



A novel decomposition-based ensemble model for short-term load forecasting using hybrid artificial neural networks

Zhiyuan Liao¹ · Jiehui Huang¹ · Yuxin Cheng¹ · Chunquan Li¹ · Peter X. Liu²

Received: 13 May 2021 / Accepted: 20 September 2021 / Published online: 20 January 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Highly accurate short-term load forecasting (STLF) is essential in the operation of power systems. However, the existing predictive methods cannot achieve an effective balance between prediction accuracy and computational cost. Furthermore, the prediction residual is rarely used to improve the predictive accuracy in STLF. This paper proposes a novel decomposition-based ensemble model for the STLF task. First, an optimized empirical wavelet transform (OEWTF) is developed to rationally decompose the STLF load by combining the approximate entropy method with the empirical wavelet transform. Particularly, OEWTF improves both prediction accuracy and computational cost in STLF. Second, a new hybrid machine learning method (named master learner) is proposed by rationally combining long short-term memory networks (LSTMs) with broad learning system (BLS) in STLF, effectively strengthening the predictive accuracy without significantly increasing the computational cost. Third, a residual learning model (named residual learner) is developed in the master learner to extract the effective predictive information from residual results, further improving the prediction accuracy in STLF. Fourth, an auxiliary learner is proposed by introducing another BLS to connect the input and output of the proposed model, enhancing the predictive robustness. The proposed decomposition-based ensemble model is compared with state-of-the-art and traditional models in STLF. Experimental results show that the model not only has high predictive accuracy and robustness but also low computational cost.

Keywords Short-term load forecasting · Optimized empirical wavelet transform · Approximate entropy · Long short-term memory · Broad learning system

Nomenclature

Abbreviations

AL	Auxiliary learner
ANN	Artificial neural network
APEN	Approximate entropy
BLS	Broad learning system
BLSTM	Hybrid model composed of BLS and LSTM
CD	Critical distance of the Nemenyi test
DBN	Deep neural network
DNN	Deep neural network

DWT	Discrete wavelet transform
EMD	Empirical mode decomposition
EWT	Empirical wavelet transform
FT	Fourier transform
LSTM	Long short-term memory
MAE	Means absolute error
ML	Master learner
NSW	New South Wales, Australia
OEWTF	Optimized empirical wavelet transform
RL	Residual learner
RMSE	Root means square error
SLTF	Short-term load prediction
SVR	Support vector regression
SSA	Singular spectrum analysis
VMD	Variational mode decomposition
WT	Wavelet transforms

Zhiyuan Liao, Jiehui Huang and Yuxin Cheng contributed to the work equally and should be regarded as co-first authors.

✉ Chunquan Li
lichunquan@ncu.edu.cn

¹ The School of Information Engineering, Nanchang University, Nanchang 330031, China

² The Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada

Indices and parameters

a_f	The activation function
B^+	The pseudo-inverse of B

DL	Whether to have the direct link between the input layer and output layer
d	The distance between two reconstructed m -dimensional vectors E_n
E_k	The enhancement nodes of BLS
E_n	The reconstructed m -dimensional vector by APEN
eta	The learning rate
e_n	The components decomposed by EWT
f_{RBF}	The radial basis functions
f_t	The forget gate of LSTM
g_t	The input node of LSTM
h_t	The intermediate output of LSTM
I	The identity matrix
i_t	The input gate of LSTM
J_i	The feature nodes of BLS
M	The maximum value of the Fourier spectrum
m	The length of the comparison vector in APEN
m_i	The maximum number of iterations
N	The number of components of EWT
n	The total number of test samples
n_{ds}	The number of data sets in Nemenyi test
n_e	The number of enhancement nodes
	layer and output layer
n_f	The number of feature nodes
n_h	The number of hidden nodes
n_l	The number of learning algorithms in Nemenyi test
o_t	The output gate of LSTM
q_α	The critical value based on the Studentized Range statistic
r	The similarity in APEN
r_b	The random batch size of each time
r_m	The randomization methods
std	The standard deviation of the time series
s_t	The memory cell state of LSTM
s_{RBF}	The spread of radial basis functions
v_m	The momentum
W	The weight matrix between the feature nodes, enhancement nodes, and output of BLS
X	The input data of BLS
\hat{Y}	The final prediction value of BLS
Y	The actual value
y_i	The load data at a certain moment
y_{max}	The maximum value in the load data
y_{min}	The minimum value in the load data
\hat{y}_i	The normalized result of y_i
\hat{y}_i	The predicted data of y_i
κ	The relative amplitude ratio of EWT
ω_n	The midpoint of the corresponding frequencies of the two adjacent maximum values above the threshold

τ_n	The transition phase of EWT
ρ	The interval for reorganizing the components $e_n(t)$
λ	The regularization parameter of BLS

Functions

$F^{-1}(\bullet)$	The inverse Fourier transform
$\eta(\bullet)$	The linear transformation
$\xi(\bullet)$	The nonlinear activation function
$\sigma(\bullet)$	The sigmoid activation function
$\varphi(\bullet)$	The tanh function.

Variables

$g(t)$	The signal to be decomposed
$\hat{T}_n(\omega)$	The scaling function
$\hat{P}_n(\omega)$	The wavelet function
$\beta(x)$	An arbitrary function that satisfies $C^k([0, 1])$
$K_g^\varepsilon(n, t)$	The detail coefficient
$\underline{K}_g^\varepsilon(0, t)$	The approximate coefficient
\overline{P}_n	The complex conjugate of P_n
\overline{T}_1	The complex conjugate of T_1

1 Introduction

Electric load forecasting, a time series forecasting task, plays an essential role in social, economic, and various other aspects of the energy sector [1]. In particular, electric load forecasting can be categorized into three types according to the forecasting horizon, namely, short-term, medium-term, and long-term load forecasting [2–5]. Short-term load prediction (STLF) is one of the most important aspects of deregulated grid planning and operation. Accurate load forecasting can be helpful for efficient energy management [6]. However, the prediction accuracy of STLF is influenced by many factors [7, 8], such as insufficient historical data, economic environment, development status, unstable meteorological environment, seasonal changes, and the development of the power grid. Therefore, the electrical charge has large fluctuation and uncertainty characteristics, which means accurate prediction is still a challenge.

In the past few years, many models have been developed for STLF, which can be divided into mathematical-statistical models and machine learning models [9]. Mathematical-statistical models include autoregressive integrated moving average models [10, 11], regression methods [12], and linear regression models [13]. These methods can provide high predictive accuracy for linear systems. However, the predictive accuracy of nonlinear forecasting tasks is limited [14]. Therefore, these methods are not suitable for highly complex and nonlinear electricity load prediction.

The other approach is machine learning models such as artificial neural networks (ANNs) [15], support vector regression (SVR) [16], deep neural networks (DBNs) [17],

broad learning systems (BLSs) [18], and long short-term memory (LSTM) [19]. They have been applied to various fields such as economic [20] and construction [21]. The machine learning model can directly learn the relationship between input and output. Therefore, such models have better learning ability for complex and nonlinear electric data compared with mathematical-statistical models. For example, SVR is proposed based on the structural risk minimization criterion, which can provide promising prediction accuracy without much computational cost. DBN is a deep learning network structure composed of multiple restricted Boltzmann machines so that it provides powerful nonlinear data processing capabilities. As a new single-layer incremental neural network, BLS is proposed by using enhancement nodes and feature nodes, which greatly reduce the cost of prediction and confer strong competitiveness [22]. LSTM is established by introducing a forgetting mechanism with different units to extract useful features in time series data, supplying more reliable and accurate prediction performance in STLF. Therefore, the LSTM method can also obtain outstanding performance [23, 24].

