextual information and the important information of the daily load of group residents. The experiments conducted on a real data set of a power grid show that our method achieves excellent improvements on five prediction error indicators, such as MAPE, which are significantly smaller than the compared baseline methods.

Keywords: daily load forecasting; group resident clustering; Bi-LSTM; self-attention mechanism

check for updates

Citation: Cao, J.; Zhang, R.-X.; Liu, C.-Q.; Yang, Y.-B.; Chen, C.-L. A Group Resident Daily Load Forecasting Method Fusing Self-Attention Mechanism Based on Load Clustering. *Appl. Sci.* **2023**, *13*, 1165. https://doi.org/10.3390/app13021165

Academic Editor: Luis Hernández-Callejo

Received: 13 December 2022
Revised: 12 January 2023
Accepted: 13 January 2023
Published: 15 January 2023

1. Introduction

1.1. Background

Stable, uninterrupted, and high-quality electricity helps to keep industry and society running. Considering that electricity cannot be stored, accurate forecasting of electricity load has a significant impact on the reliability of a power system and economic development. In particular, daily load forecasting plays a vital role in the daily operational management of power companies, such as energy transfer scheduling, unit combination, and load dispatch [1,2]. As an integral part of the daily operation and management of power companies, the accurate prediction of residents' daily load is of great significance to urban power grid planning and power market operation. Overestimation will increase operating costs, and underestimation will lead to power shortages [3,4]. With more and more interactive adjustments between residents' electricity demand, power supply, and power grid, the information interaction between large-scale users and a power grid generates massive data [5,6]. Residential load data have the characteristics of fine granularity, strong volatility, significant difference, and large data volume. This brings new challenges to grid load forecasting [7].

In traditional resident load forecasting, scholars at home and abroad have put forward some methods, which can be divided into two types. One is the time series method, and the other is the artificial intelligence method. Traditional time series methods include

Autoregressive Integrated Moving Averages, Seasonal Autoregressive Integrated Moving Averages, and Vector Autoregressive Moving Averages [8]. ARIMA and SARIMA are suitable for univariate load forecasting [9,10]. VARMA is suitable for multivariate time series forecasting, and it requires a high stability of the series [11]. Traditional time series models are considered linear models, while the load forecasting problem is nonlinear. Recently, more and more researchers prefer to use artificial intelligence methods on load forecasting problems.

Artificial intelligence prediction methods include support vector machines, artificial neural networks, etc. However, existing microgrid daily load forecasting research mainly focuses on the power generation side, for example, on the prediction of thermal power and solar lamp power generation equipment. There is insufficient research on short-term load forecasting on the electricity side. Research on residential load forecasting often focuses on residential load forecasting for individual households, and there is a lack of research on group residential load forecasting. For example, one study [12] used wavelet transform and support vector machines to predict the time series of residential loads. However, the above methods are for load forecasting of individual residents. The authors did not take into account the situation of the population of the group. Another study [13] proposed a linear regression short-term load forecasting model considering the influence of working time and meteorological factors, but its error is relatively large. The main reason is that it lacks the ability to handle sharp fluctuations in the load profile, and noisy data have a large negative impact on load forecasting. Another reason is that it lacks the treatment of the strong fluctuation of the load curve. Undecomposed load noise is large, and the direct prediction error is large. Reference [14] integrates empirical mode decomposition, particle swarm optimization, and an adaptive network-based fuzzy inference system. This method decomposes the load and reduces the noise to a certain extent, but it does not introduce an attention mechanism based on the neural network. As a result, it obtains the same prediction feature weights and can still be further improved.

To summarize, the previous methods have three problems in the daily load forecasting of group residents. Firstly, the load data of group residents have the characteristics of high dimensionality and large amount of data. Existing research tends to predict the load of individual residents, and there is a lack of research on the data characteristics of group residents' load. Secondly, there is a lack of treatment for the fluctuation of the load curve. The traditional method is fixed in the noise reduction time base function, and there is no adaptive matching signal which leads to noisy forecast data. Thirdly, the time series information is assigned the same weight, and the important information of the load data is not fully utilized in the prediction. Therefore, to solve the above problems, we propose a group resident daily load forecasting method. This method successfully reduces the forecast error and realizes the different daily load forecasts of group residents.

The structure of this paper is as follows: Section 2 introduces the theoretical basis of the daily load forecasting method. Section 3 introduces the experimental analysis and experimental results, and Section 4 is the conclusion.

1.2. Related Works

Accurate load prediction is very important for the energy generation, energy dispatching, and smooth operation of a power grid. It can also promote the maximum utilization of renewable resources and reduce the loss of primary energy in the power grid [15]. The user-side power grid, with residential quarters and commercial residences as the main components, is an effective carrier for local consumption and utilization of renewable energy. It undertakes the function of coordinating distributed power and user load [16]. According to existing statistics, the urban user-side power grid load reached more than 60% of the total non-industrial load of a city, becoming an important part of the urban power system [17]. In the power grid environment, due to the difference in consumption patterns among residents, the load prediction of group residents is more challenging than that of substations.

Hippert et al. [18] classified electrical load based on forecast range into short-term load forecast, medium-term load forecast, and long-term load forecast. Short-term load forecasting models range from a few minutes to a few days. Daily load forecasts fall within the scope of short-term load forecasting. Short-term load forecasting is often used in a power grid to bridge the gap between energy generation and demand.

Different from traditional residential load forecasting, load forecasting methods for a single building or a small number of buildings have challenges when dealing with the large number of users collected by smart meters. For example, cluster-based load forecasting methods have attracted more and more attention. In Ref. [19], a K-Means algorithm is applied to specific features of the load profile. They used the average consumption over a day, the average consumption for each day of the week, and the location of the peak over the year as feature clusters. The final sum of predictions is derived from the deep learning model. They improved the predictive accuracy of Irish datasets and smart meters in New York by 11%. In Ref. [20], a load prediction model based on improved fuzzy c-mean clustering is proposed. It filters out the weakly correlated features from adjacent load values and uses similar local daily data as input features. The above methods did not study th load forecasting of the group residents. The problems of large dimension of load data and obvious difference in load curve are not taken into account.

Although clustering is a good solution, it can be optimized before making predictions. Because of the nonlinearity and nonstationarity in residential load data, it is difficult to describe the moving tendency of electric load and to improve the forecast accuracy. To establish a suitable and effective forecasting model, the original data features of the residential short-term electric load need to be fully considered and analyzed. In Refs. [21,22], both papers propose a method for short-term electrical load prediction combining wavelet transforms and neural networks. The authors combine the wavelet transforms with neural networks for short-term electrical load forecasting. Despite the wavelet transforms becoming a standard for the analysis of nonlinear and nonstationary signals, there is still the problem of high prediction noise for computing and failure.

In terms of predictive models, cutting-edge artificial intelligence methods represented by deep learning are developing rapidly. For example, one study [23] proposes an improved deep learning method that improves the accuracy and generalization ability of an infrared small object detection problem. In contrast to shallow learning, deep learning generally refers to stacking multiple layers of neural networks and relying on stochastic optimization to perform learning tasks. Among these methods, the convolutional neural network (CNN) is a type of feedforward neural network that includes convolutional calculations and has a deep structure [24,25]. A recurrent neural network (RNN) is an artificial neural network in which nodes are directed to form a ring. The internal state of such a network can exhibit dynamic timing behavior. In theory, it can use historical information of any length, so it can model time series more completely [26]. However, RNN has the problem of gradient disappearance and gradient explosion during training. Therefore, researchers have improved it and proposed the Long Short-Term Memory (LSTM) network [27]. One study [28] proposes a deep Long Short-Term Memory (LSTM)-based ultra-short-term prediction method for regional-level loads based on big data resources. It visually demonstrates the extraction of abstract features from the load data by using a deep learning algorithm and confirms its good feature learning capability. Reference [29] predicted daily peak load based on Bi-directional Long Short-Term Memory (Bi-LSTM) and feature correlation analysis. Another study [30] proposes an LSTM-based ultra-short-term prediction method for regional-level loads based on big data resources. It visually demonstrates the extraction of abstract features from the load data by using a deep learning algorithm and confirms its good feature learning capability. However, the authors did not take into account the issue of the weighting of adjacent individual load values on the prediction results. This results in the hidden layer missing such critical information, which ultimately reduces the prediction accuracy.

This paper addresses the method of daily load forecasting for groups of residents by first clustering similar load curves. The prediction models in each cluster are then fitted. Finally, the predictions from each cluster are summed to obtain the final predicted value.

2. Methodology

This paper comprehensively considers the problems of residents' load differentiation, high noise, and the same weight. We propose a group resident daily load forecasting method fusing a self-attention mechanism based on load clustering.

