Energy Reports 9 (2023) 3800–3812

Contents lists available at ScienceDirect

Energy Reports

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

Research paper

A hybrid Autoformer framework for electricity demand forecasting

Ziqian Wang [a,1], Zhihao Chen [a,1], Yang Yang [a,*], Chanjuan Liu [b], Xi'an Li [c], Jinran Wu [d,*]

[a] College of Automation & College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing 210023, PR China
[b] School of Business Administration and Customs, Shanghai Customs College, Shanghai 201204, PR China
[c] Ceyear Technology Co., Ltd., Qingdao 266000, PR China
[d] Institute for Learning Sciences & Teacher Education, Australian Catholic University, Brisbane 4001, Australia

ARTICLE INFO

Article history:
Received 21 November 2022
Received in revised form 30 January 2023
Accepted 27 February 2023
Available online 6 March 2023

Keywords:
Autoformer
Bootstrap
Auto-correlation
Deep learning
Prediction

ABSTRACT

Electricity demand forecasting is of great significance to the electricity system and residents' life, but it is difficult to forecast the electricity demand series because of the influence of cyclical factors. Electricity demand forecasting also faces the problem of small data amounts. Therefore, we need to design a model that is less affected by data volume and can cope with complex electricity demand series. Based on the Autoformer model, this paper establishes a novel forecasting framework with excellent performance. In the part of data preprocessing, multiple linear regression with 10 variables and Bootstrap processing are added. In the part of the model, the Auto-Correlation mechanism is modified to better extract the historical and nonlinear characteristics of electricity demand series from different time spans. Using this framework, we further analyze the impact of working days and seasonal changes on the electricity demand in Taixing City and New South Wales. In addition, we propose a new electricity demand forecasting method, which can adjust the original sequence according to the actual situation. The experimental results show that this method can achieve good precision in demand forecasting. Taking Taixing of China and New South Wales of Australia as examples, the forecasting performance with the proposed framework is better than that of Autoformer, Reformer, Informer, and other mainstream models. The forecasting indexes with our proposed framework of the test set are MAE: 35.05, RMSE: 47.28, MAPE: 1.63 in Taixing and MAE: 193.17, RMSE: 239.96, MAPE: 2.43 in NSW.

1. Introduction

For electricity enterprises, an accurate electricity supply is an essential step in regional grid planning (Lerner et al., 2009; Drew et al., 2017). Clearly, electricity demand forecasting is a key link to maintaining the dynamic balance between the electricity supply and electricity demand (Hong et al., 2020; Feinberg and Genethliou, 2005; Kumar Singh et al., 2012). Particularly, short-term electricity demand forecasting (Ud Din and Marnerides, 2017; Tang et al., 2019) is to forecast the next few hours or days, which can help electricity enterprises to set reasonable electricity prices and facilitate users to know the demand peak and low peak and arrange electricity reasonably. Meanwhile, when the electricity grid executes the transferred business, it can make an electricity generation plan and dispatch plan according to the short-term electricity demand forecasting data and the operation parameters of each generator (Metaxiotis et al., 2003). With the further development of the electricity market, the influence of short-term demand forecasting on the economic operation of the electricity system will become more and more obvious. Many models need to be based on a certain amount of historical information when making electricity demand forecasting. However, when electricity demand data is counted once a day and only a thousand or two thousand data can be used, the model tends to be less effective. So how to sufficiently extract the information from these data is essential in electricity demand forecasting. Meanwhile, electricity demand is often affected by many factors such as holidays, working days, seasons, and emergencies (Khatoon et al., 2014; Mustapha et al., 2015). These problems make the forecast more difficult. Therefore, how to build a short-term electricity demand forecasting method with accurate forecasting performance is crucial. This method can consider some factors affecting the electricity demand based on a small amount of data, and achieve effective short-term forecasting.

1.1. Literature review

In statistical learning, the Autoregressive Moving Average model (ARMA) (Pappas et al., 2010) and Autoregressive Integrated

* Corresponding authors.
E-mail addresses: b20090211@njupt.edu.cn (Z. Wang), a1365050734@gmail.com (Z. Chen), yyang@njupt.edu.cn (Y. Yang), liuchanjuan@shcc.edu.cn (C. Liu), lixian9131@163.com (X. Li), ryan.wu@acu.edu.au (J. Wu).

1 These authors contributed equally to this work.

https://doi.org/10.1016/j.egyr.2023.02.083

Moving Average model (ARIMA) (Zhu and Shen, 2012) are often used for short-term electricity demand forecasting (Wu et al., 2023). Researchers adopt the re-sampling method Bootstrap (Xiao et al., 2022) to deal with a small amount of data. Except for the small amount of electricity demand data, the electricity demand is not a stationary sequence due to the influence of various cyclical factors such as working days, weekends, holidays, and seasons. Researchers generally use the Multiple linear regression method (MLR) (Zhao et al., 2022; Wu et al., 2023) to describe the influence of these factors.

However, the forecasting performance of statistical learning methods is not ideal in the face of nonlinear electricity demand. To get better forecasting performance, researchers have proposed many neural network methods to deal with historical and nonlinear characteristics of time series. As early as 1990, Elman (1990) proposed a Recurrent Neural Network (RNN), based on a concept of memory mechanism: the network state information at the last moment will act on the network state at the next moment, to predict the time series. However, due to the vanishing gradient of the loss function and activation function in BPNN (Werbos, 1990), RNN cannot consider the information of past time steps. Hochreiter and Schmidhuber (1997) introduced Long–Short Term Memory (LSTM) in 1997 to control the transmission of information by introducing a forgetting gate, input gate, and output gate, which changes the cumulative multiplication in gradient into accumulation and solves the problem of vanishing gradient. At the same time, Cell State (Ct) variable is introduced to store the long-term situation so that the model has long-term memory. Gated Recurrent Unit (GRU) (Cho et al., 2014), the variant of LSTM, merges the input gate and the forgotten gate into the update gate and replaces the Ct variable in LSTM with a linear relation. These operations ensure the original forecasting accuracy and simplify the model calculation. Vaswani et al. (2017) proposed a model called Transformer based on self-attention mechanism and Feed Forward Neural Network (Sarraf, 2020), which is different from the sequential processing of RNN model and its evolution model LSTM. The model can dynamically compute based on all inputs and achieve better performance without using sequentially aligned cyclic architecture. This makes performance degradation due to long-term dependencies much less likely. But Transformer's self-attention mechanism is difficult to find reliable temporal dependence from complex temporal patterns. Wu et al. (2021) proposed an Autoformer model that uses deep decomposition architecture and an Auto-Correlation mechanism. Compared with the sparse attention mechanism of the Transformer, the Auto-Correlation mechanism realizes efficient connection at the sequence level, which can better aggregate information and break the bottleneck of information utilization. However, when dealing with complex and small amounts of electricity demand series, it is difficult for Autoformer to achieve ideal forecasting performance relying on the original Auto-Correlation. As for the research on Autoformer such as Zhong et al. (2022) and Jiang et al. (2022), there is still room for improvement in the improvement of Auto Correlation. Because of the limitations of these models, many hybrid models have been proposed to process the features of time series in stages (Ajitha et al., 2022; Inteha et al., 2021; Alhussein et al., 2020).

From the above studies in the field of statistical learning and neural network methods, we conclude the following three points should be considered to optimize the electricity demand forecasting performance:

(1) How to solve the problem that the model's forecasting performance is not ideal in the case of a small amount of data.

(2) How to design a novel hybrid framework to process the cyclical factors of electricity demand series in stages.

(3) How to improve the learning ability of historical and nonlinear characteristics of electricity demand series.

1.2. Motivation

Autoformer uses Auto-Correlation to extract sequence correlation rules of time series and works well when dealing with complex time series based on sufficient data. However, Autoformer, a single model, cannot solve the three problems we raised in the previous section well by relying on the original algorithm.

We want to improve on Auto-Correlation and design a novel hybrid framework based on Autoformer that can better deal with the complex sequences to achieve more accurate electricity demand forecasting and our framework can no longer be based on massive data support. This framework can help the electricity department to predict the electricity demand in the short term accurately and do corresponding planning.

1.3. Contributions

This paper analyzes and evaluates the electricity demand data set of Taixing City in Jiangsu Province and New South Wales in Australia. We extract weekly, monthly, yearly, and seasonal patterns to look for some peculiar change rules of electricity demand in Taixing and New South Wales. Our contributions can be summed up as follows.

(1) We design a novel hybrid Autoformer framework (HAFF) that can process the electricity demand sequence step by step. The first stage focuses on analyzing its periodic characteristics, and the second stage focuses on extracting its seasonal information and trend information. HAFF can achieve excellent forecasting results and does not need to be based on large amounts of electricity demand data.

(2) We analyzed electricity demand from two data sets over different time spans. An MLR model with 10 variables is constructed from the analysis results. The training set of the processed sequence is re-sampled through bootstrap and then spliced with the original sequence as the input sequence. Based on the above two processes, HAFF is able to analyze the cyclical factors of electricity demand in advance and reduce the impact of low data volume.

(3) We improve the Auto-Correlation mechanism to enable it can process the correlation of various data points in the sequence from different time spans. Improved Auto-Correlation can better extract historical and nonlinear characteristics of electricity demand series.

Finally, through the comparison experiments, our HAFF achieves better forecasting performance compared with Autoformer in the data set of Taixing and New South Wales.

1.4. Structure of this paper

The remainder of this article is organized as follows. Section 2 introduces the definitions of some terms, Section 3 details the proposed HAFF, and Section 4 shows the analysis of MLR and the comparative experiments of HAFF on electricity demand data from Taixing, China, and New South Wales, Australia. Section 5 summarizes this paper.