Although these machine learning models have been used in load forecasting and achieve satisfactory prediction accuracy, they also have certain shortcomings, such as excessive training time and overfitting [25, 26]. Furthermore, smart power grid development involving many influencing factors and new energies, leading to increasing uncertainty and volatility of the electrical charge. Therefore, a traditional single model cannot achieve promising prediction results.

To improve the prediction accuracy of a single model, ensemble hybrid models are proposed by combining several neural networks to optimize single models. In [27], deep neural network (DNN) and LSTM were rationally combined to perform the forecasting operation. In [28], researchers presented a network based on deep residual networks with a convolution structure to carry out STLF. He [29] et al. proposed a model based on least absolute shrinkage and a selection operator-quantile regression neural network for probability density forecasting. In [30], researchers proposed a wind power probability density forecasting method based on cubic spline interpolation and support vector quantile regression. In [31], researchers proposed a new forecasting method based on multi-order fuzzy time series, technical analysis, and a genetic algorithm.

On the other hand, the “divide and conquer” method has been combined in some hybrid models for predictive tasks. These hybrid models often introduce decomposition-based methods to decompose the predictive series into multiple components and then feed the components into the neural network to obtain the prediction results. The decomposition methods mainly include wavelet transform (WT) [32], Fourier transform (FT) [32, 33], empirical mode

decomposition (EMD) [34], singular spectrum analysis (SSA) [35–37], variational mode decomposition (VMD) [36–39], and empirical wavelet transform (EWT) [40]. For example, in [41], researchers rationally combined ANN and WT technology to improve the forecasting ability. Based on EMD and DBN, the EMD-DBN hybrid network for STLF was proposed in [42]. In [43], researchers proposed a hybrid incremental learning approach composed of discrete wavelet transform (DWT), EMD, and RVFL, which obtains promising results in STLF. By combining EWT, LSTM, and regularized extreme learning machine, Li [44] proposed a hybrid model for wind speed forecasting. In [45], the author proposed a novel ensemble method based on ensemble empirical mode decomposition, least absolute shrinkage, and a selection operator-quantile regression neural network for forecasting wind power. In [46], researchers proposed a novel decomposition-ensemble learning approach based on the complete ensemble empirical mode decomposition and stacking-ensemble learning based on machine learning algorithms to forecast wind energy. In [47], researchers combine bi-directional LSTM with WT, EMD, ensemble EMD, and EWT respectively to predict wind speed.

Although this type of hybrid model can achieve better prediction results, it also has the following shortcomings:

- 1) FT method is suitable for the exactly or approximately periodic time-dependent phenomena but performs poorly when series are non-stationary [48]. On the other hand, VMD is suitable for narrowband signals but performs poorly for wide-band non-stationary signals [37]. However, power load data series have non-stationary signal or wide-band signal characteristics. Therefore, FT and VMD have limited application in STLF tasks. SSA is a non-parametric technique based on the principles of multivariate, and few parameters are required to model the series under analysis. However, there are no general rules for the selection of parameters in SSA [48]. The decomposition accuracy of WT is linked with the parameters of the filter base function; however, improper parameter selection may degenerate the decomposition accuracy of WT [48]. Furthermore, EMD easily produces mode aliasing, resulting in the attenuation of accuracy [48]. Although the EWT method can provide high decomposition accuracy, it easily produces redundant components, resulting in a large computational cost.
- 2) For the regression task, the existing hybrid machine learning methods [27, 28] rarely consider the compromise between prediction accuracy and computation cost. Actually, it is a very challenging problem to design a hybrid machine learning method that can not only pro-

vide high prediction accuracy but also has a reasonable computational cost.

- 3) Most existing models rarely consider how to use the prediction residual to establish a residual neural network model [42–48]. In fact, the prediction residual includes the effective prediction information. By mining the prediction residual information, the prediction accuracy of the model can be effectively improved.

To solve the abovementioned problems, a novel decomposition-based ensemble model is proposed for the STLF task in this paper. The decomposition method can decompose the raw power load data into multiple components, effectively smoothing nonlinear and non-stationary power load sequence signals to obtain competitive predictive performance. Therefore, an optimized empirical wavelet transform (OEWt) method is developed by rationally combining the approximate entropy (APEN) method [49] with the EWT method to improve the decomposition accuracy and eliminate redundant components. The EWT is first introduced to smooth the original power load data, which can solve the problem of parameter selection in the WT method and mode aliasing in the EMD. Then, APEN is used to determine similar components obtained by EWT and perform merging, which can reduce redundant decomposition information. LSTM can extract more useful features in time series data and supply more reliable and accurate prediction performance. Furthermore, the BLS is a new method that uses less computational cost to obtain better prediction performance. Therefore, a new hybrid machine learning method [named master learner (ML)] is proposed by rationally combining the LSTM with BLS in series, which can not only provide high prediction accuracy but also has a reasonable computational cost. In the master learner, a residual learning model [named residual learner (RL)] is developed to extract the effective predictive information from residual results. To further enhance the robustness of the proposed model, another BLS [named auxiliary learner (AL)] is adopted to connect the input and output of the proposed hybrid model. Specifically, the prediction result of ML will be fed into the input layer of AL, which means that the ML will expand the training data. Therefore, by rationally combining ML and AL, the proposed hybrid model can obtain outstanding prediction results.

In summary, the main contributions of our work are as follows:

- 1) The OEWt method is proposed by rationally combining the EWT with APEN to improve the decomposition accuracy and eliminate redundant components.
- 2) A new hybrid machine learning method (named master learner) is proposed by rationally combining the LSTM with BLS in series, which can effectively provide the prediction without significantly increasing the computational cost.
- 3) The residual learning model (named residual learner) is developed to extract the effective predictive information from residual results, which can further improve predictive accuracy.
- 4) An auxiliary learner is established based on the master learner, which can rationally connect the input layer and output layer of our prediction block to enhance the robustness and prediction accuracy of the model.
- 5) A novel decomposition-based ensemble model was first proposed for the STLF task by rationally combining the OEWt method, master learner, residual learner, and auxiliary learner, which can provide high predictive accuracy and satisfactory robustness in STLF.

The rest of the paper is described according to the following structure. In Section 2, the framework and related theories of the proposed model are introduced. Section 3 is data analysis and parameter settings. Section 4 is the case study with discussions. Finally, a conclusion is given in Section 5.

2 The forecasting framework

The framework of the proposed model is divided into two parts: the decomposition block and the forecast block. Figure 1 shows the framework of the proposed model. First, the power load data are decomposed into multiple subseries data points by OEWt. Second, each subseries data point is trained and predicted by the forecast block. Specifically, the forecast block is divided into ML and AL. In the ML, the preliminary prediction results are obtained by using the LSTM. Then, the prediction residuals caused by the LSTM are sent to the RL for learning. Here, the RL is the BLS. The preliminary prediction results of the LSTM and the residual learning results of the RL are superimposed as the prediction result of the ML. In the AL, the prediction results of the ML are recombined with the subseries data as the input of the AL to obtain the final predicted result of each subseries data. Finally, by combining the prediction results of each subseries data set, the final prediction result of the original data can be obtained.