In the first step of the method in this paper, the missing values are first filled in the original residential electricity data set, and then the time granularity is set. In the second step, this paper proposes to use the singular value decomposition (SVD) method to reduce the dimensionality of the residential load data to solve the problem of slow processing of massive power consumption data. Then, we use the K-Shape method to cluster the residential electricity consumption data and conduct differential analysis to solve the difference in residential electricity consumption. In the third step, the clustered typical resident load is decomposed by using the superior signal decomposition ability of the modal decomposition method to obtain the IMF component data set. This overcomes the problem of strong data volatility and reduces noise. In the fourth step, the mixed forecasting method is selected, and a Bi-LSTM-Attention (BLA) load forecasting model considering the time series feature weight is constructed to obtain accurate forecasting results. The Bi-LSTM model has the advantage of obtaining contextual information about the time series data, and it can take into account the influence of different time dimensions in the input sequence on the load. The introduction of the attention mechanism enables different weights to be assigned to the load features at different times, which can make full use of the key information of the hidden layer. In the fifth step, the component predictions are summed to obtain the final prediction. The overall research idea of this method is shown in Figure 1.

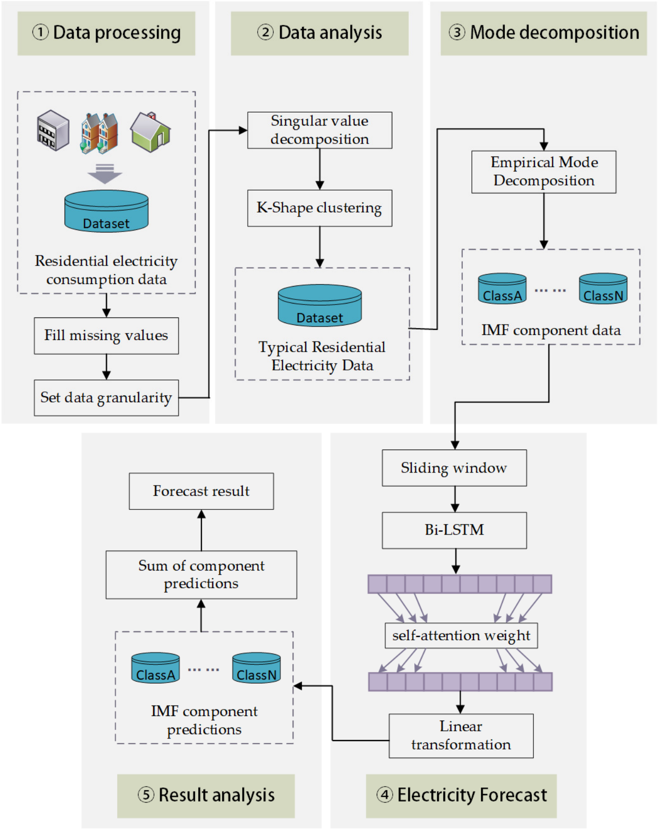


Figure 1. The framework of the daily load forecasting method.

2.1. Clustering Method Based on Dimensionality Reduction

The model proposed in this paper is based on the Bi-LSTM recurrent neural network, which is highly robust for modeling time series data. On this basis, we introduce the attention mechanism to assign different weights to the temporal features and the external features at the same time. This highlights the key features within the input data that play a key role in the residential electricity usage forecasting process. It helps to improve the accuracy of short-term load forecasting.

2.1.1. Dimensionality Reduction Method Based on SVD

Singular value decomposition (SVD) is an important matrix decomposition in linear algebra, it is based on the generalization of arbitrary matrices [31]. Singular value decomposition is similar in some respects to eigenvector-based diagonalization of symmetric or Hermite matrices [32].

Given a matrix A of size m × m, it is decomposed diagonally, such as in Equation (1):

$$A = U\Lambda U_{-1} \tag{1}$$

In this equation, each column of U is an eigenvector, and the elements on the Λ diagonal are the eigenvalues arranged from large to small. If U is recorded as $U = \left(\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m\right)$, then the results are calculated as shown in Equation (2):

$$\begin{aligned} AU = A\left(\vec{u}_1, \vec{u}_2, \quad \ldots, \vec{u}_m\right) &= \left(\lambda_1 \vec{u}_1, \lambda_2 \vec{u}_2, \ldots, \lambda_m \vec{u}_m\right) \\ &= \left(\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m\right) \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_m \end{bmatrix} \\ &\Rightarrow AU = U\Lambda \Rightarrow A = U\Lambda U_{-1} \end{aligned} \tag{2}$$

when matrix A is a symmetric matrix, and it is decomposed symmetrically and diagonally, as shown in Equation (3):

$$A = Q\Lambda Q^T \tag{3}$$

In this equation, each column of Q is a mutually orthogonal eigenvector and is a unit vector, and the elements on the Λ diagonal are eigenvalues arranged from large to small.

When the matrix Q is written as $Q = \left(\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_m\right)$, then the matrix A can also be written as shown in Equation (4):

$$A = \lambda_1 \vec{q}_1 \vec{q}_1^{\,T} + \lambda_2 \vec{q}_2 \vec{q}_2^{\,T} + \ldots \lambda_m \vec{q}_m^{\,T} \tag{4}$$

The goal of the singular value decomposition method is to find a mapping matrix, and then multiply it with the original data A, to achieve the effect of dimensionality reduction, which is suitable for the massive residential electricity numerical data collected by smart meters. Dimensionality reduction of data through singular value decomposition not only reduces the data dimension and accelerates model training, but it also effectively filters noise data and improves model generalization.

2.1.2. Clustering Method Based on K-Shape

Clustering is an unsupervised method that can divide data sets into several groups based on the similarity and distance between the data without prior information [33]. We divide the residents into different clusters according to the annual load curve and established a separate model for each cluster. Existing clustering algorithms mainly include hierarchical clustering, partition-based clustering, density-based clustering, grid-based clustering, and model-based clustering.

K-Shape is one of the most advanced time series clustering algorithms based on K-Means [34]. To cope with time series, K-Shape uses shape-based distance to evaluate the similarity between two curves. In addition, shape-based distance uses cross-correlation distance to identify the minimum distance between two curves, even if they are not properly aligned. Specifically, the K-Shape method is shown in the following steps:

Step 1: K-Shape calculates the distance between two-time series by using the cross-correlation method. We suppose there are two-time series X and Y, both of length m. To achieve translational invariance, Y is set as the constant, and we delimit X step by step and calculate the difference between X and Y at each step. This is shown in the formula below:

$$\vec{x_i}(s) = \begin{cases} \left(\overbrace{0, \ldots, 0}^{|s|}, x_1, x_2, \ldots, x_{m-s} \right), & s \geq 0 \\ \left(x_{1-s}, \ldots, x_{m-1}, x_m, \underbrace{0, \ldots, 0}_{|s|} \right), & s < 0 \end{cases} \tag{5}$$

Step 2: The difference depends on the number of cross-relations, which represents the number of K-Shape cross-relations $CC_\omega\left(\vec{x}, \vec{y}\right) = R_{\omega-m}\left(\vec{x}, \vec{y}\right)$, including $\omega \in \{1, 2, \ldots, 2m-1\}$. $R_{\omega-m}\left(\vec{x}, \vec{y}\right)$ can be divided into two cases, as shown in Equation (6):

$$R_{\omega-m}\left(\vec{x}, \vec{y}\right) = \begin{cases} \sum_{l=1}^{m-k} x_l + k \cdot y_l, & \omega - m \geq 0 \\ R_{\omega-m}\left(\vec{y}, \vec{x}\right), & \omega - m < 0 \end{cases} \tag{6}$$

$R_{\omega-m}$ is used to calculate the similarity between X and Y at each step. The dot product is calculated at the positions present in both X and Y. The final R is the sum of the dot products of the valid regions. It can be said that the larger R is, the more similar the two sequences are.

Step 3: K-Shape defines the shape-based distance (*SBD*). The more the blocks overlap, the bigger the shape is like *CC*. To compare the similarity values of all possible positions, we take the most similar Max (*CC*) and then use $1 - max(CC)$ to obtain *SBD*, as shown in Equation (7):

$$SBD\left(\vec{x}, \vec{y}\right) = 1 - \max_{\omega}\left(\frac{CC_\omega\left(\vec{x}, \vec{y}\right)}{\sqrt{R_0\left(\vec{x}, \vec{x}\right) \cdot R_0\left(\vec{y}, \vec{y}\right)}}\right) \tag{7}$$

Step 4: The more similar the shape, the smaller the distance *SBD* is. The normalized *NCC* value is between [−1, 1] and, therefore, the *SBD* value is between [0, 2]. The *NCC* is calculated as shown in Equation (8):

$$NCC_q\left(\vec{x}, \vec{y}\right) = \begin{cases} \frac{CC_\omega\left(\vec{x}, \vec{y}\right)}{m}, & q = \text{“}b\text{''}(NCC_b) \\ \frac{CC_\omega\left(\vec{x}, \vec{y}\right)}{m - |\omega|}, & q = \text{“}u\text{''}(NCC_u) \\ \frac{CC_\omega\left(\vec{x}, \vec{y}\right)}{\sqrt{R_0\left(\vec{x}, \vec{x}\right) \cdot R_0\left(\vec{y}, \vec{y}\right)}}, & q = \text{“}c\text{''}(NCC_c) \end{cases} \tag{8}$$

Step 5: Once the distance is defined, you need to adjust the centroid algorithm based on the distance logic. By looking for $\vec{\mu_k}^*$, K-Shape makes the similarity between $\vec{\mu_k}^*$ and each sequence x_i of P_k as large as possible. $\vec{\mu_k}^*$ is calculated as shown in Equation (9):

$$\begin{aligned}\vec{\mu_k}^* &= \underset{\vec{\mu_k}}{argmax} \sum_{\vec{x_i}\in P_k} NCC_c\left(\vec{x_i}, \vec{\mu_k}\right)^2 \\ &= \underset{\vec{\mu_k}}{argmax} \sum_{\vec{x_i}\in P_k} \left(\frac{max_\omega CC_\omega\left(\vec{x_i},\vec{\mu_k}\right)^2}{\sqrt{R_0\left(\vec{x_i},\vec{x_i}\right)\cdot R_0\left(\vec{\mu_k},\vec{\mu_k}\right)}}\right)^2\end{aligned} \tag{9}$$

The final clustering method is realized through iteration, and each iteration is divided into two steps. The first step is to recalculate the centroid. The second step is to redistribute each sequence to different clusters according to the distance between each sequence and the new centroid. We keep iterating until the tag stops changing.