2. Definition of terms

2.1. Multiple linear regression

Multiple linear regression (MLR) is used to determine mathematical relationships between multiple random variables. In other words, MLR can examine how multiple independent variables are related to a dependent variable. Once each self-factor is identified to predict the dependent variable, information about multiple variables can be used to create an accurate forecasting of their level of influence on the resulting variable. The model creates relationships in the form of straight lines (linear) that are closest to all individual data points. Its general formula is as follows:

$$y_i = \omega_1 x_{i1} + \omega_2 x_{i2} + \cdots + \omega_j x_{ij} + b, \tag{1}$$

where y_i represents dependent variables, x_i represents explanatory variables, ω_j represents slope coefficients for each explanatory variable and b is constant term.

2.2. Autoformer

2.2.1. Series decomp

Based on the idea of moving average (Lu and Li, 2021), the serial decomposition part smooths the time series and separates the cyclical term from the trend term of the original series. The algorithm of the serial decomposition part is as follows:

$$\begin{aligned} X_t &= Avgpool(Padding(X)),\\ L_t &= (L + 2padding - kernelsize)/stride + 1,\\ X_s &= X - X_t. \end{aligned} \tag{2}$$

In the above formula, X is the original sequence, X_t stores the mean value of each sliding kernel to represent the trend term of the sequence, and X_s represents the smooth sequence with seasonality. Padding makes the length of X_t equal to X by adding length to the beginning and the end of the X sequence. *Avgpool* represents the operation of calculating the average value of the moving step size. Its operation on the length of the sequence corresponds to the second expression: L_t represents the length of the output sequence, L represents the length of the input sequence *kernelsize* represents the size of the sliding module, *stride* represents the step size of the moving module and the *padding* in this expression defaults to 0. The above formula is summarized as:

$$X_t, X_s = SeriesDecomp(X). \tag{3}$$

2.2.2. Autoformer-correlation

To capture the correlation information of time series, Autoformer uses the design method of an Auto-Correlation mechanism as the attention layer. The correlation coefficient R of the input time series can be obtained by fast Fourier transform (*FFT*) (Heckbert, 1995), which can reflect the similarity between the two input time series. In the application of electricity data, Autoformer introduces Q, K, and V (Vaswani et al., 2017) into the Auto-Correlation mechanism. The application of this attention mechanism can capture the cyclical characteristics of data very well. This series of treatments can be summarized as follows:

$$R_{Qt,Kt}(\tau) = F^{-1}(F(Q_t)F^{conj}(K_t)), \tag{4}$$

where $R_{Qt,Kt}$ is auto correlation between Q and K, F represents *FFT*, F^{-1} represents inverse transformation, and F^{conj} represents conjugate treatment.

After calculating autocorrelation, according to the aggregation processing method of time series, the information with the highest confidence of k sub-sequences is selected to reduce the complexity of the information.

$$\begin{aligned} \tau_1, \ldots, \tau_k &= argTopk(R_{Qt,Kt}(\tau)),\\ S_1, \ldots, S_k &= softmax(R_{Qt,Kt}(\tau_1), \ldots, R_{Qt,Kt}(\tau_k)). \end{aligned} \tag{5}$$

Then, after normalization operation (Ye et al., 2022), it is multiplied with the input time series after *Roll* operation.

$$AutoCorrelation(Q, K, V) = \sum_i^k Roll(V_t, \tau_i)S_i, \tag{6}$$

where *argTopk* refers to the maximum k parameters in the extracted sequence (the value of k is set manually). V_t refers to the input time series after mapping processing, and *Roll* refers to the rolling processing of the input sequence with a delaying length of τ.

2.2.3. Encoder and Decoder

In the Encoder part, the weighted sum of the original sequence is obtained by the Auto-Correlation, and the trend item is removed by the Series Decomp after adding the original sequence with the result of Auto-Correlation. After adding the Feed Forward and the second Series Decomp, the global seasonal item after treatment is passed to the Decoder. The algorithm of this part can be summarized as follows:

$$\begin{aligned} S_{en}^{l,1} &= SeriesDecomp(AutoCorrelation(X_{en}^{l-1}) + X_{en}^{l-1}),\\ S_{en}^{l,2} &= SeriesDecomp(FeedForward(S_{en}^{l,1}) + S_{en}^{l,1}),\\ X_{en}^{l} &= S_{en}^{l,2}, \end{aligned} \tag{7}$$

where X_{en}^l denotes the output of lth encoder layer, $S_{en}^{l,i}, i \in [1, 2]$ represents the seasonal component after the ith series decomposition block in the lth layer respectively.

In the Decoder part, season and trend items are dealt with. For the season item, it will add the result to itself after each Auto-Correlation processing, the season item is obtained as Q; secondly, the season item transmitted by Encoder is regarded as K and V; Q, K and V are used with Auto-Correlation to obtain the weighted sum of global historical information. After Series Decomp, repeat Feed Forward and Series Decomp operations to extract seasonal information and get the final seasonal forecasting. For the part of the trend item, the trend item of the original series is weighted and aggregated with the trend information obtained from the three Series Decomp to obtain the final trend prediction. The algorithm of this part can be summarized as follows:

$$\begin{aligned} S_{de}^{l,1}, T_{de}^{l,1} &= SeriesDecomp(AutoCorrelation(X_{de}^{l-1}) + X_{de}^{l-1}),\\ S_{de}^{l,2}, T_{de}^{l,2} &= SeriesDecomp(AutoCorrelation(S_{de}^{l,1} + X_{en}^{N}) + S_{de}^{l,1}),\\ S_{de}^{l,3}, T_{de}^{l,3} &= SeriesDecomp(FeedForward(S_{de}^{l,2}) + S_{de}^{l,2}),\\ T_{de}^{l} &= T_{de}^{l-1} + W_{l,1}T_{de}^{l,1} + W_{l,2}T_{de}^{l,2} + W_{l,3}T_{de}^{l,3},\\ X_{de}^{l} &= S_{de}^{l,3}, \end{aligned} \tag{8}$$

where X_{de}^l denotes the output of lth decoder layer, $S_{de}^{l,i}, T_{de}^{l,i}, i \in [1, 2, 3]$ represent the seasonal component and trend-cyclical component after the ith series decomposition block in the lth layer respectively. X_{en}^N represents the output of the N encoder layers. $W_{l,i}, i \in [1, 2, 3]$ represents the linear projector for the ith extracted trend $T_{de}^{l,i}$ as the adaptor.

Thus, we can draw a conclusion: For the season item, the Encoder–Decoder structure uses Auto-Correlation to perform dependency mining based on the cyclical nature of the sequence and aggregates sub-sequences with similar processes. For the trend item, the cumulative method is used to gradually extract the trend information from the predicted hidden variables.

3. The proposed model

3.1. Multiple linear regression

To eliminate the effects of working days, weekends, seasons, and other cyclical factors on electricity demand. We use the MLR model to construct 10 variables that fit these factors to describe the relevant effects.

At the same time, we use a harmonic function to represent time-related variables, which enables our MLR model to describe

Table 1
10 variables constructed for multiple linear regression model.

Variables	Definition	Variables	Definition
weekend	$= \begin{cases} 1 & \text{if in weekend} \\ 0 & \text{otherwise} \end{cases}$	Time	$= t = 1, 2, \ldots$
WeeklySin	$= \sin(\frac{2\pi t}{T}), T = 7$	WeeklyCos	$= \cos(\frac{2\pi t}{T}), T = 7$
MonthlySin	$= \sin(\frac{2\pi t}{T}), T = \frac{365}{12}$	MonthlyCos	$= \cos(\frac{2\pi t}{T}), T = \frac{365}{12}$
SeasonlySin	$= \sin(\frac{2\pi t}{T}), T = \frac{365}{4}$	SeasonlyCos	$= \cos(\frac{2\pi t}{T}), T = \frac{365}{4}$
YearlySin	$= \sin(\frac{2\pi t}{T}), T = 365$	YearlyCos	$= \cos(\frac{2\pi t}{T}), T = 365$

time series more accurately. Finally, we eliminate the results of the MLR model to obtain the relatively stationary series for subsequent processing. Details of the variables in this section are shown in Table 1.

Now we can construct a multiple linear regression model with those variables:

$$\begin{aligned} F(\omega) = & \omega_1 weekend + \omega_2 WeeklySin + \omega_3 WeeklyCos \\ & + \omega_4 MonthlySin + \omega_5 MonthlyCos \\ & + \omega_6 SeasonlySin + \omega_7 SeasonlyCos + \omega_8 YearlySin \\ & + \omega_9 YearlyCos \\ & + \omega_{10} time + b, \end{aligned} \tag{9}$$

where $\omega = [\omega_1, \omega_2, \ldots, \omega_{10}]^T$ is the coefficient vector for the variables, and b is an intercept item.

3.2. Bootstrap

After obtaining the output result of the MLR model, we subtract the original electricity demand sequence X_0 from the result $F(\omega)$ to reduce the influence of the variables constructed in the MLR model. The result after subtracting is X. Then, we divided the training set and test set and conducted a Bootstrap resampling operation on the training set to avoid information leakage during subsequent training. Finally, we spliced the resampled sequence with X. The preceding operations are as follows:

$$\begin{aligned} X &= X_0 - F(\omega), \\ X_1 &= Bootstrap(X), \\ X &= Concat(X_1, X). \end{aligned} \tag{10}$$

3.3. Auto-correlation

Auto-Correlation uses Autocorrelation Function (ACF) (Scargle, 1989) to calculate the degree of correlation between a data point and its delay of k unit data point. ACF is calculated as:

$$R_{xx}(k) = \lim_{L\to\infty} \frac{1}{L} \sum_{t=0}^{L-1} X_t X_{t-k}. \tag{11}$$

The processing helps Auto-Correlation extract similar sequences from the overall sequence for information aggregation. This means that this processing can improve the learning ability of historical and nonlinear characteristics of electricity demand series. However, we found that ACF can only calculate the correlation degree between a certain day and k days before it if the electricity demand data is counted at intervals of one day. That means its ability to learn these characteristics is limited when considering the small amount of data and the influence of cyclical factors. Therefore, our idea is to allow the ACF of Auto-Correlation to calculate the correlation of various data points in the series over different time spans. We add one-dimensional convolution to Auto-Correlation to handle input sequences. After this layer of processing, the length of the sequence will be reduced to $1/n$ of the original length. It is clear that one data point in the new sequence can represent the information of the original n data points. So through ACF, we can approximate the correlation on n days span. This improvement enables the model to fully learn the historical and nonlinear characteristics of the electricity demand series even with a small amount of data. The processing of this part is shown in Fig. 1.