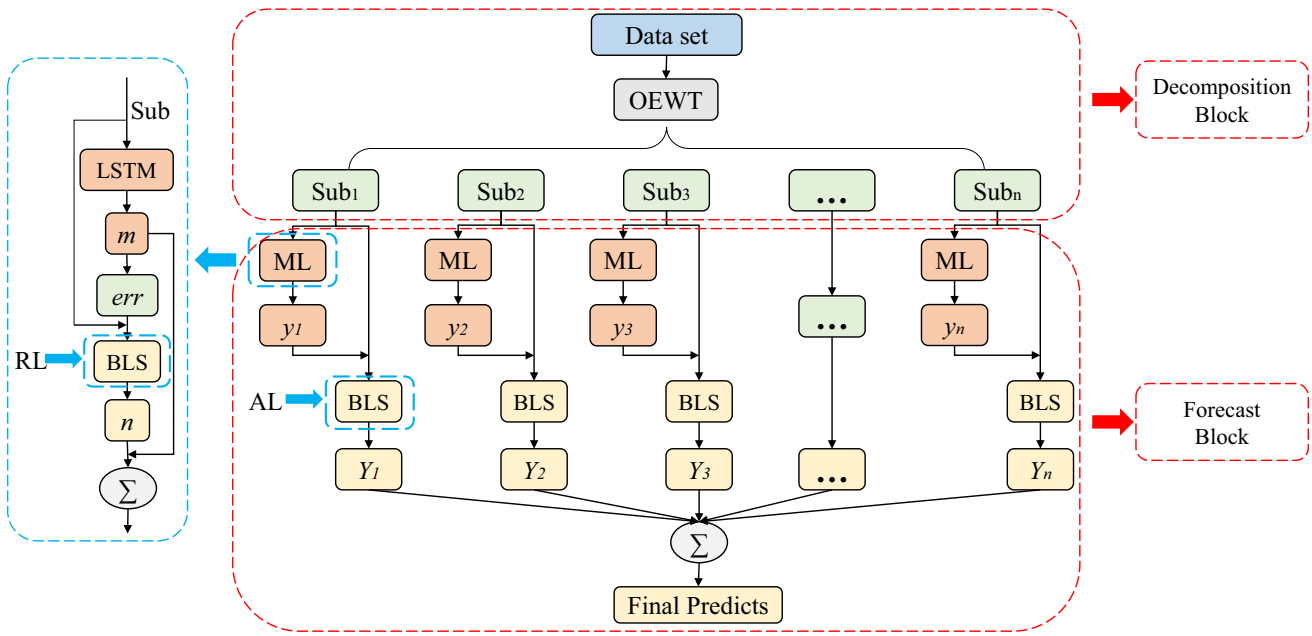


Fig. 1 The proposed short-term load forecasting framework consists of the decomposition block and forecast block

2.1 Decomposition Block

Due to the high uncertainty and volatility of the power load data, the EWT divides the power load data into multiple components to smooth the power load data. The most significant advantage of this method is that it can decompose signals adaptively, and its fundamental idea is to obtain the intrinsic mode of the signal by devising a proper wavelet filter bank. However, although the EWT method can provide high decomposition accuracy, it easily produces redundant components, resulting in a large computational cost. To guarantee decomposition accuracy and eliminate redundant components to reduce the computational cost, OEWT is proposed by rationally combining the APEN and the EWT method. Here, APEN is a nonlinear dynamic parameter that is used to quantify the regularity and volatility of time series fluctuations, which can effectively reduce the redundant components of EWT. Specifically, the decomposition steps of the OEWT are as follows:

Step 1: Adaptive spectrum division. First, the Fourier spectrum of the time series $g(t)$ is normalized to $[0, \pi]$. Second, the sequence $\{M_i\}_{k=1}^M$, which is composed of the maximum value M of the Fourier spectrum and regularized to $[0, 1]$, is recorded and rearranged according to the magnitude $M_1 \geq M_2 \geq \dots \geq M_M$. Then, to decide the number of components N , the threshold value $M_M + \kappa(M_1 - M_M)$ is set, where $\kappa \in (0, 1)$ is the relative amplitude ratio. N is set to the number of maxima greater than the threshold in $\{M_i\}_{k=1}^M$. Finally, ω_n is set to be the midpoint of the corresponding frequencies of the two adjacent maximum values above the

threshold, where $n = 1, 2, \dots, N - 1$, $\omega_0 = 0$, and $\omega_N = \pi$. With each ω_n as the centre, a transition phase $2\tau_n$ is defined, where $\tau_n = \chi\omega_n$, $0 < \chi < 1$.

$$\hat{T}_n(\omega) = \begin{cases} 1, & \text{if } |\omega| \leq (1 - \nu)\omega_n \\ \cos\left[\frac{\pi}{2}\beta\left(\frac{1}{2\nu\omega_n}(|\omega| - (1 - \nu)\omega_n)\right)\right], & \text{if } (1 - \nu)\omega_n \leq |\omega| \leq (1 + \nu)\omega_n \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$\hat{P}_n(\omega) = \begin{cases} 1, & \text{if } (1 + \nu)\omega_n \leq |\omega| \leq (1 - \nu)\omega_{n+1} \\ \cos\left[\frac{\pi}{2}\beta\left(\frac{1}{2\nu\omega_{n+1}}(|\omega| - (1 - \nu)\omega_{n+1})\right)\right], & \text{if } (1 - \nu)\omega_{n+1} \leq |\omega| \leq (1 + \nu)\omega_{n+1} \\ \sin\left[\frac{\pi}{2}\beta\left(\frac{1}{2\nu\omega_n}(|\omega| - (1 - \nu)\omega_n)\right)\right], & \text{if } (1 - \nu)\omega_n \leq |\omega| \leq (1 + \nu)\omega_n \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$\beta(x) = \begin{cases} 0, & \text{if } x < 0 \\ \text{and } \beta(x) + \beta(1 - x) = 1 \forall x \in [0, 1] \\ 1, & \text{if } x \geq 1 \end{cases} \quad (3)$$

$$\beta(x) = x^4(35 - 84x + 70x^2 - 20x^3) \quad (4)$$

Step 2: Construct the wavelet function and scaling function. By using the construction methods of the Littlewood-Paley and Meyer wavelets, the scaling function $\hat{T}_n(\omega)$

and the wavelet function $\widehat{P}_n(\omega)$ are constructed, which are denoted as formula (1) and formula (2), respectively. In these formulas, $\nu < \min_n \{(\omega_{n+1} - \omega_n) / (\omega_{n+1} + \omega_n)\}$, the function $\beta(x)$ is an arbitrary function that satisfies $C^k([0,1])$, as shown in formula (3). Many functions satisfy this property, and the most commonly used function is (4) [50].

Step 3: Empirical wavelet transform. The detailed coefficient $K_g^\epsilon(n, t)$ and the approximate coefficient $K_g^\epsilon(0, t)$ can be obtained by the inner product of $g(t)$ with $\widehat{T}_n(\omega)$ and $\widehat{P}_n(\omega)$, respectively. The calculation process is shown in formulas (5) and (6), respectively.

$$K_g^\epsilon(n, t) = \langle g, P_n \rangle = \int g(\tau) \overline{P_n(\tau - t)} d\tau = F^{-1} \left(G(\omega) \overline{\widehat{P}_n(\omega)} \right) \tag{5}$$

$$K_g^\epsilon(0, t) = \langle g, T_1 \rangle = \int g(\tau) \overline{T_1(\tau - t)} d\tau = F^{-1} \left(G(\omega) \overline{\widehat{T}_1(\omega)} \right) \tag{6}$$

where $\overline{P_n}$ and $\overline{T_1}$ represent the complex conjugates of P_n and T_1 , respectively. $G(\omega)$ represents the Fourier transform of $g(t)$. $F^{-1}(\bullet)$ represents the inverse Fourier transform. By convolution operations of $K_g^\epsilon(0, t)$ and $T_1(t)$, $K_g^\epsilon(n, t)$ and $P_n(t)$, respectively, the components $e_0(t)$ and $e_n(t)$ can be obtained. As shown in formulas (7) and (8).