In the daily load prediction of power grid group residents, the K-Shape method is used to deal with the load curve of group residents and cluster it. This method can balance the large difference in electricity consumption habits among residents and reduce the heavy workload of forecasting individual residents. K-Shape relies on an extensible iterative refinement process that creates homogeneous and well-separated clusters. As its distance measure, K-Shape uses a normalized version of the cross-correlation measure to consider the shape of the time series when comparing them. K-Shape is a domain-independent, highly accurate, and efficient clustering method, and it is very suitable for time series.

2.2. *Noise Reduction Method Based on Empirical Mode Decomposition*

After the differential analysis of residents, we can obtain the typical residential electricity consumption data. However, the data for each type of residential electricity consumption are strong and noisy. To reduce the noise and further improve the prediction accuracy, we use the empirical mode decomposition method to decompose the complex original signal. This method can decompose complex and unstable load signal into a relatively stable Intrinsic Mode Function (IMF), which is reconstructed into a more stable time series. By using the empirical mode decomposition method, the IMF component is selectively input into the subsequent prediction model to reduce noise and further improve prediction accuracy. The specific process is shown in Figure 2.

Empirical mode decomposition (EMD) is a time–frequency analytical method of signal proposed in 1998, where the signal refers to a time series signal [35]. Common time series signal processing methods can be divided into three categories: time domain, frequency domain, and time–frequency domain [36]. Time-domain analytical features include meaning, variance, kurtosis, peak-to-peak value, etc. Frequency-domain features include frequency, energy, etc. Time–frequency domain analysis includes wavelet transform, etc.

The EMD theory holds that all signals are composed of a finite number of Intrinsic Mode Functions. The IMF component contains local characteristic signals of different time scales of the original signal. The empirical mode decomposition method can make the non-stationary data smooth, and then it performs the Hilbert transform to obtain the time spectrum diagram and the frequency with physical meaning [37]. EMD decomposes the input signal into several eigenmode functions and a residual, which is composed of the following formula:

$$I(n) = \sum_{m=1}^{M} IMF_m(n) + Res_M(n)I(n) \tag{10}$$

where I(n) represents the input signal, $IMF_m(n)$ represents the intrinsic mode function of M_{th}, and $Res_M\ (n)$ represents the residual. Where in the process of extracting IMF is called screening, and the process of screening is as follows.

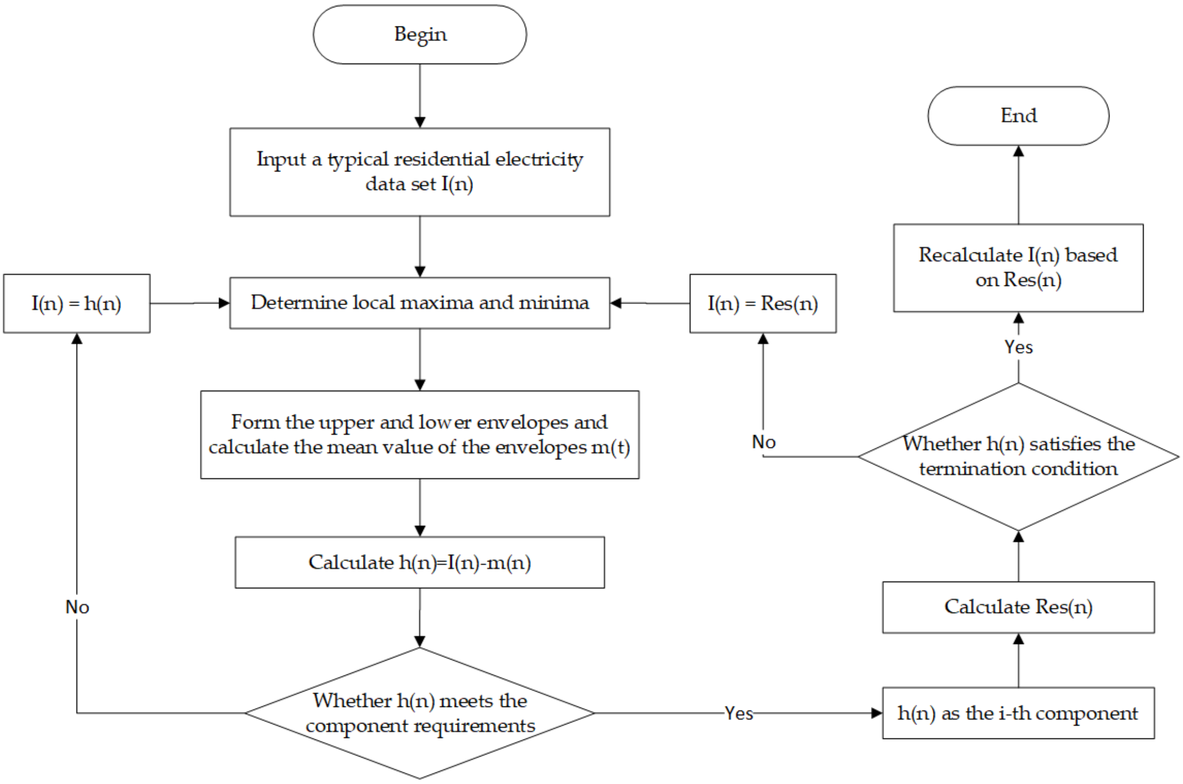

Begin
Input a typical residential electricity data set I(n)
I(n) = h(n)
Determine local maxima and minima
I(n) = Res(n)
Form the upper and lower envelopes and calculate the mean value of the envelopes m(t)
Calculate h(n)=I(n)-m(n)
Whether h(n) meets the component requirements
No
Yes
h(n) as the i-th component
Calculate Res(n)
Whether h(n) satisfies the termination condition
No
Yes
Recalculate I(n) based on Res(n)
End

Figure 2. The structure of the EMD.

First, we mark the local extreme points. Then, we connect the maximum points to form the upper envelope through a cubic spline line and connect the minimum points to form the lower envelope. After that, we find the mean value m_1 of the upper and lower envelope. Finally, we subtract the mean value m_1 of the upper and lower envelopes from the input signal $X(t)$.

$$I(n) - m_1 = h_1 I(n) - m_1 \tag{11}$$

One iteration of the above process cannot guarantee that h_1 is an IMF, and the above process needs to be repeated until h_1 is an IMF. The iterative stopping criterion produces the number of executions of an intrinsic modulus function screening process, and the stopping criterion standard deviation (SD) used in this method is shown in Equation (12):

$$\mathrm{SD_k} = \sum_{n=0}^{T} \frac{\left|h_{k-1}(n) - h_k(n)^2\right|}{h_{k-1}^2(n)} \tag{12}$$

In summary, EMD is atime–frequency domain signal processing method that can decompose signals based on the time-scale characteristics of the data without presetting any basic functions. EMD has obvious advantages in dealing with non-stationary and nonlinear data. It is suitable for analyzing nonlinear and non-stationary signal sequences.

This is because the basis functions of EMD are derived from the signal itself. Therefore, this analysis is adaptive compared to traditional methods where the basis functions are fixed. EMD is based on the sequential extraction of energy associated with various intrinsic time scales of the signal, starting from finer time scales (high-frequency modes) to coarser time scales (low-frequency modes). The sum of IMFs is well-matched to the signal, thus reducing noise while maintaining integrity.

2.3. Bi-LSTM-Attention Model for Residential Daily Load Forecasting

To identify the hidden information in the time series for subsequent prediction, this paper introduces a self-attention mechanism based on Bi-LSTM to assign different weights to the time series data. The Bi-LSTM is used to capture the contextual information, and the important information of the time series data can be fully used by using the self-attention mechanism.

2.3.1. Bi-LSTM Method

LSTM is proposed by Hochreiter, and it is a special RNN that can solve the problem of gradient explosion and gradient disappearance by adding a gate control mechanism [38]. To selectively update memory units, LSTM introduces the unit state c_t to preserve the long-term memory based on the hidden layer state h_t. This reflects the dependencies of adjacent times learned by the deep network at any time step and the institutional characteristics of the input data long before.