Since Auto-Correlation uses the attention mechanism, so we let the original input Q, K, V pass one-dimensional convolution. The processed length of Q, K, and V will be $1/n$ of the original length. Then we repeat the Auto-Correlation results of one-dimensional convolution processing. So that the result is the same size as the one without the one-dimensional convolution. Finally, we add up the results of these different treatments. Improved Auto-Correlation can analyze the trend characteristics and historical characteristics of electricity demand series from different time spans. The frame diagram of the Auto-Correlation mechanism before and after modification is shown in Fig. 2. The specific implementation process of the algorithm is shown in Algorithm 1.

3.4. Overall structure of HAFF

We added the MLR model and Bootstrap to the data preprocessing part of Autoformer and improved its Auto-Correlation. Finally, we proposed HAFF, and the overall algorithm flow of HAFF is as follows:

Step 1: Multiple linear regression. This section sets the variables associated with the electricity demand series. Then we subtract the result of this part to get the final residual. This step makes the sequence relatively stable in advance to facilitate the analysis of subsequent models. Since the cyclical factors in the sequence are not eliminated, we will deal with them in the next steps.

Step 2: Bootstrap. In response to the small amount of electricity demand data, we set a specific threshold for the sequence with insufficient length, extend the training set of it with the Bootstrap method, and then splice it with the original sequence to Step 3.

Step 3: Encoder. For the new sequence after processing, the original features of the samples are put into the Encoder for Auto-Correlation and Feed Forward processing. After each segment, Series Decomp is conducted to transfer the season information of the whole sequence to Step 4.

Step 4: Decoder. First, Series Decomp is used to decompose the processed new sequences into trend and season items, and then two improved Auto-Correlations are used to aggregate the season item to aggregate similar sequences. In the second Auto-Correlation, the information from Step 3 is used as the K, V parameter. The improved Auto-Correlation algorithm can improve the ability of Autoformer model to capture the historical and nonlinear characteristics of sequences.

Step 5: Predict. The forecast results consist of the final season and trend item. The season item in the Decoder will be processed continuously. Finally, the trend item and season item after processing are added to get the forecast results.

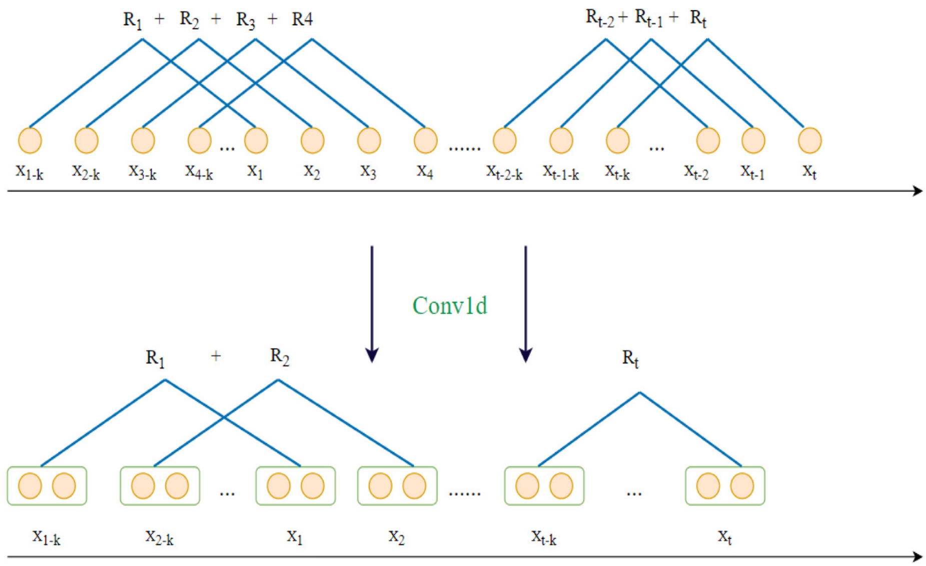


Fig. 1. The schematic diagram of the ACF before and after one-dimensional convolution, in which the sequence length L becomes $\frac{L}{n}$. ACF can calculate the correlation on the time span of n days. This figure takes $n = 2$ as an example.

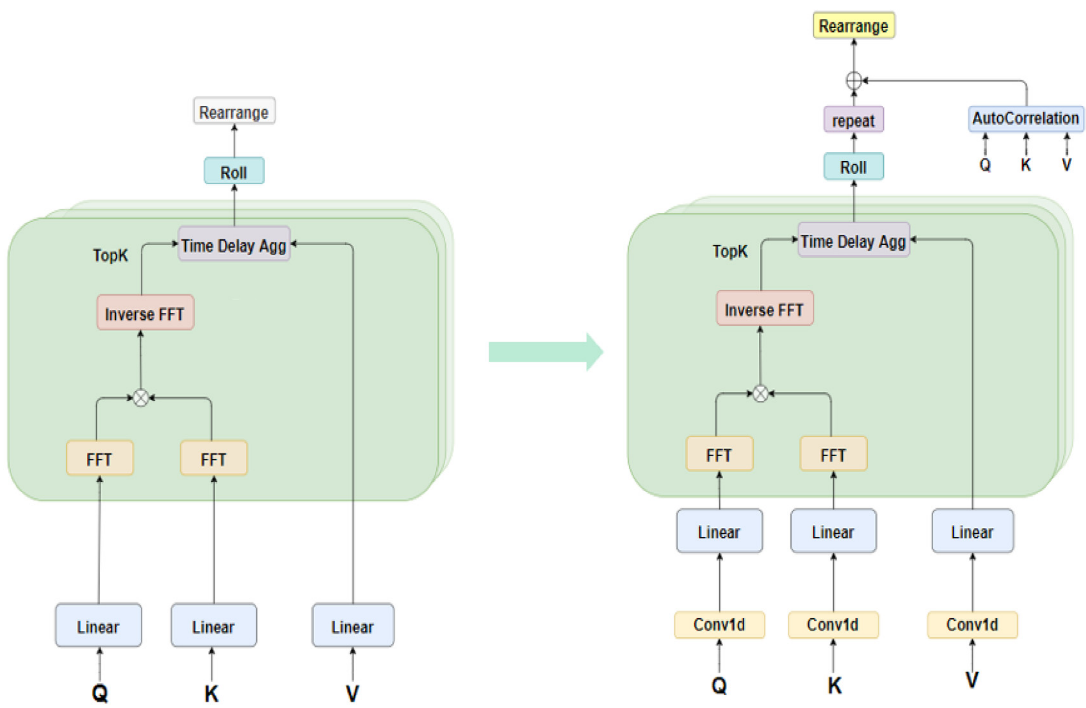


Fig. 2. For the Auto-Correlation part before and after modification, Q, K, V enter Auto-Correlation after one-dimensional convolution, and the output is repeated to the original size and added with the original Auto-Correlation output.

Here, parameters in Step 1 can be added according to specific situations, and appropriate parameters can improve the final forecasting accuracy. In Step 2, if features are constructed for samples in advance, sequence expansion will also expand corresponding features. The overall structure of HAFF is shown in Fig. 3 below. The specific implementation process of the HAFF model is provided in Algorithm 2.

4. Case studies

In the case studies, we first analyzed the data set and the MLR part of the design and then designed a comparative experiment for HAFF to analyze the performance of HAFF further.

Algorithm 1: Modified Auto-Correlation

Input: Electricity demand data $X_i (i = 1, 2, 3, \ldots, T)$.
Output: New sequence after time delay aggregation $X_{atten} (i = 1, 2, 3, \ldots, T \times \frac{d_{model}}{N-head})$.

1 X_i is transposed to Q, K, V by W_Q, W_K, W_V $(i = 1, 2, 3, \ldots, T \times \frac{d_{model}}{N-head})$;
2 Q, K, V goes through the convolution layer to Q_1, K_1, V_1 $(i = 1, 2, 3, \ldots, \frac{T}{n} \times \frac{d_{model}}{Nhead})$;
3 $AutoCorrelation(Q, K, V)$ and $AutoCorrelation(Q_1, K_1, V_1)$ are added together to get X_{atten}.

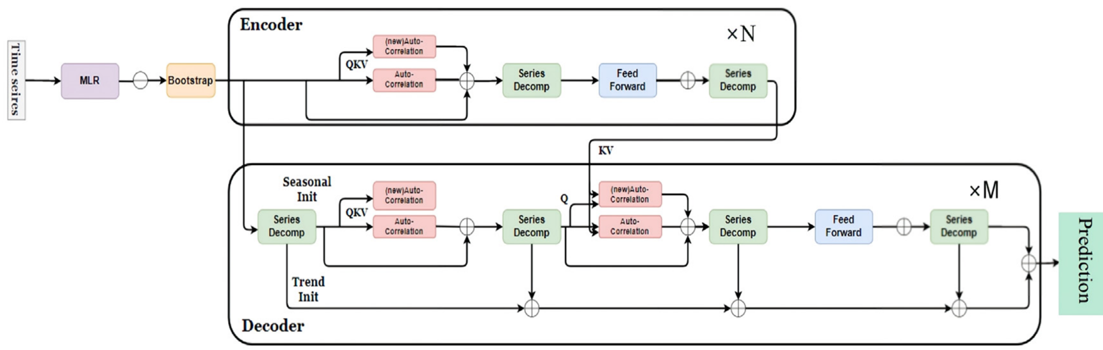


Fig. 3. The whole structure of HAFF.