$$e_0(t) = K_g^\epsilon(0, t) * T_1(t) \tag{7}$$

$$e_n(t) = K_g^\epsilon(n, t) * P_n(t) \tag{8}$$

Step 4: Eliminate redundant components by APEN. First, the algorithm-related parameters m and r are defined, where m is an integer and represents the length of the comparison vector, and r is a real number, which represents a measure of similarity. Usually, choose the parameter $m = 2$ or $m = 3$, $r = 0.2 \times std$ (std is the standard deviation of the time series). Then, reconstruct the m -dimensional vector $E_n(1), E_n(2), \dots, E_n(T - m + 1)$ as shown in formula (9).

$$E_n(M) = [e_n(M), e_n(M + 1), \dots, e_n(M + m + 1)], \tag{9}$$

$$M = 1, 2, \dots, T - m + 1$$

For $1 \leq M \leq T - m + 1$, count the number of vectors that meet the conditions $C_M^m(r) = (\text{number of } E_n(j) \text{ such that } d[E_n(M), E_n(j)] \leq r) / (T - m + 1)$, where $d[E_n, E_n^*]$ is defined as $d[E_n, E_n^*] = \max_M |e_n(a) - e_n^*(a)|$. $e_n(a)$ is the element of the vector $E_n(M)$. d represents the distance between vectors $E_n(M)$ and $E_n(j)$, which is determined by the maximum difference of the corresponding elements. The value range of j is $[1, T - m + 1]$, including $M = j$. By defining $\psi^m(r)$ as formula (10), APEN is expressed as

$APEN = \psi^m(r) - \psi^{m+1}(r)$. Finally, APEN values of all components are calculated. Based on the APEN of each component, an interval ρ is defined to reorganize the components that are in the same interval into a new subseries as the output of the OEWT.

$$\psi^m(r) = (T - m + 1)^{-1} \sum_{M=1}^{T-m+1} \log(C_M^m(r)) \tag{10}$$

2.2 Forecast block

The forecast block consists of two parts: the ML and the AL. The ML can provide a compromise between prediction accuracy and computation cost. The AL can improve the prediction accuracy and robustness of the model. The specific description of the ML and AL is given as follows:

1) Master Learner

The existing hybrid machine learning methods rarely consider the balance between prediction accuracy and computation cost. In addition, the prediction residual includes the effective prediction information. To balance the prediction accuracy and computational cost, and mine the prediction residual information, the ML is designed by mixing the LSTM and BLS, and the mixed model is named BLSTM. In this model, LSTM is used for a preliminary prediction. Then, the prediction residuals caused by the LSTM are sent to the RL for learning. Here, the RL is the BLS. Finally, the preliminary prediction results of the LSTM and the residual learning results of the RL are superimposed as the prediction result of the ML. Specifically, the design process is given as follows:

First, LSTM is used to make a preliminary prediction. Let $\{x_1, x_2, \dots, x_T\}$ denote a typical input sequence for an LSTM, where $x_t \in R^k$ represents a k -dimensional vector of real values at the t time step. To establish temporal connections, the LSTM defines and maintains an internal memory cell state throughout the whole life cycle, which is the most important element of the LSTM structure. The memory cell state s_{t-1} interacts with the intermediate output h_{t-1} and the subsequent input x_t to determine which elements of the internal state vector should be updated, maintained, or erased based on the outputs of the previous time step and the inputs of the present time step. In addition to the internal state, the LSTM structure also defines input node g_t , input gate i_t , forget gate f_t , and output gate o_t . The formulations of all nodes in an LSTM structure are given by formulas (11) to (16).

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \tag{11}$$

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \tag{12}$$

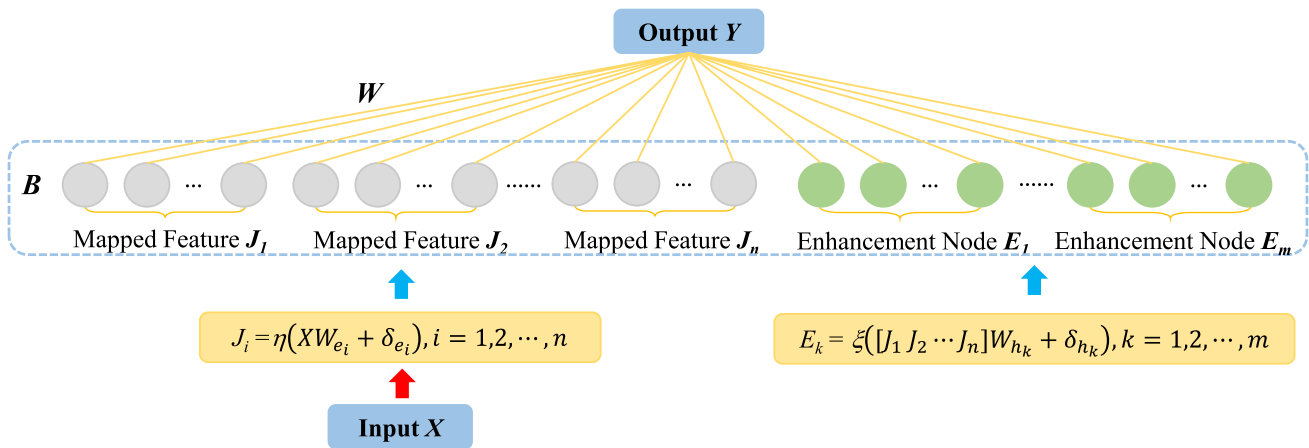


Fig. 2 The framework of the broad learning system

$$g_t = \phi(W_{gx}x_t + W_{gh}h_{t-1} + b_g) \tag{13}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \tag{14}$$

$$s_t = g_t i_t + s_{t-1} f_t \tag{15}$$

$$h_t = \phi(s_t) o_t \tag{16}$$

where $W_{gx}, W_{gh}, W_{ix}, W_{ih}, W_{fx}, W_{fh}, W_{ox},$ and W_{oh} are weight matrices for the corresponding inputs of the network activation functions; σ represents the sigmoid activation function, while ϕ represents the tanh function.

Then, the prediction residuals caused by LSTM are learned by the RL. Here, BLS can use less computational cost to obtain better prediction performance. Therefore, BLS was selected as the RL. More details about BLS will be given in the AL.

Finally, by superimposing the prediction result of LSTM and the prediction result of RL, the final prediction result of ML can be obtained.

2) Auxiliary Learner

By using the prediction results of ML to expand the training data, the prediction accuracy and robustness of the model can be improved. To achieve this goal, the AL is introduced.