Each LSTM calculation unit contains three control gates, an input gate i_t, an output gate o_t, and a forgetting gate f_t. When the input sequence is $\{x_1, x_{2,}, \ldots, x_T\}$, including $x_t \in \left\{x_{t,1}, x_{t,2}, \ldots, x_{t,k}\right\}, \in R^k$ represents the k-dimensional real vector data under the t-time step. The internal updating process of the unit is as follows:

The forgetting gate f_t is proposed to forget the state of the upper memory cell c_{t-1} information. It can be expressed as shown in Equation (13):

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{13}$$

The W_f is the weight matrix of the forgetting gate. b_f is the offset of the forgetting gate. x_t is the current sample input. h_{t-1} is the output of the previous sequence. σ, represent the sigmod function. The input gate i_t and the memory cell candidate status $\widetilde{c_t}$ are calculated as shown in Equations (14) and (15):

$$\widetilde{c_t} = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{14}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{15}$$

where W_i represents the weight matrix of the input gate; W_c represents the weight matrix of the candidate state; and b_i and b_c are the corresponding offset. By combining the last-moment memory state c_{t-1} and the current moment candidate's memory state $\widetilde{c_t}$, i_t and f_t update the current moment memory unit state c_t, as shown in Equation (16):

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \widetilde{c_t} \tag{16}$$

where $\cdot$ represents multiplication by element. The input gate o_t is mainly used to control the output of the memory unit state value. The calculation of o_t is shown in Equation (17):

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_0) \tag{17}$$

where W_o is the weight matrix of the output gate, and o_t. b_0 is the offset of the output gate. The hidden layer output value h_t is obtained by using nonlinear calculation, as shown in Equation (14):

$$h_t = o_t \cdot tanh(c_t) \tag{18}$$

The unit weight and bias of each control cell in the above formula are used to predict the load in the time series through training and learning. Usually, the LSTM network information is a one-way transmission, and it cannot use future information. To adapt to the various characteristics of daily load amplitude, this paper selects Bi-LSTM to construct the prediction model. Bi-LSTM is formed by the combination of forward and backward LSTM, and the structure is shown in Figure 3.

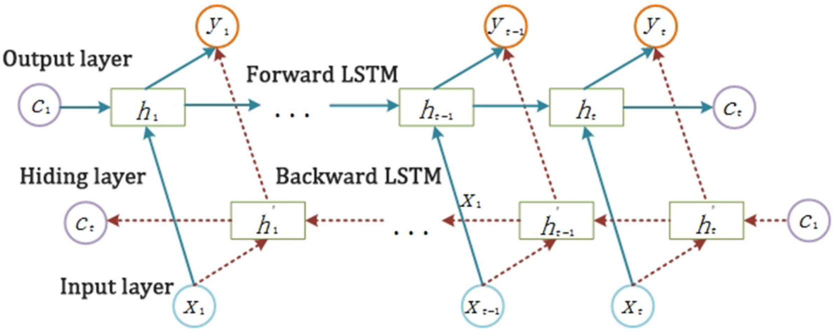


Figure 3. The structure of the Bi-LSTM model.

Forward LSTM can obtain the past data information of the input sequence. Backward LSTM can obtain the future data information of the input sequence. The forward and backward LSTM training of time series is realized to further improve the overall integrity of feature extraction. In t time step, the output H_t of the hidden layer of Bi-LSTM consists of the forward $\overrightarrow{h_t}$ and the backward $\overleftarrow{h_t}$:

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(h_{t-1}, x_t, c_{t-1}), t \in [1, T] \tag{19}$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(h_{t+1}, x_t, c_{t+1}), t \in [T, 1] \tag{20}$$

$$H_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{21}$$

2.3.2. Self-Attention Mechanism

The attention mechanism is a probability-weighted mechanism that mimics the attention of the human brain [39]. When the human brain observes things, it will focus on specific places and ignore other places. There is an intrinsic correlation between hidden features. Hidden features of different time steps have different effects on the prediction results, which is unrecognizable by the Bi-LSTM network. Therefore, the self-attention mechanism is well suited to load prediction methods involving LSTM networks. This method highlights the more important factors by assigning different probability weights to the inputs, ultimately further improving the prediction accuracy of the model.

The attention layer assigns the feature weight of the model learning to the input vector of the next time step to highlight the impact of key features on the sequence. The final data are entered into the fully connected layer. After the virtual function processing of the fully connected layer, the predicted load value is obtained. The implementation process is as follows:

Step 1 Calculate the correlation between each current input feature and the current load.

Step 2 Use the Softmax formula to convert each correlation into a probabilistic form.

Step 3 Multiply each obtained probability by the implicit representation of the corresponding input feature to represent the contribution of the feature to the predicted load. The contributions of all the input features are then added together as the input parts to predict the next load data.

The process can be expressed by Equations (22)~(24):

$$e_t = V tanh(W h_t + b) \tag{22}$$

$$\alpha_t = \frac{exp(e_t)}{\sum_{j=1}^{n} exp(e_j)} \tag{23}$$

$$C_t = \sum_{t=1}^{n} a_t h_t \tag{24}$$

where e_t and α_t are, respectively, the weight score and the attention weight corresponding to different features at the current time t; V and W is the weight of the multilayer perceptron when calculating the attention weight; b is the bias parameter of the multilayer perceptron when calculating the attention weight; N is the dimension of the input vector of the prediction model; and C_t is the output of the attention mechanism at time t. This paper introduces attention mechanisms based on past and future input characteristics. Thus, the model can give different weights to the input features to highlight the influence of strong correlates and reduce the shadow of weakly correlated factors.

As seen in Figure 4, after receiving the input for a time window, the model passes the sequence data to the forward LSTM hidden layer and the reverse LSTM hidden layer, and the two combine to output the processed vector. The attention layer takes the data processed by the LSTM layer as input, calculates the weight vector, and then combines the weight vector with the shallow output to obtain a new vector input into the linear transformation layer. The green boxes in the figure indicate the resulting attention weight parameters. Finally, the linear transformation layer calculates the predicted value.

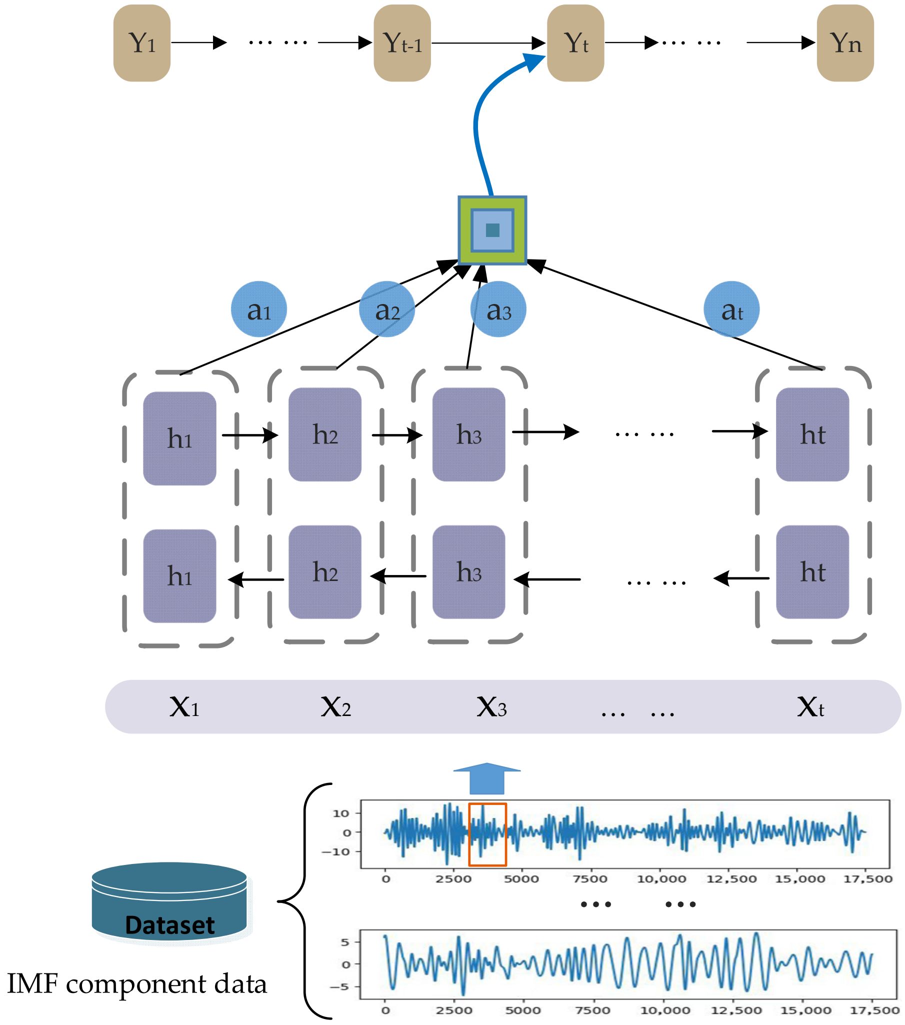


Figure 4. The structure of the Bi-LSTM-Attention Model.