Algorithm 2: HAFF

Input: Electricity demand data $X_i (i = 1, 2, 3, \ldots, T)$.
Output: Final forecasting result $Y_i (i = 1, 2, 3, \ldots, T)$.

X_i enters MLR model to get $F(\omega)$;
X_i minus $F(\omega)$ and then bootstrap expands to get $X_i^{'}$;
Determine the total number of sub-sequences C;
$X_i^{'}$ is decomposed into X_s and X_t by Series Decomposition;
for $k = e - layers$ **do**
  $X_i^{'}$ goes into the Encoder layer and passes through Modified Auto-Correlation to get a new sequence after delay aggregation then linked by residuals to X_{en}^{l-1} $(i = 1, 2, 3, \ldots, T \times \frac{d_{model}}{N-head})$;
  X_{en}^{l-1} is goes through Series Decomp to get $S_{en}^{l,1}$ $(i = 1, 2, 3, \ldots, T \times \frac{d_{model}}{N-head})$;
  $S_{en}^{l,1}$ passes through the FeedForward layer and is connected to $X_{en}^{l,1}$ by residuals $(i = 1, 2, 3, \ldots, T \times \frac{d_{model}}{N-head})$;
  $X_{en}^{l,1}$ is decomposed into $S_{en}^{l,2}$ by Series Decomp;
  $S_{en}^{l,2}$ goes into the Decoder layer and is transposed to $K^{'}$, $V^{'}$ by $W_K^{'}$, $W_V^{'}$ $(i = 1, 2, 3, \ldots, T \times \frac{d_{model}}{N-head})$.
end
for $k = d - layers$ **do**
  X_s goes into the Decoder layer and passes through Modified Auto-Correlation then linked by residuals to $X_{de}^{l,1}$ $(i = 1, 2, 3, \ldots, T \times \frac{d_{model}}{N-head})$;
  $X_{de}^{l,1}$ is decomposed into $S_{de}^{l,1}$ and $T_{de}^{l,1}$ by Series Decomp;
  $S_{de}^{l,1}$ is transposed to $Q^{'}$ by $W_Q^{'}$ $(i = 1, 2, 3, \ldots, T \times \frac{d_{model}}{N-head})$;
  $Q^{'}$ and the $K^{'}$, $V^{'}$ from Encoder pass through Modified Auto-Correlation then linked by residuals to $X_{de}^{l,2}$;
  $X_{de}^{l,2}$ is decomposed into $S_{de}^{l,2}$ and $T_{de}^{l,2}$ by Series Decomp;
  $S_{de}^{l,2}$ passes through the FeedForward layer and is connected to $X_{de}^{l,3}$ by residuals;
  $X_{de}^{l,3}$ is decomposed into $S_{de}^{l,3}$ and $T_{de}^{l,3}$ by Series Decomp;
  $T_{de}^{l,1}$, $T_{de}^{l,2}$, $T_{de}^{l,3}$ are added up by weigh matrix $W_{l,1}$, $W_{l,2}$, $W_{l,3}$ respectively to get T_{de}^{l};
  T_{de}^{l}, $S_{de}^{l,3}$ are added up to get the final forecasting result Y_i.
end
for $k = 1, 2, 3, \ldots, C$ **do**
  Send updated weighs to the central server, aggregate weighs, and update the global model.
end

To quantitatively analyze the forecasting results, we introduce Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) as our evaluation indexes. The simulation performance and reasonable degree of different models are measured by the following indexes:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{13}$$

and

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{14}$$

where n is the sample size, and y_i and $\hat{y}_i$ are the ith observed value and the forecasting respectively.

Our HAFF uses the adam optimizer. Batch size is set to 32, seq-len is set to 48; d_{model} is 8, d_{ff} is 512; e-layers is 2, d- layers is 1; N-head is set to 8; dropout is 0.05 and lr is 0.01. The n in the modified Auto Correlation is set to 8. The training process is early stopped within 10 epochs. The whole validation experiment is performed on PyCharm Community Edition 2021.1.2 x64 environment, using Windows 10 and a 4.20 GHz AMD Ryzen 7 4800H CPU with 64-bit support and 16 GB RAM.

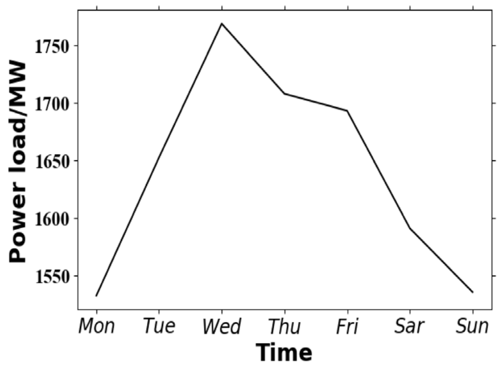
4.1. Case 1: Electricity demand in Taixing

In this section, we use an electricity demand dataset from Taixing, China for single-step forecasting. The data was recorded once a day for a total of 1178 data points (from May 13, 2018, to August 2, 2021). The data set is divided into training set (60%), validation set (20%) and test set (20%). The 1178 data points are expanded to 1854 data points after Bootstrap. The skewness before the extended sequence is 0.377 and the kurtosis is 0.559. The skewness after the extended sequence is 0.553 and the kurtosis is 1.075. Fig. 4 shows the transformation law of electricity demand under different time spans in Taixing. Table 2 records the single-use performance of the three adjustment methods used in our HAFF and the performance indicators of the comparison models. Figs. 5 and 6 show part of the single-step forecasting curve captured from the experimental result curve of the final test set. The case will be analyzed in detail in the next three sections.

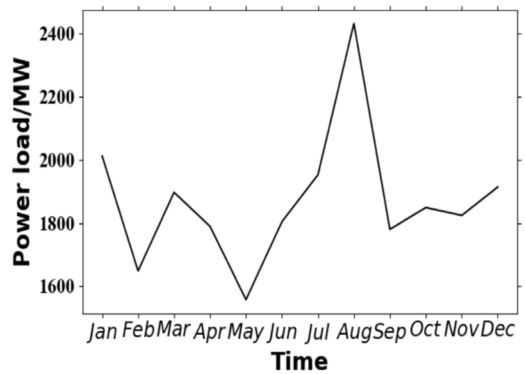
4.1.1. Analysis of data set and MLR

In this section, we analyze the electricity demand data set of Taixing to obtain the variation rule of its electricity demand under different time spans. The researched information will serve as a basis for the design of MLR and an important criterion for testing whether these variables can describe the electricity demand.

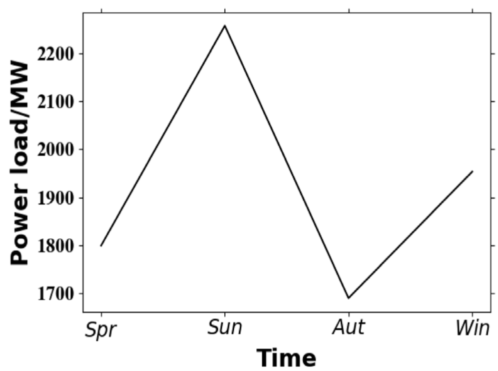
We find that weekly, monthly, and seasonal electricity demand levels show different rules. In general, electricity demand shows an upward trend. The specific electricity demand variation rules are shown in Fig. 4



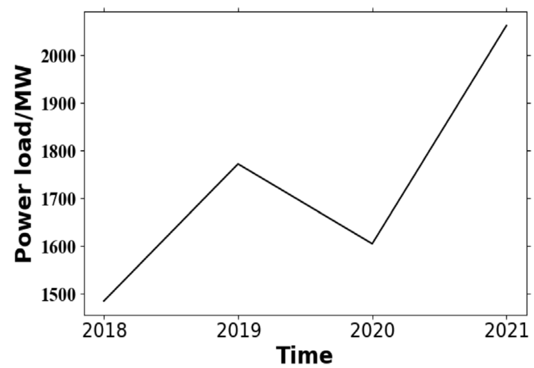
(a) Weekly trend



(b) Monthly trend



(c) Seasonal trend



(d) Yearly trend

Fig. 4. Transformation law of electricity demand under different time span in Taixing.

Table 2
One-step forecasting results of HAFF, its applied methods and comparison models on Taixing dataset.

Model	Train			Test		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Autoformer	124.629	163.565	7.069	110.36	142.92	5.28
Autoformer with MLR	111.48	141.64	6.15	99.35	134.02	4.73
Autoformer with Bootstrap	95.40	123.52	5.44	141.19	169.03	6.55
Autoformer with new Auto-Correlation	40.11	52.06	2.21	67.09	82.43	3.23
BPNN	191.46	246.77	11.48	143.71	214.07	6.88
LSTNet	115.47	163.97	6.85	208.45	259.47	9.46
LSTM	42.09	57.35	2.37	137.67	170.01	6.55
Transformer	56.362	74.998	3.183	102.86	142.71	4.61
Reformer	46.20	61.65	2.61	57.74	86.94	2.63
Informer	58.12	75.23	3.24	147.68	195.18	6.54
HAFF	36.24	47.24	2.05	35.05	47.28	1.63

From the figure, we can see that there are differences in electricity demand between working days and weekends. During working days, especially Wednesday and Thursday, the electricity demand is significantly higher than the other days. By calculating the average electricity demand, we find that the average electricity demand for working days in Fig. 4(a) is 1671 MW/d, while that for weekends, is 1563 MW/d, which indicates that there is a significant difference between the electricity demand for working days and that of weekends, and this difference is consistent with most cases after multiple calculations. At the same time, we calculated the average electricity demand of Taixing each quarter: 1775 MW/d in spring, 2030 MW/d in summer, 1795 MW/d in autumn, and 1884 MW/d in winter. This statistical result is consistent with Fig. 4(c), so the electricity demand of Taixing is higher in summer and winter.