In the AL, the output of the ML and the original subseries data is reorganized as the input of the AL, which can be considered feedback. Therefore, through the proposed feedback, the connection between the ML and the AL is realized. Another BLS is chosen as the machine learning model in the AL. The structure of BLS is shown in Fig. 2. Specifically, the theory of BLS can be concluded as follows:

Let the input data X form n feature nodes J_i through feature mapping, as shown in formula (17). All feature nodes

are combined and defined as $J^n = [J_1, J_2, \dots, J_n]$. Then, m enhancement nodes E_k are acquired by enhancing and transforming with J^n , as shown in formula (18). In formulas (17) and (18), $\eta(\bullet)$ is a linear transformation by default, and $\xi(\bullet)$ is a nonlinear activation function. Generally, the hyperbolic tangent function in formula (19) can be selected as the activation function. $W_{e_i}, W_{h_k}, \delta_{e_i},$ and δ_{h_k} are randomly generated weight matrices and bias matrices, which are fine-tuned by a sparse encoder [18]. All enhancement nodes are combined and defined as $E^m = [E_1, E_2, \dots, E_m]$. The symbol B is introduced for the convenience of representation, expressed as $B = [J^n | E^m]$.

$$J_i = \eta(XW_{e_i} + \delta_{e_i}), i = 1, 2, \dots, n \tag{17}$$

$$E_k = \xi(J^n W_{h_k} + \delta_{h_k}), k = 1, 2, \dots, m \tag{18}$$

$$\xi(x) = \tanh x \tag{19}$$

From Fig. 2, the final prediction value of BLS \hat{Y} can be expressed as $\hat{Y} = BW$. Here, W represents the weight matrix between the feature nodes, enhancement nodes, and output Y . Since $W_{e_i}, \delta_{e_i}, W_{h_k},$ and δ_{h_k} are randomly generated and fine-tuned by the sparse encoder, they remain unchanged. Moreover, the actual value Y is known when training. So

$$W = B^+ Y \tag{20}$$

, where B^+ is the pseudo-inverse of B . Ridge regression is used to find a suitable W to transform the above problem into

$$\arg \min_W \left(\| \hat{Y} - Y \|^2 + \lambda \| W \|^2 \right). \tag{21}$$

Here, λ is the regularization parameter; when $\lambda \rightarrow 0$, the solution is

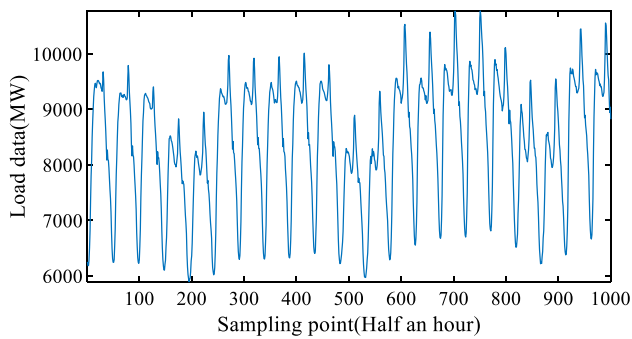


Fig. 3 Load data of New South Wales, Australia in 2009

Table 1 The statistical data of the historical load data of NSW

	Max	Min	Mean	Std
NSW	14274.15	5498.36	8894.00	1409.05

$$W = (\lambda I + BB^T)^{-1} B^T Y. \quad (22)$$

Where I is the identity matrix. Specifically, we have that

$$B^+ = \lim_{\lambda \rightarrow 0} \left((\lambda I + BB^T)^{-1} B^T \right). \quad (23)$$

Since the subseries data are decomposed from the original power load data through the decomposition block, the final prediction result of the model can be obtained by superimposing the prediction results of all subseries data.

3 Data analysis and parameter settings

3.1 Data sets

The data set, which is used as the experimental sample in this paper, is the historical load data of New South Wales (NSW), Australia, in 2009. The sampling interval of the data set is 30 min, so 48 load data samples are contained in one day. Figure 3 shows the data of 1,000 sampling points. From the figure, we can find that the load data fluctuate greatly at the peak. Table 1 shows the statistical data of this data set, including the maximum, minimum, average, and standard deviation. To avoid the negative impact of singular sample data on the prediction accuracy, the load data are normalized and restricted to the range of $[0,1]$ before the experiment. The normalization formula is shown in (24).

$$\tilde{y}_i = \frac{y_{max} - y_i}{y_{max} - y_{min}} \quad (24)$$

Here, \tilde{y}_i is the normalized result, y_i is the load data at a certain moment, and y_{max} and y_{min} are the maximum and minimum values in the load data, respectively.

3.2 Performance estimation

The accuracy of the prediction result needs to be evaluated by the evaluation function. This paper uses common evaluation methods in load forecasting to assess the prediction performance, including the root mean square error (RMSE) and the mean absolute error (MAE). They are defined in formulas (25) and (26).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (25)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (26)$$

where \hat{y}_i is the predicted data, y_i is the real data, and n is the total number of test samples. For both evaluation indicators, the smaller the value is, the higher the accuracy of the prediction.

3.3 Parameter settings

The parameter settings are significant for the prediction accuracy of the model. Before the formal experiment, some pre-experiments are conducted to filter out the best parameters. In the forecast block, the number of hidden nodes and the learning rate of LSTM have a more significant impact on the prediction performance. Furthermore, the number of feature nodes and enhancement nodes of BLS are essential parameters that affect its prediction accuracy. To find the optimal parameters in LSTM and BLS, the controlled variable method is adopted. According to experience, we conducted pre-experiments on different hidden nodes ($[100, 300]$ with an interval of 10) and learning rate ($[0.001, 0.01]$ with a gap of 0.001) of LSTM. Moreover, pre-experiments with different feature nodes and enhancement nodes (both in $[1, 30]$ with one as the interval) of BLS are conducted. Meanwhile, RMSE is selected as the evaluation index. The pre-experimental results are shown in Fig. 4.

From Fig. 4(a), the results show that when the learning rate of LSTM varies from 0.006 to 0.01, its prediction performance is unstable. In this interval, the hidden nodes dominate in prediction performance. However, when the learning rate varies from 0.005 to 0.001, the prediction performance remains relatively stable, and the hidden nodes also have little effect on prediction performance. Meanwhile, from Fig. 4(b), the results show that the number of enhancement nodes in BLS will not significantly impact its prediction

Fig. 4 RMSE performance of the LSTM and BLS pre-experiment

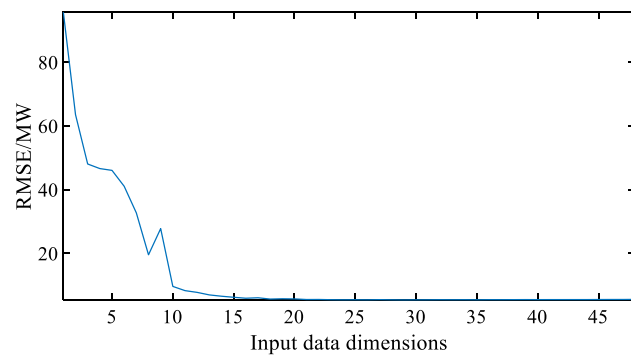
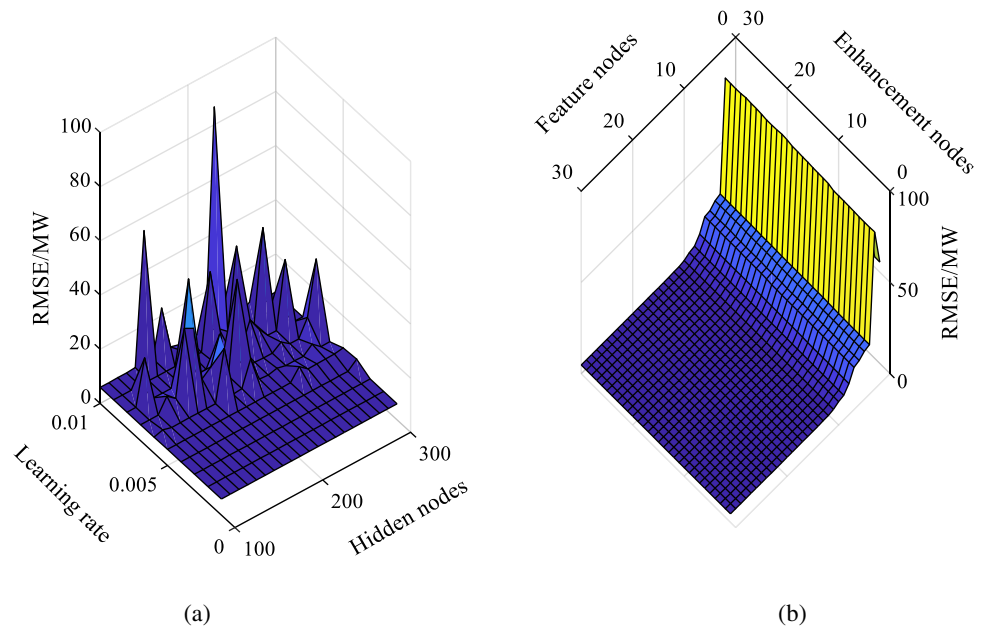


Fig. 5 RMSE performance of the input data dimensions pre-experiment

performance. For the feature nodes, when the number of feature nodes is within 10, the RMSE of the prediction results shows a rising trend and then falls. However, when the number of feature nodes increases to 10, the performance of the model does not change much while the number of feature nodes increasing.