This paper combines the Bi-LSTM algorithm with an attention mechanism. This method mines the contribution of the contextual information in the time series data, the historical resident data in different time dimensions, and the external features to predict the results. Thus, it highlights the effective features and further improves the accuracy of residential electricity forecasting.

3. Example Analysis

3.1. Data Sources

This paper is based on the British Grid's residential electricity consumption data set for London in 2014. The dataset contains partial consumption readings from November 2011 to February 2014 for a sample of 5567 London households, with readings taken every half hour in kilowatt-hours. In this paper, the electricity consumption data of households with relatively complete data from January to December 2012 are selected for research. Each household has 1753 load data pieces, and there are 3533 households, with a total of 6,193,349 load data pieces.

3.2. Error Indicator

The missing data can be discussed in the following two situations: For a single missing data, according to the characteristics of the smooth variation in the power load curve, the average load of the two data points before and after the data point is taken to fill in the missing data. For continuous multiple missing data, according to the characteristic that the load curves of the power system have roughly the same trend, we use similar load curves in adjacent dates to replace them.

We use five predictors to demonstrate the validity of the method proposed in this paper. The formulae for the forecast error assessment indicators are shown below.

Mean Absolute Percentage Error (*MAPE*):

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{P_{ture} - P_{fore}}{p_{ture}}\right| \tag{25}$$

Mean Absolute Error (*MAE*):

$$MAE = \frac{1}{N}\sum_{i=1}^{n}\left|P_{ture} - P_{fore}\right| \tag{26}$$

Mean Square Error (*MSE*):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(P_{ture} - P_{fore}\right)^2 \tag{27}$$

Symmetric Mean Absolute Percentage Error (*SMAPE*):

$$SMAPE = \frac{100\%}{N}\sum_{i=1}^{N}\frac{\left|P_{ture} - P_{fore}\right|}{\left(\left|P_{ture}\right| + \left|P_{fore}\right|\right)} \tag{28}$$

Root Mean Square Error (*RMSE*):

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(P_{true} - P_{fore}\right)^2} \tag{29}$$

In the above formulae, N is the predicted point; P_{ture} is the actual value of the resident load; and P_{fore} is the predicted resident load.

3.3. Experimental Analysis and Verification

3.3.1. Clustering Visualization and Comparison Experiment

We used SVD to downscale the load data of the 3533 households from 17,500 to 2000. The time required for clustering was significantly reduced from seven days to less than one day. We used the K-Shape algorithm to cluster the 3533 real residential user load curves into two categories. To display the data distribution after clustering more vividly, we used Principal Component Analysis to map the data from high-dimensional space to two-dimensional space. The clustering visualization diagram is shown in Figure 5.

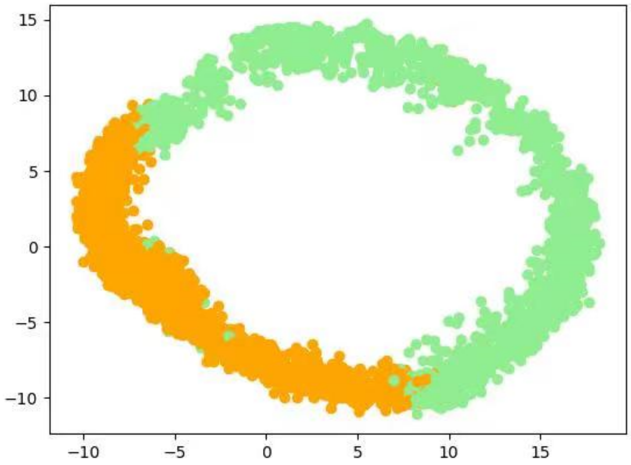


Figure 5. Cluster visualization.

The x-axis and y-axis, respectively, represent the first dimension and the second dimension after the data are reduced to two dimensions, and the difference between different clusters is obvious after clustering. To further demonstrate the clustering effect, the average daily load of residents in different clusters after clustering is shown in Figure 3. The vertical axis represents the load in kW. The horizontal axis represents time, with a minimum time unit of 30 min and a total of 48-time steps.

As can be seen in Figure 6, the K-Shape clustering algorithm can divide residents into two categories according to the difference in residential electricity consumption. The first is high-electricity residents, and the other is low-electricity residents. Among them, the peak daily load of Class I residents is 0.33 kw/h, which is about 33% higher than that of Class II residents at 0.22 kW/h. The daily load valley value of Class I residents is 0.18 kW/h and that of Class II residents is 0.13 kW/h, while the daily trough value of Class I residents is about 38% higher than that of Class II residents, with obvious differences.

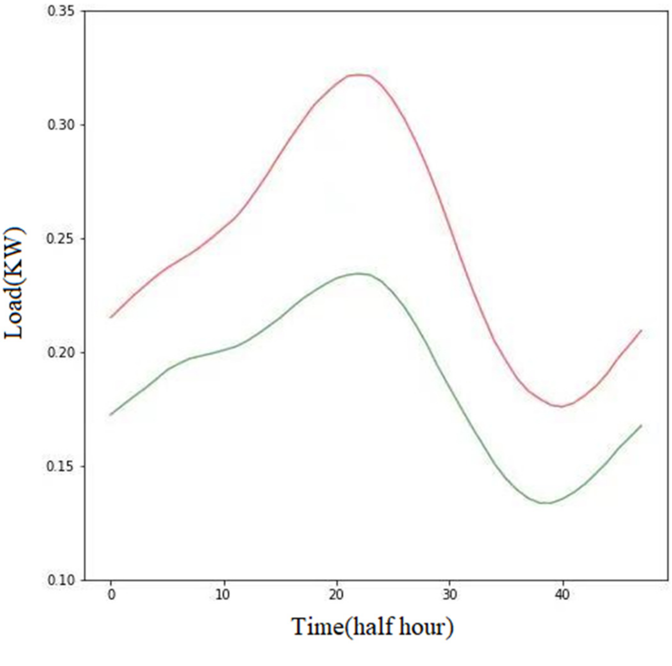


Figure 6. Daily average electricity consumption of two types of residents after clustering.

To further verify the effect of differentiated clustering on subsequent experiments, ablation experiments were performed on the above results. In this project, residents are divided into two categories, high-electricity residents and low-electricity residents, and the daily load is forecasted on the same grid residential electricity consumption dataset. In Table 1, ALL represents the results of the direct prediction without clustering, and K-Shape_2 indicates the results of the prediction after clustering into two categories.

Table 1. Cluster ablation experimental results.

Algorithm	MAPE	MAE	MSE	RMSE	SMAPE
ALL	2.16%	46.36	3177	56.36	3.29
K-Shape_2	1.49%	23.77	2994	59.67	2.84

As shown in Table 1, when the clustering is two classes, the MAPE is 1.49%, and the MAE is 23.77, both of which are better than the unclustered results. This is because we divide the load curves of group residents into two categories through clustering, which are high power consumption curves and low power consumption curves. Predicting them separately can ensure that the input vectors of the prediction model have a higher similarity so that the prediction is more accurate.

As shown in Table 2, K-Shape has a significant improvement compared to K-means and DBSAN. Compared to K-means and DBSAN, the method in this paper improves the MAPE by 5.82% and 14.47%, respectively. Improvements of 47.22 and 107.52 are achieved on the MAE, respectively. Compared to the other two methods, K-Shape significantly improves the results of all the metrics. This is mainly due to the optimization of the distance calculation method, the centroid calculation method, and the introduction of the frequency-domain feature extraction method.

Table 2. Cluster comparison experimental results.

Algorithm	MAPE	MAE	MSE	RMSE	SMAPE
Dbscan	16.24%	122.07	20,496	143.17	15.62
k-means	7.59%	61.77	10,699	103.44	10.16
K-Shape	1.49%	23.77	2994	59.67	2.84

Overall, the prediction performance after clustering is better than that without clustering. Through clustering, the load sequences with the highest similarity in the samples can be screened out and grouped as subsequent prediction training samples. We selected the historical load sequence with the highest similarity to the input vector. It can be guaranteed that the output of the predictive model is closer to the true value, and the input space of the predictive model can be mapped more reasonably.

3.3.2. Noise Reduction Visualization and Comparison Experiment

After preprocessing the input data used in the experiments in this paper, the EMD method was performed on the typical residential electricity consumption data clustered into two categories to obtain eight IMF components. Each component of the IMF represents each frequency component in the original signal. The components are arranged in order from high frequency to low frequency. Modal decomposition is a preprocessing method for signal feature extraction, and each IMF component is used as the input of the subsequent analysis, which is often used to remove noise. Thetime–frequency decomposition diagrams and corresponding spectrum diagrams are shown in Figure 7.