In the experiment, we obtain the specific coefficients of these 10 variables through training. According to these variables and coefficients, we construct an MLR model:

$$\begin{aligned} F(\omega) = & -23.54 weekend - 15.76 WeeklySin - 7.89 WeeklyCos \\ & + 8.91 MonthlySin - 2.92 MonthlyCos \\ & - 8.85 SeasonlySin - 30.93 SeasonlyCos + 83.15 YearlySin \\ & - 7.66 YearlyCos \\ & + 0.38 time + 1673. \end{aligned} \tag{15}$$

The variables of the specific parameters we see the coefficient of $weekend$ is −23.54, indicating that the above condition has a

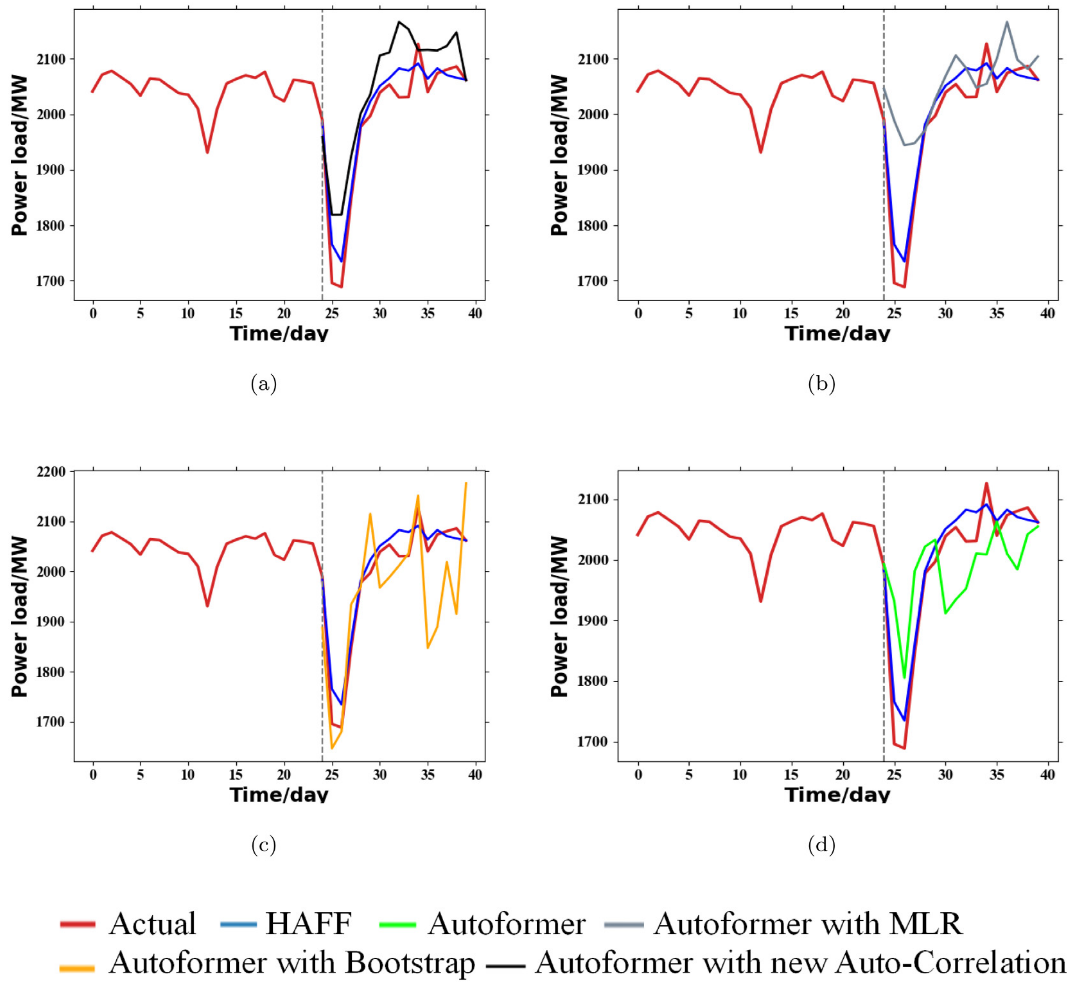


Fig. 5. Autoformer and Autoformer with applied methods: the captured electricity demand forecasting one-step ahead prediction curve of Taixing from 2021/4/16 to 2021/5/26.

negative effect on Taixing electricity demand. According to the *time* coefficient 0.38, we conclude that despite some negative effects, the overall electricity demand of Taixing is still on the rise. The analysis of the original data sets shows that the MLR model we construct is effective.

4.1.2. The performance of the methods we proposed

This section analyzes the first four comparison experiments in Table 2 to verify the performances of the three adjustment methods we propose.

The results show that MLR, Bootstrap, and new Auto-Correlation adjustment methods are used to improve the forecasting performance of Autoformer in the Taixing electricity demand training set. In the test set, the single use of MLR (MAE: 99.35, RMSE: 134.02, MAPE: 4.73) and the new Auto-Correlation (MAE: 67.09, RMSE: 82.43, MAPE: 3.23) improve the performance of the original model. In both the training set and test set, the new Auto-Correlation method we use provides the most significant improvement over the original model. This is because when the electricity demand fluctuates are more dramatic, our method can better capture the trend information of the electricity demand and optimize the sequence decomposition part of the Autoformer.

It is worth noting that Bootstrap (MAE: 141.19, RMSE: 169.03, MAPE: 6.55) alone did not work well in the test set. To verify the effectiveness of Bootstrap, we conducted additional experiments HAFF without Bootstrap, and the results are recorded as the training set (MAE: 42.50, RMSE: 54.29, MAPE: 2.38) and Test set (MAE: 52.26, RMSE: 70.49, MAPE: 2.49). We find that when the Bootstrap method is removed, the HAFF becomes less effective, thus the Bootstrap method can be used with the other two methods to achieve better results. This is because time series forecasting has certain requirements for the length of the series.

Figs. 5(a), 5(b), 5(c), and 5(d) visually show the forecasting curve comparison of Autoformer with new Auto-Correlation, MLR, Bootstrap, and original Autoformer. According to Fig. 5, Autoformer with new Auto-Correlation has the best fitting performance on the electricity demand curve and can predict the fluctuation of electricity demand. Autoformer with MLR also reflects its tendency to conform to the fluctuation of electricity demand to some extent. This indicates that our proposed method is reasonable and can effectively predict the electricity demand under a complicated situation.

4.1.3. Comparative experiment between our HAFF and other comparison models

In this section, we use the simple feed-forward multilayer neural network BPNN (Rumelhart et al., 1986) and the classical time series forecasting model LSTNet (Lai et al., 2018), LSTM

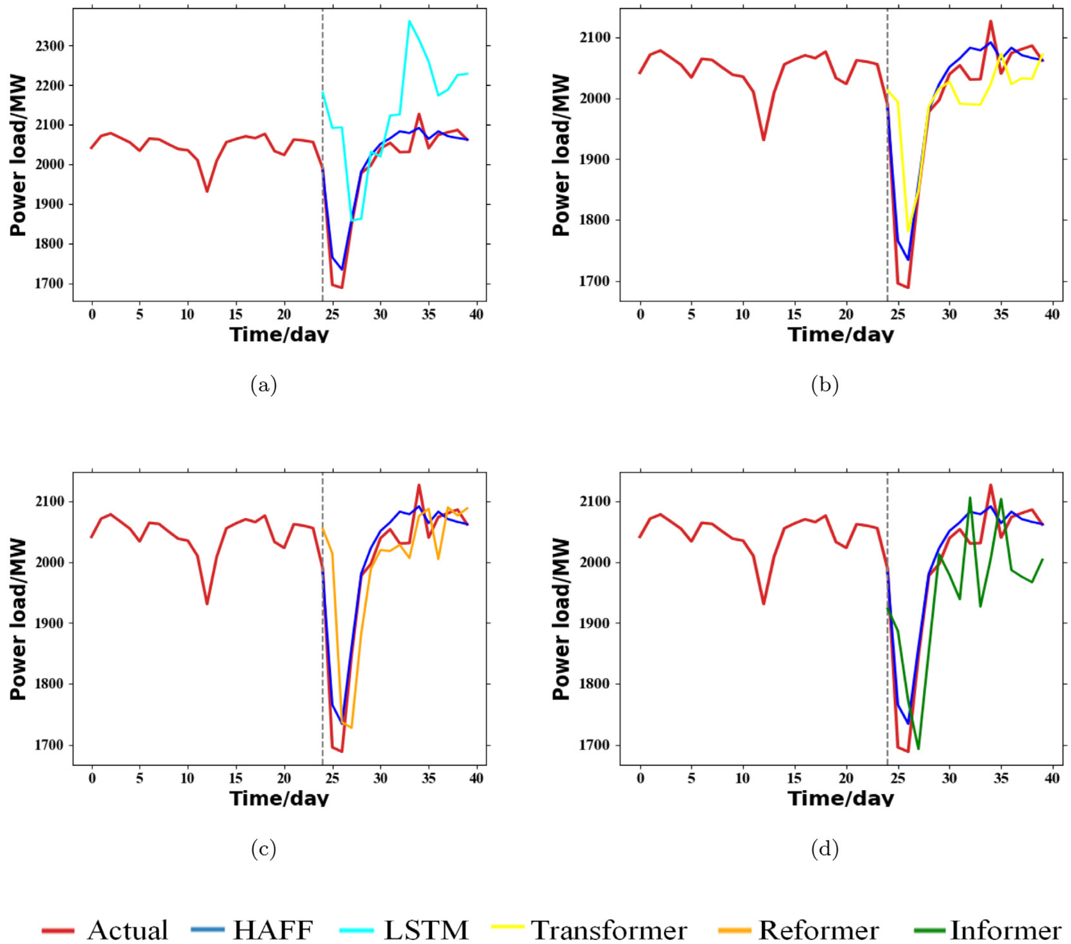


Fig. 6. HAFF and comparison models: the captured electricity demand forecasting one-step ahead prediction curve of Taixing from 2021/4/16 to 2021/5/26.