There is a specific connection between the load value at a certain moment and the load value before that moment. How long this correlation performance lasts is a question worthy of discussion. Therefore, we conducted a pre-experiment on the parameter of input data dimension in [1, 48] (0.5 to 24 h), and the experimental results are shown in Fig. 5. The pre-experiment result shows that when the input data dimension increases from 1 to 15 (0.5 to 7.5 h), the prediction performance of the model gradually improves. However, when the input data dimension is greater than 15 (after 7.5 h), the prediction performance of the model does not significantly

improve with the increase of the input data dimension. Therefore, we can conclude that the load value at a certain moment has a strong correlation with the load value within 7.5 h before that moment.

Based on the pre-experiment results, we set the number of hidden nodes and learning rate of LSTM to 200 and 0.005, respectively, and the number of feature nodes and enhancement nodes of BLS to 24 and 15, respectively. The dimension of the input data of the forecast block is set to 48 (24 h). In the OEWT, the parameter of comparison vector length m is set to 2, the similarity measure r is set to $0.2 \times std$, and the interval ρ is set to 0.1.

To confirm the performance of the proposed model, other methods were compared with our model. Moreover, a series of experiments were also performed to determine the optimal parameters of these methods. Each model in different parameter values was run 20 times, and the average RMSE was used as the evaluation index. Due to space limitations, more specific experimental results have been provided in the Supplementary File. As a result, the optimal parameters of each model are selected and shown in Table 2.

3.4 Method for model assessment

To verify the stability of our method and avoid overfitting, K-fold cross-validation is used to test the model. The data set is divided into the training set, validation set, and test set. The training set is used to train forecasting models, the validation set is used to select the best performing models, and the testing set for result evaluation.

Table 2 The parameter settings of the comparison method

Model	Optimal parameters
ANN	$n_h = 200, m_i = 10, eta = 0.01$
DBN	$n_h = 10, eta = 0.001, r_b = 1, a_f = Sigmoid, v_m = 0.01, m_i = 20$
RBFNN	$f_{RBF} = Gaussian, s_{RBF} = 50$
RVFL	$n_e = 10,000, a_f = Sigmoid, DL=true, r_m = uniform$
EMD-BLS [22]	$n_f = 24, n_e = 15$
SWT-LSTM [24]	$n_h = 200, eta = 0.01$
DWT-EMD-RVFL [43]	$n_e = 10,000, a_f = Sigmoid, DL=true, r_m = Gaussian$
EMD-DBN [42]	$n_h = [100100], eta = 0.001, a_f = Sigmoid, r_b = 2, v_m = 0.01, m_i = 500$

n_h -the number of hidden nodes; m_i - the maximum number of iterations; eta -learning rate; a_f -activation function; r_b -the random batch size of each time; v_m -momentum value; f_{RBF} - radial basis functions; s_{RBF} -the spread of radial basis functions; n_e -the number of enhancement nodes; DL -whether to have the direct link between the input layer and output layer; r_m -randomization methods; n_f -the number of feature nodes

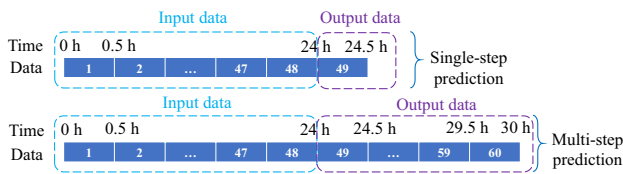


Fig. 6 The difference between single-step prediction and multi-step prediction

Furthermore, to compare the differences between our model and other models, statistical analysis is introduced. A multicomparison statistical procedure is first applied to test the null hypothesis that all learning algorithms obtained the same results on average. Specifically, we used the Friedman test [51, 52]. If the Friedman test rejected the null hypothesis, post hoc tests were applied. Here, the post hoc test is the Nemenyi test. If the corresponding average rank of two models differs by at least a critical distance CD, then we speculate that there are obvious differences between the two methods. The calculation of CD is shown in formula (27), where n_l is the number of learning algorithms, n_{ds} is the number of data sets, and q_α is the critical value based on the Studentized Range statistic [53].

$$CD = q_\alpha \sqrt{\frac{n_l(n_l + 1)}{6n_{ds}}} \tag{27}$$

4 Case study

According to the parameters set in Section 3, we divide the 2009 data set into four parts according to the seasons, namely, spring, summer, autumn, and winter. Therefore, the influence of seasons can be eliminated. Note that each seasonal data set includes 3 months. The data of the previous

two months are used as the training set and validation set. Specifically, K-fold cross-validation is used to divide the training set and the validation set. The data of the following month are used as the test set for formal experiments. In addition, because power plants often need to allocate power loads in advance, a multi-step forecasting experiment 6 h ahead is performed to test the performance of the model in multistep prediction. Figure 6 shows the difference between single-step prediction and multi-step prediction. Specifically, the data from 1 to 48 is used to predict the data of 49 in single-step prediction; the data from 1 to 48 is used to predict the data from 49 to 60 in multi-step prediction.

To verify the effectiveness of our model, it is compared with the state-of-the-art machine learning methods and hybrid models, including the ANN, RBFNN, DBN, RVFL, EMD-DBN, SWT-LSTM, DWT-EMD-RVFL, and EMD-BLS. Each compared method was trained and tested on a personal computer 64-bit operating system, 8.00 GB, RAM, Intel(R) Core (TM) i5-7300HQ, CPU@2.50. The forecasting results of each model are shown in Table 3, where the first and second-best predictive results from the compared methods are emphasized in bold text and italic, respectively.

4.1 Single-step prediction

In the single-step prediction experiment, the prediction duration is 0.5 h. Interestingly, via the two evaluation criteria, we can observe that our proposed model is predominant over the others in each season, as shown in Table 3. Furthermore, EMD-BLS obtains the second-best predictive accuracy in each season, but the prediction performance of EMD-BLS is still much worse than that of our model.