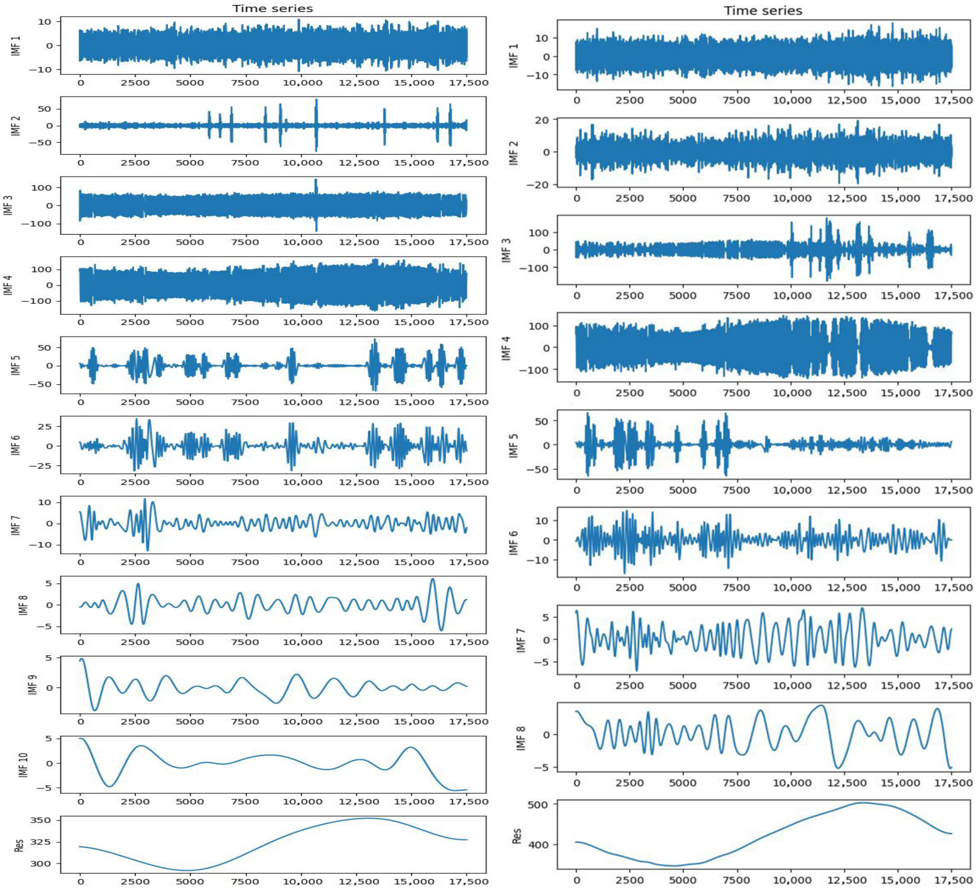


Figure 7. Mode decomposition spectrogram.

To verify the effectiveness of the modal decomposition, all components and the first five components were used as the input data, and experiments were carried out on the power grid residential electricity data set used in this project. The results are shown in Table 3, where EMD_all means using the whole fraction as the input, and EMD_5 means using the first five components as the input.

Table 3. Mode decomposition ablation experimental results.

Model	**MAPE**	**MAE**	**MSE**	**RMSE**	**SMAPE**
EMD_all	3.01%	24.32	1862	43.15	3.82
EMD_5	2.50%	20.46	3130	55.95	6.06

By reducing the IMF component as the input, noise is reduced and the indicators, such as MAPE, are reduced. MAPE decreases from 3.01% to 2.50%, which is a decrease of 0.51%. MAE decreases from 24.32 to 20.46, which is a decrease of 3.86. This shows that the EMD method can remove noise and reduce experimental errors.

3.3.3. Daily Load Forecasting Results from Visualization and Comparison Experiment

Based on clustering, this experiment builds a Bi-LSTM-Attention (BLA) residential electricity consumption prediction model. This model use sliding windows and adds up the electricity consumption of each category. each window size is one week (7 × 48). It uses the window data to predict the next day's 24 h (48 o'clock) residential electricity consumption data. The ratio of the training set and the test set is 9:1, the training set has 1577 time points, and the test set has 175 time points. The training accuracy is 0.000001, the maximum number of iterations is 1000, the learning rate is 0.1, and the maximum number of failures is 10 times. The clustering results based on K-Shape clustering into two, three, five, and eight categories are predicted, and the daily load prediction results are shown in Figure 8.

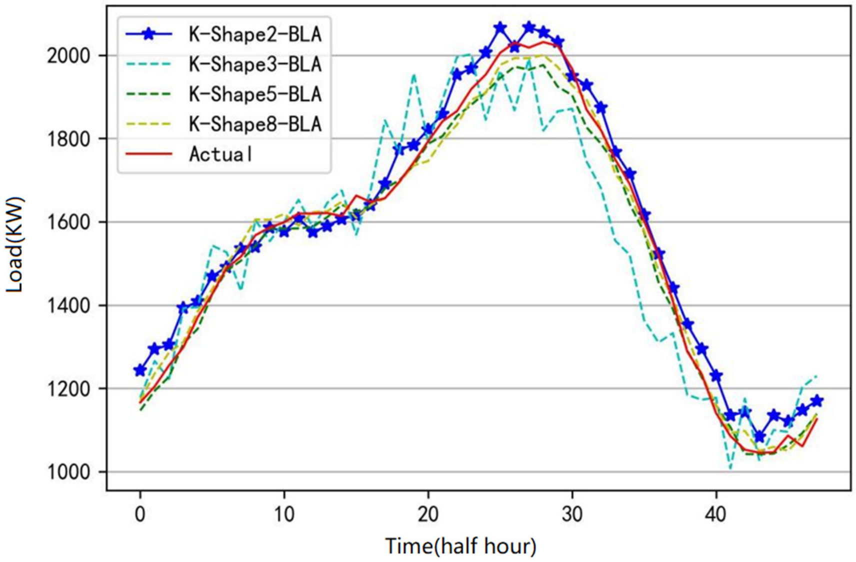


Figure 8. Graph of daily load forecast results.

It can be seen from Figure 4 that the BLA model performs well, and the error of the daily load prediction results after clustering is very small. To verify the effectiveness of the BLA model introduced by the attention mechanism established in this project, an attention mechanism ablation experiment was carried out on the same British power grid residential load data set. The BLA model proposed in this paper is compared with a Bi-LSTM model [40] which does not introduce an attention mechanism. To make the experimental results more convincing, this paper conducts experiments using different cluster center numbers, and the experimental results are shown in Table 4.

Table 4. The load forecasting experimental results with different cluster center numbers.

Algorithm	MAPE	MAE	MSE	RMSE	SMAPE
Bi-LSTM_2	1.88%	45.87	3560	59.67	2.84
BLA_2	1.49%	23.77	2994	55.66	1.84
Bi-LSTM_3	3.12%	34.75	1845	42.95	2.34
BLA_3	1.51%	23.75	3015	40.12	2.10
Bi-LSTM_5	2.78%	40.80	2561	50.61	2.82
BLA_5	2.33%	35.06	2503	48.16	2.62
Bi-LSTM_8	2.32%	34.75	1845	42.95	2.34
BLA_8	2.04%	28.56	1558	39.97	2.09
Bi-LSTM_10	2.31%	65.84	6918	83.18	4.10
BLA_10	1.86%	25.69	5360	72.50	3.53

The error percentages of the five clustering numbers of the proposed method and the original Bi-LSTM method are shown in the table. The MAPE of the proposed method in terms of two, three, five, eight, and ten clustering numbers is smaller than that of the original Bi-LSTM method, and the average absolute percentage error is reduced by 0.39%, 1.61%, 0.45%, 0.28%, and 0.63%, respectively. The experiments prove that the proposed model shows good performance in the prediction of this kind of data. The average MAPE of the original Bi-LSTM model is 2.48%, and the average MAPE of the proposed method is 1.85%.

To better illustrate the predictive efficiency of the BLA residential electricity consumption prediction model in the above experiments, the relative error and the average error are used to test the accuracy of the model. The average absolute percentage error percentage is expressed independently of scale and can be used to compare predictions at different scales. Therefore, the average absolute percentage error is used as the evaluation criterion for the predictive effectiveness of the model, and the confidence is high. A calculation of the average absolute percentage error of the BLA prediction model and the prediction data of the original Bi-LSTM was performed, and the results are shown in Figure 9.