(Hochreiter and Schmidhuber, 1997), Transformer (Hochreiter and Schmidhuber, 1997), Reformer (Kitaev et al., 2019) and Informer (Zhou et al., 2021). The hyperparameters of each comparison model are recorded in Table A.4. Take the electricity demand data set in Taixing as an example, the error indexes of the above comparison model and HAFF under single-step forecasting are calculated and counted. In addition, we also used the statistical model ARIMA for similar processing. Through the training set, p, d, and q are respectively set as 2, 1, and 4. The final error index in the test set is (MAE: 54.31, RMSE: 74.04, and MAPE: 2.54). According to the results in Table 2 and Fig. 6, HAFF has the best forecasting performance in both the training set and the test set. Compared with the original Autoformer, the test set performance in MAE of HAFF is improved by about 68%. Compared with Reformer, another excellent time series forecasting model, the test set performance in MAE of HAFF is improved by about 39%. It is worth noting that, except for the Autoformer model, the validation performance of the above comparison models is lower than that of the training set, and the LSTM model is the most obvious. This is one of the reasons we finally chose Autoformer as our base model. Through a series of adjustment methods, the forecasting ability of Autoformer is greatly improved.

Moreover, we choose the model with a better forecasting index from the comparison models, then we plot forecasting results with LSTM, Transformer, Reformer, Informer, and HAFF in Figs. 6(a), 6(b), 6(c), and 6(d) respectively. As can be seen from the partial forecasting curve in Fig. 6(c), although the forecasting performance of the Reformer is second only to that of the HAFF in this experiment, it cannot well fit the sudden fluctuation of Taixing electricity demand, and this phenomenon is most obvious in the second half of the sequence. Curves in Figs. 6(a), 6(b) and 6(d) also reflect this problem: when the stationary series has fluctuations, the model is often difficult to make accurate forecasting due to the lack of analysis ability of local trends.

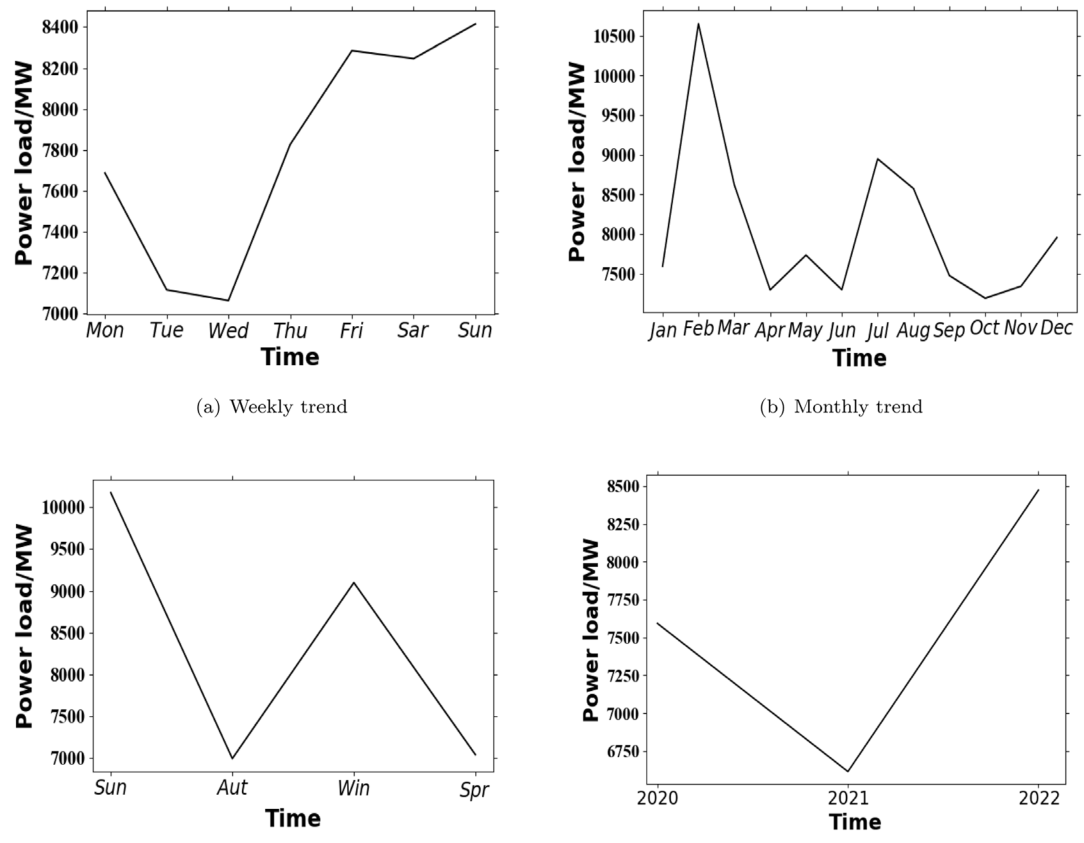
4.2. Case 2: Electricity demand in NSW

In this section, to prove the generality of our model, we selected the electricity demand data in New South Wales, Australia, which has more significant fluctuations and is recorded once a day. A total of 1012 data points (from January 2, 2020, to September 7, 2022) are expanded to 1588 data points by Bootstrap. The data set is divided into a training set (60%), a validation set (20%), and a test set (20%). The skewness before the extended sequence is 0.425 and the kurtosis is −0.302. The skewness after the extended sequence is 0.254 and the kurtosis is −0.571. Table 3 records the forecasting error indexes for each experiment using this data set. Figs. 8 and 9 show the single-step forecasting curves of HAFF and other experiments.

4.2.1. Analysis of data set and MLR

In this section, we analyze the electricity demand data set of NSW, and we find that it also shows special rules on weekly, monthly, yearly, and seasonal levels in Fig. 7.

From Fig. 7(a), we find that the electricity demand of NSW is opposite to that of Taixing in weekly mode, and the electricity demand for weekends is much higher than that of working days. After calculating the average electricity demand, we find that



(a) Weekly trend (b) Monthly trend

(c) Seasonal trend (d) Yearly trend

Fig. 7. Transformation law of electricity demand under different time span in NSW.

Table 3
One-step forecasting results of HAFF, its applied methods and comparison models on NSW dataset.

Model	Train			Test		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Autoformer	325.65	429.35	4.19	464.87	592.69	5.89
Autoformer with MLR	425.55	562.39	5.42	453.37	562.39	5.80
Autoformer with Bootstrap	354.50	462.13	4.52	375.83	490.64	4.77
Autoformer with new Auto-Correlation	232.08	306.73	3.01	264.77	322.12	3.38
BPNN	672.88	812.96	8.44	651.38	781.01	9.48
LSTNet	540.59	653.68	7.29	513.81	631.76	6.57
LSTM	293.82	386.30	3.83	564.72	702.01	6.90
Transformer	417.96	520.38	5.38	589.37	705.93	7.17
Reformer	278.05	364.53	3.59	285.22	360.80	3.57
Informer	471.54	604.89	6.10	542.38	670.36	6.99
HAFF	327.04	423.67	4.18	264.21	332.67	3.32

the electricity demand is 7609 MW/d on working days and 9533 MW/d on weekends.

By observing Figs. 7(b) and 7(c), we also find that the electricity demand of NSW is consistent with Taixing in terms of overall change because the electricity demand in summer and winter is significantly higher than that in the other two seasons. The average electricity demand of the four seasons is 7176 MW/d in spring, 8127 MW/d in summer, 7358 MW/d in autumn, and 8493 MW/d in winter.