Table 3 Forecast errors of each model in single-step and multi-step prediction

Forecast length	Data set	Metrics	EMD-BLS	SWT-LSTM	DWT-EMD-RVFL	EMD-DBN	ANN	RBFNN	DBN	RVFL	Our model	
0.5 h	Spring	RMSE	43.27	111.28	113.84	128.84	133.89	67.18	115.47	117.62	7.65	
		MAE	33.93	83.92	85.04	104.44	109.78	51.72	85.54	85.83	6.13	
	Summer	RMSE	79.20	125.11	113.69	127.13	167.88	81.98	150.69	118.97	8.04	
		MAE	58.89	89.78	84.04	94.44	134.17	63.34	110.6	88.12	6.40	
	Autumn	RMSE	54.67	116.44	104.65	124.65	144.05	76.61	128.04	117.41	5.97	
		MAE	42.86	88.27	76.61	96.54	115.38	57.98	96.86	84.13	4.82	
	Winter	RMSE	39.99	123.50	124.77	146.71	136.81	81.16	121.06	106.06	5.07	
		MAE	30.55	94.13	91.67	115.17	112.46	61.56	88.72	79.02	4.01	
	6 h	Spring	RMSE	454.27	313.48	325.25	401.85	384.48	195.48	250.62	185.73	49.53
			MAE	300.65	225.49	243.37	290.58	312.24	149.65	180.72	134.82	28.36
		Summer	RMSE	410.34	274.49	320.06	225.90	462.01	231.75	318.48	262.69	52.69
			MAE	319.99	203.43	222.17	168.24	359.07	170.03	240.57	178.65	31.58
Autumn		RMSE	338.15	369.19	333.83	257.45	493.84	249.17	384.90	178.95	24.66	
		MAE	253.79	283.15	232.39	204.42	343.14	162.77	295.00	125.06	16.31	
Winter		RMSE	202.12	327.43	325.25	401.85	344.17	139.12	246.41	165.23	37.98	
		MAE	151.50	233.09	243.37	290.58	277.93	110.5	184.67	127.36	24.46	

4.2 Multi-step prediction

To explore the performance of our model on multi-step prediction, we conducted experiments 6 h ahead of the forecast. It can be seen from Table 3 that as the forecast length increases, the RMSE and MAE values of all models increase, which indicates that the accuracy of the prediction will gradually decrease with the increase of the forecast length, as well as our model. However, interestingly, it can be seen from Table 3 that our model still has the best predictive performance compared with the eight others models. Furthermore, RVFL obtains the second-best predictive accuracy in spring and autumn. EMD-DBN and RBFNN obtain the second-best predictive accuracy in summer and winter, respectively.

4.3 Model assessment

The experimental results of K-fold cross-validation are shown in Table 4. In the experiment, the fold number K was set to 8, and each fold was repeated 5 times. From Table 4, it can be seen that the difference between the validation-RMSE and the test-RMSE is small, which indicates the reliable convergence and stability of our model.

To compare the differences between the algorithms in Table 3, statistical analysis is conducted. Since 8-fold cross-validation is performed in each season, and each fold is repeated 5 times, the number of data sets n_{ds} is 160. Furthermore, the number of learning algorithms n_l is 9. By calculation, the Friedman test rejected the null hypothesis that all nine learning algorithms performed the same on average. Therefore, we applied the post hoc Nemenyi test at $\alpha = 0.1$ to test the difference between the algorithms. In this condition, the value of CD is 0.87. The Friedman diagram is shown in Fig. 7. In the Friedman diagram, if the two models have no overlapping area, it proves that there are obvious differences between the two models. Figure 7 shows that our model has obvious differences from other models, except EMD-BLS.

4.4 Effect of OEWT, master learner (ML), Residual Learner (RL), and Auxiliary Learner (AL)

To evaluate the effect of OEWT, ML, RL, and AL, we compared our model (OEWT-BLSTM-BLS) with EWT-BLSTM-BLS, BLSTM-BLS, OEWT-LSTM-BLS, and OEWT-BLSTM. All models are used to forecast the load

Table 4 The experimental results of K-fold cross-validation

	Spring	Summer	Autumn	Winter
Train-RMSE	5.35	5.34	5.34	6.20
Validation-RMSE	5.31	5.35	5.41	6.17
Test-RMSE	6.81	7.52	6.02	6.88

data of the four seasons in single steps and multi-steps. The forecasting results and running time are shown in Table 5.

1) Effect of OEWT

To verify the effectiveness of OEWT, we first compared OEWT-BLSTM-BLS with BLSTM-BLS. The difference between OEWT-BLSTM-BLS and BLSTM-BLS is that the former contains OEWT, and the latter does not contain OEWT. From Table 5, the experimental results show that OEWT-BLSTM-BLS has better prediction performance than BLSTM-BLS. This indicates that the OEWT can effectively smooth nonlinear and non-stationary power load sequence signals to obtain competitive predictive performance.

To verify the performance of OEWT and EWT, we compared EWT-BLSTM-BLS with OEWT-BLSTM-BLS. The difference between EWT-BLSTM-BLS and OEWT-BLSTM-BLS is that the former only uses a single EWT, and the latter contains EWT and APEN. From Table 5, the experimental results show that OEWT-BLSTM-BLS has better prediction performance than EWT-BLSTM-BLS in most cases.

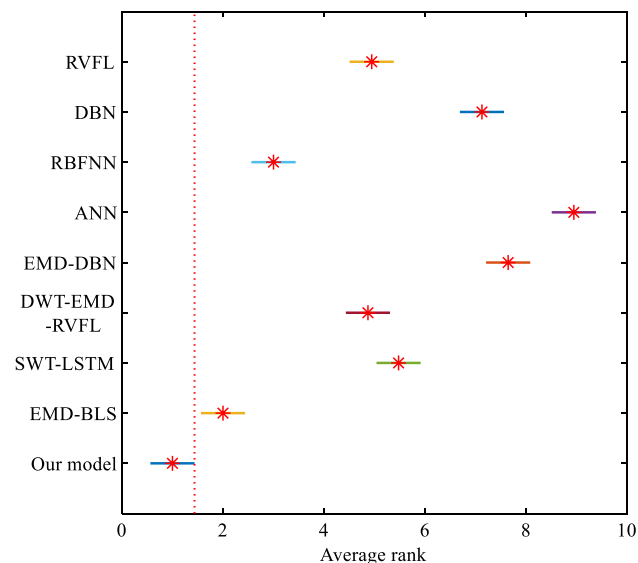


Fig. 7 The results of the Friedman test (If the two models have no overlapping area, it proves that there are obvious differences between the two models)

Furthermore, OEWT-BLSTM-BLS has a significantly lower computation time than EWT-BLSTM-BLS. This indicates that OEWT not only guarantees the compromise in prediction accuracy but also significantly reduces the computational cost.

2) Effect of Master Learner and Residual Learner

To verify the performance of the master learner (BLSTM) and the residual learner (RL), we compare OEWT-BLSTM-BLS with OEWT-LSTM-BLS. The difference between OEWT-BLSTM-BLS and OEWT-LSTM-BLS is that the former has an additional residual learner BLS in BLSTM, and the latter does not contain this learner. From Table 5, the experimental results show that OEWT-BLSTM-BLS has better prediction performance than OEWT-LSTM-BLS. This indicates that by introducing the RL into LSTM, the master learner (BLSTM) has better performance than LSTM; the also indicates the RL can further improve the prediction accuracy by extracting the effective predictive information from residual results.

3) Effect of Auxiliary Learner

To verify the performance of the AL, we compared OEWT-BLSTM-BLS with OEWT-BLSTM. Here, the AL is BLS. The difference between OEWT-BLSTM-BLS and OEWT-BLSTM is that the former contains an AL, and the latter does not. From Table 5, the experimental results show that OEWT-BLSTM-BLS has better prediction performance than OEWT-BLSTM. This indicates that the AL can also further improve the prediction accuracy.