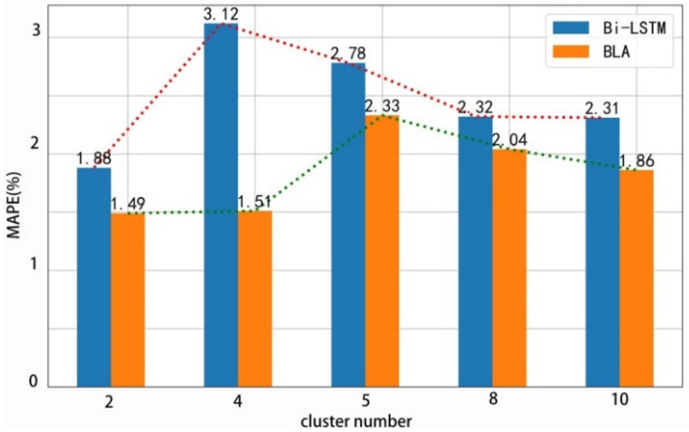
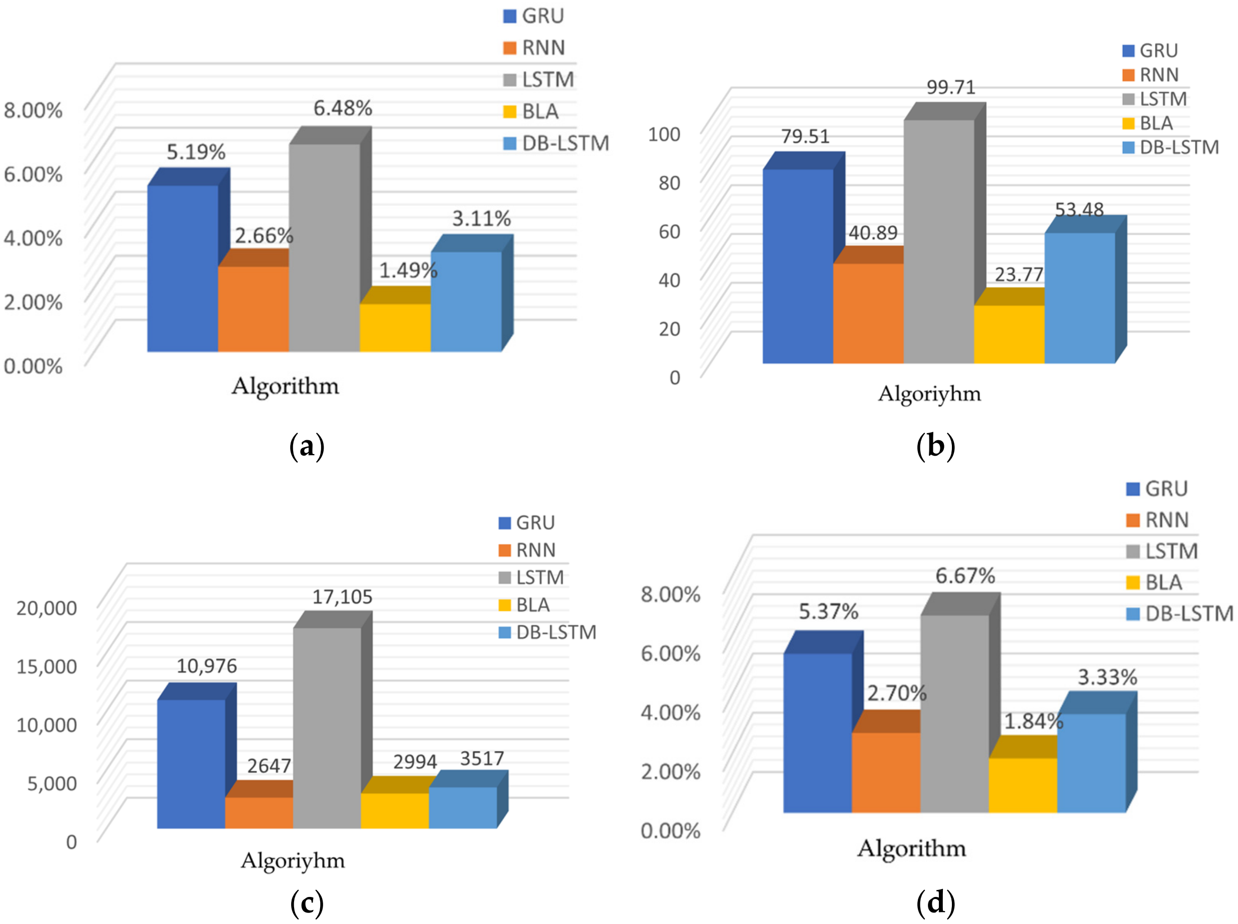


Figure 9. Comparison of MAPE between the Bi-LSTM model and the BLA model under different cluster numbers.

It can be seen that the Bi-LSTM errors are larger than the BLA residential electricity prediction model proposed in this topic. To further verify the complexity of the method in this paper, we calculated the parameter amount and the floating-point operations per second (FLOPS) of the BLA model in this paper. After calculation, the number of parameters of the BLA model is 88,148. The model proposed in this paper is mainly composed of Bi-LSTM and self-attention. The input vector dimension of the Bi-LSTM is 7 × 48, the hidden state dimension is 50, and the FLOPS is 80 K. The FLOPS of self-attention is 9.8 k. The remaining fully connected layer FLOPS is 44.2. So, the FLOPS of BLA is only 134. Since the model was very fast to train, we trained for 226 epochs in just 427 s on a single V100 GPU.

To further verify the effect of the BLA residential electricity consumption prediction model proposed in this paper, a horizontal comparison was performed. We chose the GRU, RNN, and LSTM algorithms commonly used in time series prediction for comparison. By introducing a self-attention mechanism, the BLA model we proposed improves the problem that the above algorithms have for the same time series information weight. The prediction results of the BLA, RNN, LSTM, and GRU models were compared and analyzed through experiments to determine the relative performance of the model proposed in this paper. In the second comparison experiment, to compare the superiority of the method in this paper, we conducted experiments on the same data set and compared the GRU [41], RNN [42], LSTM [43], and DB-LSTM [44] methods. Among them, the DB-LSTM method means that DBSCAN is used for clustering first, and then LSTM is used for prediction. The results are shown in Figure 10.

By observing Figure 10, it is not difficult to find that the model in this paper is more accurate than the other four models when clustering is the second class. The MAPE is 3.70%, 1.17%, 4.99%, and 1.62% lower than that of the GRU, RNN, LSTM, and DB-LSTM methods, and the prediction error is significantly reduced. There are two main reasons for this. First, our proposed model is based on a bidirectional recurrent neural network, which is highly robust for modeling time series data. Second, we introduce an attention mechanism on top of this. It enables the model to assign different weights to temporal features and extrinsic features simultaneously, thereby highlighting the key features of the input data that play a key role in the residents' load forecasting process, which helps to improve the accuracy of short-term load forecasting.



(**a**) (**b**)

(**c**) (**d**)

Figure 10. Performance comparison chart of different algorithms. (**a**) MAPE comparison. (**b**) MAE comparison. (**c**) MSE comparison. (**d**) SMAPE comparison.

4. Conclusions

This paper proposes a group resident daily load forecasting method fusing a self-attention mechanism based on load clustering to achieve accurate daily load forecasting. Firstly, a K-Shape-based clustering method is used for group resident loads. Because of the characteristics of high dimensionality and obvious differences in residential electricity consumption data, we use the K-shape algorithm to cluster the load data of group residents. It is divided into two types of typical residential electricity consumption curves: the high electricity consumption residential curve and the low electricity consumption residential curve. Therefore, in the subsequent prediction, the input vector similarity of the model is higher, and more accurate prediction results are obtained. Secondly, we use empirical mode decomposition to adaptively extract the IMF components and the residuals from the load data and reduce the noise by reducing the IMF component input. Finally, we propose an attention-based bidirectional neural network to predict residents' daily load. The self-attention mechanism is used to assign different weights to the time series information during prediction, which makes full use of the contextual information and the important information about the daily load of group residents.

To summarize, the group resident daily load forecasting method proposed in this paper has an average accuracy rate of over 98%. At the same time, we also compared previous prediction methods using the same data set. The experimental results show that the method we proposed performs better than the previous methods in five evaluation metrics. Therefore, the group resident daily load forecasting method fusing the self-attention mechanism based on load clustering proposed in this paper can accurately predict the daily load of residents. This method can help power companies to accurately understand the individualized and differentiated needs of users. It can improve the accuracy and efficiency of electricity consumption forecasting. Meanwhile, it can assist power companies to expand their business and provide data support for power demand management.

In future research, the method proposed in this paper can be used for short-term electricity load forecasting for other data sets. Other factors, including social factors and residential electricity consumption habits, can also be combined in the forecasting model. Other powerful artificial intelligence techniques can also be introduced into the hybrid model to further improve prediction accuracy.

Author Contributions: Conceptualization, J.C. and R.-X.Z.; Methodology, J.C.; Investigation, R.-X.Z. and Y.-B.Y.; Formal analysis, R.-X.Z.; Validation, R.-X.Z.; Resources, J.C.; Data curation, R.-X.Z. and C.-Q.L.; Writing—original draft, R.-X.Z. and C.-Q.L.; Writing—review and editing, C.-L.C., R.-X.Z. and Y.-B.Y.; Software, C.-Q.L.; Visualization, C.-Q.L.; Supervision, C.-L.C.; Project administration, C.-L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Development Plan projects of Jilin Province, No. 20210201134GX.

Institutional Review Board Statement: This study is based entirely on theoretical basic research. It does not involve humans.