However, it is worth noting that the yearly electricity demand data, different from the change rule of Taixing, does not show a trend of decline or rise. Therefore, when building the MLR variable, we removed the *time* variable to avoid it affecting the accuracy of the other variables. From the above analysis, we built an MLR model:

$$\begin{aligned} F(\omega) = &-395.38weekend + 107.62WeeklySin \\ &+ 326.62WeeklyCos - 46.16MonthlySin \\ &+ 29.29MonthlyCos \\ &+ 48.45SeasonlySin - 77.79SeasonlyCos - 4.74YearlySin \\ &- 470.66YearlyCos \\ &+ 7775.37. \end{aligned} \tag{16}$$

4.2.2. The performance of the methods we proposed

Through the first four comparison experiments in Table 3, it can be found that all three adjustment methods we use can improve the performance of the Autoformer model in the test set. Among them, our new Auto-Correlation method (MAE: 264.77,

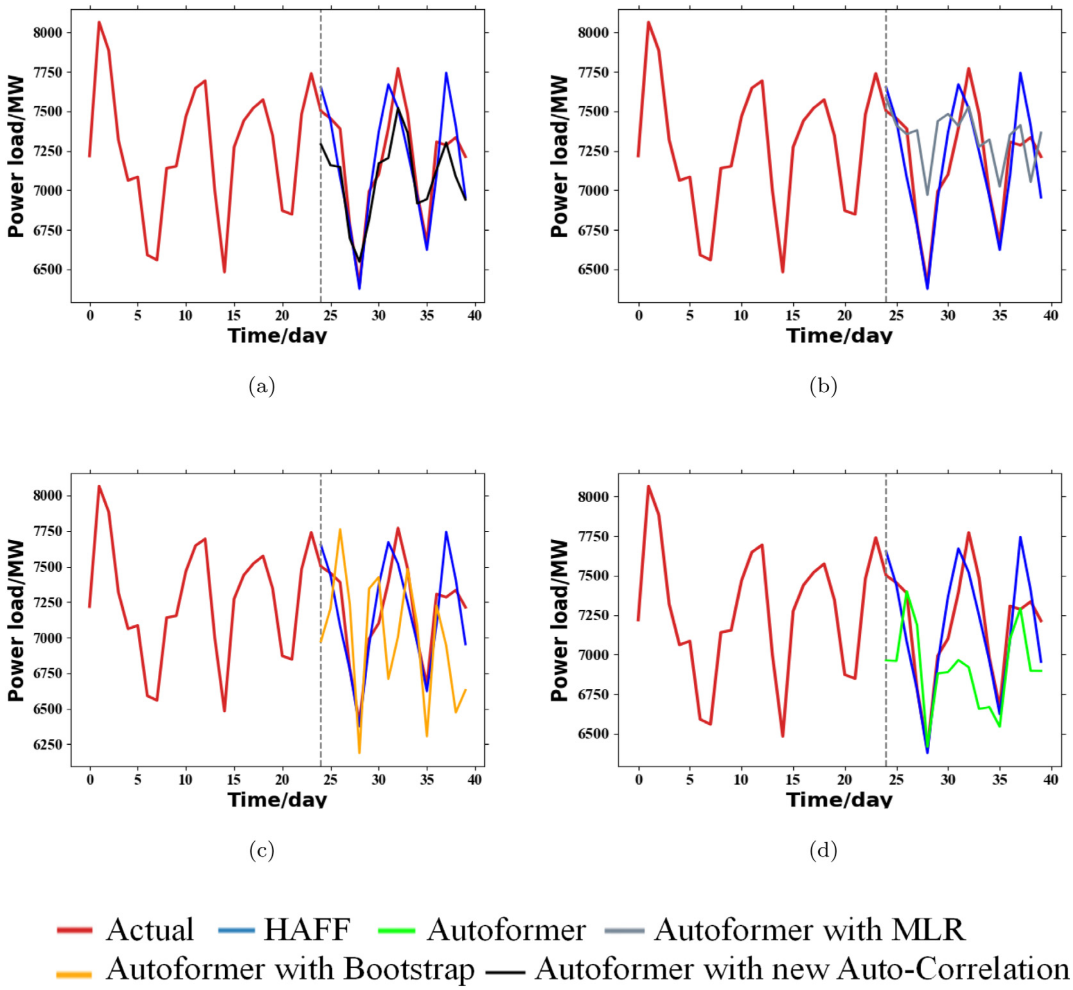


Fig. 8. Autoformer and Autoformer with applied methods: the captured electricity demand forecasting one-step ahead prediction curve of NSW from 2022/3/6 to 2022/4/15.

RMSE: 322.12, MAPE: 3.38) has the most significant improvement. This is because, in addition to the electricity demand fluctuation caused by some cyclical factors, NSW itself has a large electricity demand fluctuation, which makes it difficult for the original Autoformer to analyze the trend part of the data set.

Figs. 8(a), 8(b), 8(c), and 8(d) visually show forecasting curves of Autoformer with new Auto-Correlation, Autoformer with MLR, Autoformer with Bootstrap, and Autoformer. By observing Fig. 8, we find that new Auto-Correlation is still the most effective one. We also found that Bootstrap does not allow the model to fit the spike of fluctuations well, but Autoformer with Bootstrap has a great improvement in the fitting of the overall amplitude of the electricity demand compared with the original Autoformer.

4.2.3. Comparative experiment between our HAFF and other comparison models

In this section we still use the LSTNet, BPNN, LSTM, Transformer, Reformer, and Informer as comparison models. For the ARIMA model, p, d, q are set to 3, 1, 6 through the training set and the error-index is (MAE: 412.65, RMSE: 518.55, MAPE: 5.16). From the forecasting results shown in Table 3 and Fig. 9, HAFF remains the best forecasting performance. In the test set, HAFF in MAE improves by about 43% compared to the original Autoformer and 7% compared to Reformer. Moreover, the performance of HAFF in the test set is still higher than that in the training set.

As seen from Fig. 9, it is difficult for the comparison model to fit the fluctuation trend of electricity demand, especially the peak part of the fluctuation, when the value changes sharply and the frequency of fluctuation increases substantially. HAFF can still achieve good forecasting performance under complex electricity demand sequences.

5. Conclusion

In recent years, with the advancement of new industrialization, the change range of short-term electricity demand has increased, which brings more pressure to electricity system operators. Therefore, improving the accuracy of short-term electricity demand forecasts has a very positive effect on the maintenance security and management efficiency of the electricity system. In order to improve the accuracy of the short-term forecast of electricity demand forecast, we designed a hybrid model HAFF which combines MLR and Bootstrap with an improved Auto Correlation mechanism. Taking complex electricity demand sequences of Taixing City in China and New South Wales State in Australia as examples, the experimental results show that HAFF can achieve excellent short-term forecasting effect without relying on a large amount of electricity demand data, and HAFF can pay attention to the seasonal and trend information of electricity demand so as to accurately track the daily variation of electricity demand.

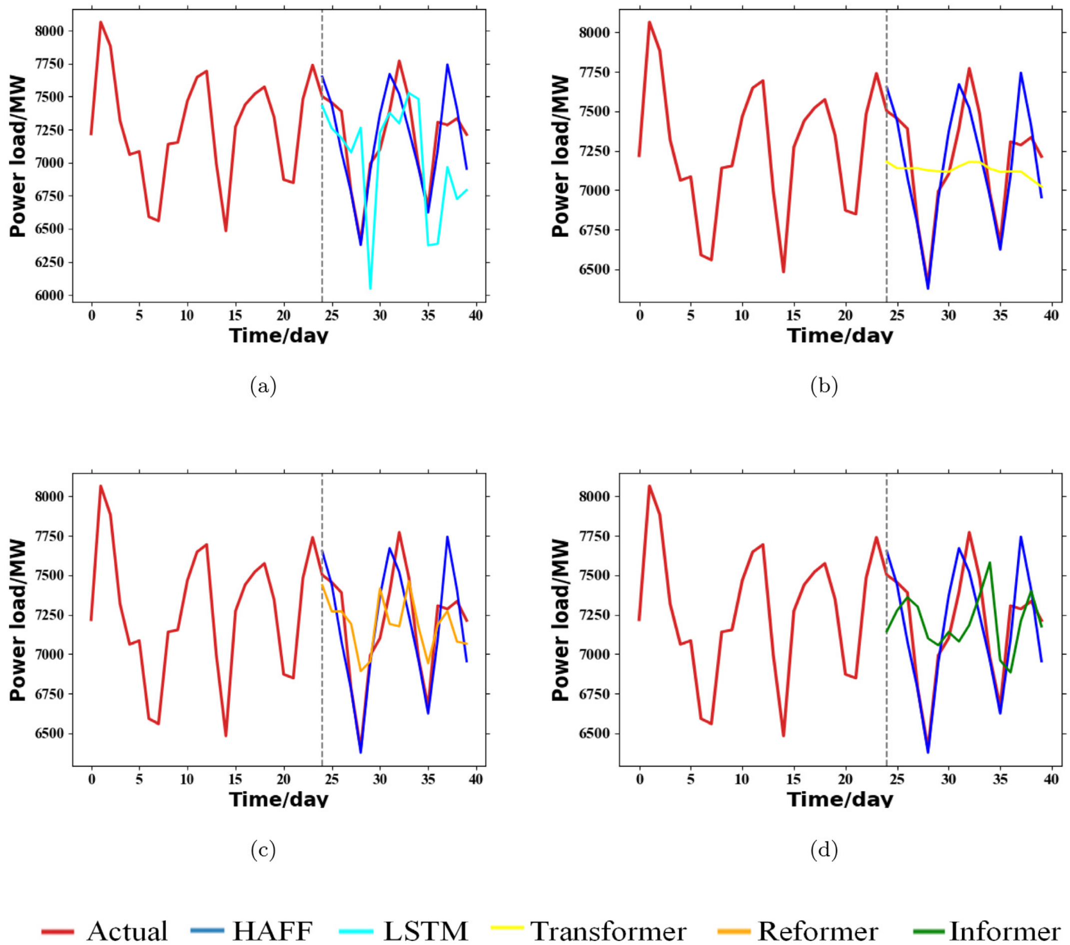


Fig. 9. HAFF and comparison models: the captured electricity demand forecasting one-step ahead prediction curve of NSW from 2022/3/6 to 2022/4/15.

Compared with the mainstream model, HAFF can provide more accurate basic data for the electricity market and facilitate its reasonable pricing. HAFF can also help sales companies reduce the discrepancy between reported and actual demand in pursuit of higher economic returns. Therefore, HAFF is of great significance to realize the modernization of electricity system management and the transformation of electricity-selling companies into the electricity spot market.

Although HAFF has achieved excellent prediction results, it still has some limitations. For example, HAFF is not very effective in multi-step prediction. The MLR part of HAFF can only be used as data preprocessing, where the weight value may not reach the best situation. Although HAFF's Auto Correlation mechanism is optimized, HAFF can only extract information from input sequences and cannot analyze electricity demand data from a long-term perspective. Based on the above limitations, our follow-up research directions are as follows: (1) Use the Boost or Bagging algorithm in HAFF to improve sample utilization. (2) Find ways to use Bootstrap more effectively such as combining with mixup methods. (3) Seek for more accurate variables to improve MLR's fitting performance to electricity demand and to cope with some emergencies. (4) Incorporate MLR into the Autoformer model to iterate over the weights of variables. (5) Autoformer focuses on reducing the complexity of the attention computation but this operation is not significant for time series prediction. We still need to optimize the Auto Correlation. (6) Improve the accuracy of HAFF in medium- and long-term forecasts.

CRediT authorship contribution statement

Ziqian Wang: Software, Visualization, Formal analysis, Writing – original draft. **Zhihao Chen:** Software, Visualization, Formal analysis. **Yang Yang:** Writing – review & editing, Funding acquisition. **Chanjuan Liu:** Writing – review & editing. **Xi'an Li:** Writing – review & editing. **Jinran Wu:** Supervision, Formal analysis, Writing – original draft, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

The work is supported by the Australian Research Council project (grantnumber DP 160104292), the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) (grant number CE140100049) and the Ministry of education of Humanities and Social Science project (22YJC630083). This work is supported in part by the National

Table A.4
The hyperparameters of the comparison models.

Models	Hyperparameters
LSTNet	hidCNN = 100; hidRNN = 100; CNN_{kernel} = 6; $highway_{window}$ = 24; skip = 24; hidskip = 5; lr = 0.01
LSTM	hidden layers = 800; num layers = 1; lr = 0.01
BPNN	hidden layers = 2048; num layers = 1; lr = 0.01
Transformer	d_{model} = 512; d_{ff} = 512; dropout = 0; n layers = 1; N head = 8; lr = 0.01
Reformer	d_{model} = 16; d_{ff} = 64; dropout = 0.2; e layers = 2; d layers = 1; N head = 8; lr = 0.01
Informer	d_{model} = 16; d_{ff} = 64; dropout = 0.2; e layers = 2; d layers = 1; N head = 8; lr = 0.01
Autoformer	d_{model} = 8; d_{ff} = 512; dropout = 0.05; e layers = 2; d layers = 1; N head = 8; lr = 0.01

Natural Science Foundation of China under Grant 61873130 and Grant 61833011, in part by the Natural Science Foundation of Jiangsu Province, China under Grant BK20191377, in part by the 1311 Talent Project of Nanjing University of Posts and Telecommunications. Also, the authors would like to acknowledge Mr. Hu Zhou and Mr. Hao Lou for their advice on the original manuscript.

Appendix. The hyperparameter setting

See Table A.4.

References

Ajitha, A., Goel, Maitri, Assudani, Mohit, Radhika, Sudha, Goel, Sanket, 2022. Design and development of residential sector load prediction model during covid-19 pandemic using lstm based rnn. Electr. Power Syst. Res. 212, 108635.
Alhussein, Musaed, Aurangzeb, Khursheed, Irtaza Haider, Syed, 2020. Hybrid cnn-lstm model for short-term individual household load forecasting. IEEE Access 8, 180544–180557.
Cho, Kyunghyun, van Merrienboer, Bart, Gülçehre, Çaglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, Bengio, Yoshua, 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: EMNLP.
Drew, Daniel R., Cannon, Dirk J., Barlow, Janet F., Coker, Phil J., Frame, Thomas H.A., 2017. The importance of forecasting regional wind power ramping: A case study for the uk. Renew. Energy 114, 1201–1208.
Elman, Jeffrey L., 1990. Finding structurein time. Cogn. Sci. 14 (2), 179–211.
Feinberg, Eugene A., Genethliou, Dora, 2005. Load forecasting. In: Applied Mathematics for Restructured Electric Power Systems. Springer, pp. 269–285.
Heckbert, Paul, 1995. Fourier transforms and the fast fourier transform (fft) algorithm. Comput. Graph. 2, 15–463.
Hochreiter, Sepp, Schmidhuber, Jürgen, 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Hong, Tao, Pinson, Pierre, Wang, Yi, Weron, Rafał, Yang, Dazhi, Zareipour, Hamidreza, 2020. Energy forecasting: A review and outlook. IEEE Open Access J. Power Energy 7, 376–388.
Inteha, Azfar, et al., 2021. A gru-ga hybrid model based technique for short term electrical load forecasting. In: 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques. ICREST, IEEE, pp. 515–519.
Jiang, Yuqi, Gao, Tianlu, Dai, Yuxin, Si, Ruiqi, Hao, Jun, Zhang, Jun, Wenzhong Gao, David, 2022. Very short-term residential load forecasting based on deep-autoformer. Appl. Energy 328, 120120.
Khatoon, Shahida, Kr Singh, Arunesh, et al., 2014. Effects of various factors on electric load forecasting: An overview. In: 2014 6th IEEE Power India International Conference. PIICON, IEEE, pp. 1–5.
Kitaev, Nikita, Kaiser, Lukasz, Levskaya, Anselm, 2019. Reformer: The efficient transformer. In: International Conference on Learning Representations.
Kumar Singh, Arunesh, Khatoon, S., Muazzam, Md, Chaturvedi, D.K., et al., 2012. Load forecasting techniques and methodologies: A review. In: 2012 2nd International Conference on Power, Control and Embedded Systems. IEEE, pp. 1–10.
Lai, Guokun, Chang, Wei-Cheng, Yang, Yiming, Liu, Hanxiao, 2018. Modeling long-and short-term temporal patterns with deep neural networks. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 95–104.
Lerner, Jeff, Grundmeyer, Michael, Garvert, Matt, 2009. The importance of wind forecasting. Renew. Energy Focus 10 (2), 64–66.
Lu, Jianhua, Li, Ze, 2021. Early prognostic factors of patients with acquired pneumonia under the analysis of autoregressive integrated moving average model-based pathogenic infectious influenza virus. Results Phys. 22, 103908.
Metaxiotis, K., Kagiannas, A., Askounis, D., Psarras, J., 2003. Artificial intelligence in short term electric load forecasting: a state-of-the-art survey for the researcher. Energy Convers. Manage. 44 (9), 1525–1534.
Mustapha, M., Mustafa, M.W., Khalid, S.N., Abubakar, I., Shareef, H., 2015. Classification of electricity load forecasting based on the factors influencing the load consumption and methods used: An-overview. In: 2015 IEEE Conference on Energy Conversion. CENCON, IEEE, pp. 442–447.
Pappas, S.Sp., Ekonomou, L., Karampelas, P., Karamousantas, D.C., Katsikas, S.K., Chatzarakis, G.E., Skafidas, P.D., 2010. Electricity demand load forecasting of the hellenic power system using an arma model. Electr. Power Syst. Res. 80 (3), 256–264.
Rumelhart, David E., Hinton, Geoffrey E., Williams, Ronald J., 1986. Learning representations by back-propagating errors. Nature 323 (6088), 533–536.
Sarraf, Aydin, 2020. A tight upper bound on the generalization error of feedforward neural networks. Neural Netw. 127, 1–6.
Scargle, Jeffrey D., 1989. Studies in astronomical time series analysis. iii-fourier transforms, autocorrelation functions, and cross-correlation functions of unevenly spaced data. Astrophys. J. 343, 874–887.
Tang, Xianlun, Dai, Yuyan, Wang, Ting, Chen, Yingjie, 2019. Short-term power load forecasting based on multi-layer bidirectional recurrent neural network. IET Gener., Transm Distrib. 13 (17), 3847–3854.
Ud Din, Ghulam Mohi, Marnerides, Angelos K., 2017. Short term power load forecasting using deep neural networks. In: 2017 International Conference on Computing, Networking and Communications. ICNC, IEEE, pp. 594–598.
Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Łukasz, Polosukhin, Illia, 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, Vol. 30.
Werbos, Paul J., 1990. Backpropagation through time: what it does and how to do it. Proc. IEEE 78 (10), 1550–1560.
Wu, Jinran, Levi, Noa, Araujo, Robyn, Wang, You-Gan, 2023. An evaluation of the impact of covid-19 lockdowns on electricity demand. Electr. Power Syst. Res. 216, 109015.
Wu, Haixu, Xu, Jiehui, Wang, Jianmin, Long, Mingsheng, 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Adv. Neural Inf. Process. Syst. 34, 22419–22430.
Xiao, Ling, Li, Miaotong, Zhang, Shenghui, 2022. Short-term power load interval forecasting based on nonparametric bootstrap errors sampling. Energy Rep. 8, 6672–6686.
Ye, Keying, Han, Zifei, Duan, Yuyan, Bai, Tianyu, 2022. Normalized power prior bayesian analysis. J. Statist. Plann. Inference 216, 29–50.
Zhao, Zixi, Wu, Jinran, Cai, Fengjing, Zhang, Shaotong, Wang, You-Gan, 2022. A statistical learning framework for spatial–temporal feature selection and application to air quality index forecasting. Ecol. Indic. 144, 109416.
Zhong, Weiheng, Mallick, Tanwi, Meidani, Hadi, Macfarlane, Jane, Balaprakash, Prasanna, 2022. Explainable graph pyramid autoformer for long-term traffic forecasting. arXiv preprint arXiv:2209.13123.
Zhou, Haoyi, Zhang, Shanghang, Peng, Jieqi, Zhang, Shuai, Li, Jianxin, Xiong, Hui, Zhang, Wancai, 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. pp. 11106–11115.
Zhu, Xianghe, Shen, Min, 2012. Based on the arima model with grey theory for short term load forecasting model. In: 2012 International Conference on Systems and Informatics (ICSAI2012). IEEE, pp. 564–567.