Informed Consent Statement: This study is based entirely on theoretical basic research. It does not involve humans.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

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

References

1. Liu, X.; Zhang, Z.; Song, Z. A comparative study of the data-driven day-ahead hourly provincial load forecasting methods: From classical data mining to deep learning. *Renew. Sustain. Energy Rev.* **2020**, *119*, 109632. [CrossRef]
2. Khan, Z.A.; Ullah, A.; Haq, I.U.; Hamdy, M.; Mauro, G.M.; Muhammad, K.; Hijji, M.; Baik, S.W. Efficient short-term electricity load forecasting for effective energy management. *Sustain. Energy Technol. Assess.* **2022**, *53*, 102337. [CrossRef]
3. Yang, Y.; Hong, W.; Li, S. Deep ensemble learning based probabilistic load forecasting in smart grids. *Energy* **2019**, *189*, 116324. [CrossRef]
4. Amjady, N.; Keynia, F.; Zareipour, H. Short-term load forecast of microgrids by a new bilevel prediction strategy. *IEEE Trans. Smart Grid* **2010**, *1*, 286–294. [CrossRef]
5. Zhanghua, Z.; Qian, A. Present situation of research on microgrid and its application prospects in China. *Power Syst. Technol.* **2008**, *32*, 27–31.
6. Eskandarnia, E.; Al-Ammal, H.; Ksantini, R.; Hammad, M. Deep Learning Techniques for Smart Meter Data Analytics: A Review. *SN Comput. Sci.* **2022**, *3*, 243. [CrossRef]
7. Lu, J.; Zhu, Y.; Peng, W. Interactive demand response method of smart community considering clustering of electricity consumption behavior. *Autom. Electr. Power Syst.* **2017**, *41*, 113–120.
8. Hansen, J.V.; Nelson, R.D. Neural networks and traditional time series methods: A synergistic combination in state economic forecasts. *IEEE Trans. Neural Netw.* **1997**, *8*, 863–873. [CrossRef]
9. Bento, P.M.R.; Pombo, J.A.N.; Calado, M.R.A.; Mariano, S.J. Stacking Ensemble Methodology Using Deep Learning and ARIMA Models for Short-Term Load Forecasting. *Energies* **2021**, *14*, 7378. [CrossRef]
10. Zheng, C.; Wu, Y.; Chen, Z.; Wang, K.; Zhang, L. A Load Forecasting Method of Power Grid Host Based on SARIMA-GRU Model. In *National Conference of Theoretical Computer Science*; Springer: Singapore, 2021; pp. 135–153.
11. Fall, S.; N'Guessan, A.; Iraqi, F.; Koutouan, A. Forecasting the French Personal Services Sector Wage Bill: A VARIMA Approach. In *International Conference on Engineering, Applied Sciences, and System Modeling*; Springer: Cham, Switzerland, 2017; pp. 119–134.
12. Protić, M.; Shamshirband, S.; Petković, D.; Abbasi, A.; Kiah, M.L.M.; Unar, J.A.; Živković, L.; Raos, M. Forecasting of consumers heat load in district heating systems using the support vector machine with a discrete wavelet transform algorithm. *Energy* **2015**, *87*, 343–351. [CrossRef]
13. Zhang, N.; Li, Z.; Zou, X.; Quiring, S.M. Comparison of three short-term load forecast models in Southern California. *Energy* **2019**, *189*, 116358. [CrossRef]

14. Panda, S.K.; Ray, P. An Effect of Machine Learning Techniques in Electrical Load forecasting and Optimization of Renewable Energy Sources. *J. Inst. Eng. Ser. B* **2022**, *103*, 721–736. [CrossRef]
15. Zhou, K.; Wei, S.; Yang, S. Time-of-use pricing model based on power supply chain for user-side microgrid. *Appl. Energy* **2019**, *248*, 35–43. [CrossRef]
16. Xu, Z.; Yang, P.; Zheng, C.; Peng, J.; Chen, Q.; Huang, J. Control device development of user-side PV-ESS microgrid. *Power Syst. Technol.* **2017**, *41*, 426–433.
17. Li, Y.; Huan, J.; Cao, H.; Gao, C.; Zhang, X. Distribution net-work planning strategy based on integrated energy collaborative optimization. *Power Syst. Technol.* **2018**, *42*, 1393–1400.
18. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
19. Bian, H.; Zhong, Y.; Sun, J.; Shi, F. Study on power consumption load forecast based on K-means clustering and FCM–BP model. *Energy Rep.* **2020**, *6*, 693–700. [CrossRef]
20. Liu, F.; Dong, T.; Hou, T.; Liu, Y. A hybrid short-term load forecasting model based on improved fuzzy c-means clustering, random forest, and deep neural networks. *IEEE Access* **2021**, *9*, 59754–59765. [CrossRef]
21. Alipour, M.; Aghaei, J.; Norouzi, M.; Niknam, T.; Hashemi, S.; Lehtonen, M. A novel electrical net-load forecasting model based on deep neural networks and wavelet transform integration. *Energy* **2020**, *205*, 118106. [CrossRef]
22. Yao, S.J.; Song, Y.H.; Zhang, L.Z.; Cheng, X.Y. Wavelet transform and neural networks for short-term electrical load forecasting. *Energy Convers. Manag.* **2000**, *41*, 1975–1988. [CrossRef]
23. Wu, X.; Hong, D.; Chanussot, J. UIU-Net: U-Net in U-Net for Infrared Small Object Detection. *IEEE Trans. Image Process.* **2022**, *32*, 364–376. [CrossRef]
24. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5966–5978. [CrossRef]
25. Zheng, K.; Gao, L.; Liao, W.; Hong, D.; Zhang, B.; Cui, X.; Chanussot, J. Coupled convolutional neural network with adaptive response function learning for unsupervised hyperspectral super resolution. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 2487–2502. [CrossRef]
26. Yang, C.; Wang, W.; Zhang, X.; Guo, Q.; Zhu, T.; Ai, Q. A parallel electrical optimized load forecasting method based on quasi-recurrent neural network. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *696*, 012040. [CrossRef]
27. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef]
28. Cai, M.; Pipattanasomporn, M.; Rahman, S. Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. *Appl. Energy* **2019**, *236*, 1078–1088. [CrossRef]
29. Oprea, S.V.; Bâra, A. Ultra-short-term forecasting for photovoltaic power plants and real-time key performance indicators analysis with big data solutions. Two cases studies-PV Agigea and PV Giurgiu located in Romania. *Comput. Ind.* **2020**, *120*, 103230. [CrossRef]
30. Mughees, N.; Mohsin, S.A.; Mughees, A.; Mughees, A. Deep sequence to sequence Bi-LSTM neural networks for day-ahead peak load forecasting. *Expert Syst. Appl.* **2021**, *175*, 114844. [CrossRef]
31. Li, H.; Liu, H.; Ji, H.; Zhang, S.; Li, P. Ultra-short-term load demand forecast model framework based on deep learning. *Energies* **2020**, *13*, 4900. [CrossRef]
32. Kalman, D. A singularly valuable decomposition: The SVD of a matrix. *Coll. Math. J.* **1996**, *27*, 2–23. [CrossRef]
33. Stewart, G.W. On the early history of the singular value decomposition. *SIAM Rev.* **1993**, *35*, 551–566. [CrossRef]
34. Paparrizos, J.; Gravano, L. k-shape: Efficient and accurate clustering of time series. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, VIC, Australia, 31 May–4 June 2015; pp. 1855–1870.
35. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.-C.; Tung, C.C.; Liu, H.H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. London. Ser. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [CrossRef]
36. Zhang, B. Foreign exchange rates forecasting with an EMD-LSTM neural networks model. *J. Phys. Conf. Ser.* **2018**, *1053*, 012005. [CrossRef]
37. Gao, X.; Li, X.; Zhao, B.; Ji, W.; Jing, X.; He, Y. Short-term electricity load forecasting model based on EMD-GRU with feature selection. *Energies* **2019**, *12*, 1140. [CrossRef]
38. Li, Y.; Liu, X.; Xing, F.; Wen, G.; Lu, N.; He, H.; Jiao, R. Daily peak load prediction based on correlation analysis and Bi-directional long short-term memory network. *Power Syst. Technol.* **2021**, *45*, 2719–2730.
39. Yu, F.; Wang, L.; Jiang, Q.; Yan, Q.; Qiao, S. Self-Attention-Based Short-Term Load Forecasting Considering Demand-Side Management. *Energies* **2022**, *15*, 4198. [CrossRef]
40. Le, T.; Vo, M.T.; Vo, B.; Hwang, E.; Rho, S.; Baik, S.W. Improving electric energy consumption prediction using CNN and Bi-LSTM. *Appl. Sci.* **2019**, *9*, 4237. [CrossRef]
41. Wang, Z.; Zhao, B.; Ji, W.; Gao, X.; Li, X. Short-term load forecasting method based on GRU-NN model. *Autom. Electr. Power Syst.* **2019**, *43*, 53–58.
42. Tokgöz, A.; Ünal, G. A RNN based time series approach for forecasting turkish electricity load. In Proceedings of the 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–4.

43. Wu, K.; Gu, J.; Meng, L.; Wen, H.; Ma, J. An explainable framework for load forecasting of a regional integrated energy system based on coupled features and multi-task learning. *Prot. Control Mod. Power Syst.* **2022**, *7*, 24. [CrossRef]
44. Yang, W.; Shi, J.; Li, S.; Song, Z.; Zhang, Z.; Chen, Z. A combined deep learning load forecasting model of single household resident user considering multi-time scale electricity consumption behavior. *Appl. Energy* **2022**, *307*, 118197. [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.