4.5 Discussion

The above experimental results indicate our model can not only effectively obtain better performance but also provide promising robustness on the STLF task. The reason behind this fact is that the proposed decomposition-based ensemble model rationally combines the OEWT, master learner, residual learner, and auxiliary learner. Specifically, (1) OEWT is developed to decompose the power load data into multiple sub-time series, which can effectively smooth nonlinear and non-stationary electric loads and eliminate redundant decomposition components that lead to an increase in computational cost. (2) the master learner integrates the advantages of LSTM and BLS, which can effectively compromise the computation cost and the prediction accuracy. (3) the residual learner is developed to learn the prediction residuals of LSTM, which can mine the effective prediction information hidden in the prediction residual to improve the prediction accuracy. (4) the auxiliary learner is established

Table 5 Forecast errors of each model in single-step and multi-step prediction

Forecast length	Data set	Metrics	OEWT-BLSTM-BLS (Our model)	EWT-BLSTM-BLS	BLSTM-BLS	OEWT-BLSTM-BLS	OEWT-BLSTM-BLS
0.5 h	Spring	RMSE	7.65	6.76	86.63	16.70	28.13
		MAE	6.13	5.42	67.00	12.44	21.84
	Summer	RMSE	8.04	7.23	110.86	20.58	28.85
		MAE	6.40	5.84	82.06	15.23	17.74
	Autumn	RMSE	5.97	5.97	107.61	16.02	21.07
		MAE	4.82	4.81	82.55	7.70	14.23
	Winter	RMSE	5.07	7.03	94.08	11.23	20.14
		MAE	4.01	5.67	71.25	7.95	15.38
6 h	Spring	RMSE	49.53	209.26	383.28	68.84	184.94
		MAE	28.36	165.00	278.83	50.30	111.99
	Summer	RMSE	52.69	217.67	422.41	67.97	195.64
		MAE	31.58	162.04	312.33	46.86	129.84
	Autumn	RMSE	24.66	250.80	486.89	50.06	209.65
		MAE	16.31	194.56	383.98	37.31	134.46
	Winter	RMSE	37.98	123.63	358.97	70.04	227.55
		MAE	24.46	92.07	301.68	54.26	144.72
Running time / s			481.34	1606.31	74.34	334.32	233.80

to rationally connect the input layer and output layer of our prediction block, further improving prediction accuracy and robustness.

5 Conclusions

The problem of parameter selection in WT or mode aliasing in EMD may result in the attenuation of decomposition accuracy. Although the EWT method can provide high decomposition accuracy, it easily produces redundant components, resulting in a large computational cost. Furthermore, the prediction residuals include effective prediction information. However, most existing models rarely consider how to use the prediction residual to establish a residual learning model. In addition, most existing hybrid machine learning methods rarely consider the compromise between prediction accuracy and computational cost. To overcome the above issues, this paper proposes a novel decomposition-based ensemble model including OEWT, master learner, residual learner, and auxiliary learner for STLF tasks. Experimental results show that the proposed model not only has high predictive accuracy and robustness but also low computational cost.

In the future, the proposed decomposition-based ensemble model plans to be applied to other predictive tasks such as wind speed, photovoltaics, 5G base station flow, and traffic flow.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grants 61863028, 81660299, and 61503177, and in part by the Science and Technology Department of Jiangxi Province of China under Grants 20161ACB21007, 20171BBE50071, and 20171BAB202033.

References

- Hong T, J. I. J. o. S, Fan F (2016) Probabilistic electric load forecasting: A tutorial review. *Int J Forecast* 32(3):914–938
- Bessani M, Massignan J, Santos T et al (2020) Multiple households very short-term load forecasting using bayesian networks. *Electr Power Syst Res* 189:106733
- Sun JX, Wang JN, Yu WX et al (2020) Power load disaggregation of households with solar panels based on an improved long short-term memory network. *J Electr Eng Technol* 15(5):2401–2413
- Dudek G, Peka P (2021) Pattern similarity-based machine learning methods for mid-term load forecasting: a comparative study. *Appl Soft Comput* 104(2):107223
- Yin L, Xie J (2021) Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems. *Appl Energy* 283(6):116328
- Raza MQ, Mithulananthan N, Li J, Lee KY (2020) Multivariate ensemble forecast framework for demand prediction of anomalous days. *IEEE Trans Sustain Energy* 11(1):27–36
- Han L, Peng Y, Li Y et al (2018) Enhanced deep networks for short-term and medium-term load forecasting. *IEEE Access* 7:4045–4055
- Tang X, Dai Y, Wang T et al (2019) Short-term power load forecasting based on multi-layer bidirectional recurrent neural network. *IET Gener Transm Distrib* 13(17):3847–3854
- Khawaja AS, Zhang X, Anpalagan A et al (2017) Boosted neural networks for improved short-term electric load forecasting. *Electr Power Syst Res* 143:431–437

10. Dosiek L (2020) The effects of forced oscillation frequency estimation error on the LS-ARMAS mode meter. *IEEE Trans Power Syst* 35(2):1650–1652
11. Moon J, Hossain MB, Chon KH (2021) AR and ARMA model order selection for time-series modeling with ImageNet classification. *Sig Process* 183(10):108026
12. Ertuğrul ÖF, Tekin H, Tekin R (2021) A novel regression method in forecasting short-term grid electricity load in buildings that were connected to the smart grid. *Electr Eng* 103:717–728
13. Xu W, Peng H, Zeng X et al (2019) A hybrid modeling method for time series forecasting based on a linear regression model and deep learning. *Appl Intell* 49:3002–3015
14. Munawar U, Wang Z (2020) A framework of using machine learning approaches for short-term solar power forecasting. *J Electr Eng Technol* 15(2):561–569
15. Xu C, Gordan B, Koopialipoor M et al (2019) Improving performance of retaining walls under dynamic conditions developing an optimized ANN based on ant colony optimization technique. *IEEE Access* 7:94692–94700
16. Elattar EE, Sabiha NA, Alsharaf M et al (2020) Short term electric load forecasting using hybrid algorithm for smart cities. *Appl Intell* 50:3379–3399
17. Dedinec A, Filiposka S, Dedinec A et al (2016) Deep belief network based electricity load forecasting: An analysis of Macedonian case. *Energy* 115:1688–1700
18. Chen CLP, Liu Z (2018) Broad learning system: an effective and efficient incremental learning system without the need for deep architecture. *IEEE Trans Neural Netw Learn Syst* 29(1):10–24
19. Zhao F, Zeng GQ, Lu KD (2019) EnLSTM-WPEO: short-term traffic flow prediction by ensemble LSTM, NNCT weight integration and population extremal optimization. *IEEE Trans Veh Technol* 99:1–1
20. Le T, Vo B, Fujita H et al (2019) A fast and accurate approach for bankruptcy forecasting using squared logistics loss with GPU-based extreme gradient boosting. *Inf Sci* 494:294–310
21. Gang Shi C, Qin J, Tao C, Liu (2021) A VMD-EWT-LSTM-based multi-step prediction approach for shield tunneling machine cutterhead torque. *Knowl-Based Syst* 228:107213
22. Zhu L, Lian C (2019) Wind Speed forecasting based on a hybrid EMD-BLS method. 2019 Chinese Automation Congress (CAC), pp 2191–2195
23. Tan M, Yuan S, Li S et al (2020) Ultra-short-term industrial power demand forecasting using LSTM based hybrid ensemble learning. *IEEE Trans Power Syst* 35(4):2937–2948
24. Yan K, Li W, Ji Z et al (2019) A hybrid LSTM neural network for energy consumption forecasting of individual households. *IEEE Access* 7:1–1
25. Chen Y, Luh PB, Guan C et al (2010) Short-Term load forecasting: similar day-based wavelet neural networks. *IEEE Trans Power Syst* 25(1):322–330
26. Yang Y, Li W, Gulliver TA et al (2019) Bayesian deep learning based probabilistic load forecasting in smart grids. *IEEE Trans Industr Inf* 99:1–1
27. Ospina J, Newaz A, Faruque MO (2019) Forecasting of PV plant output using hybrid wavelet-based LSTM-DNN structure model. *IET Renew Power Gener* 13(7):1087–1095
28. Sheng Z, Wang H, Chen G et al (2021) Convolutional residual network to short-term load forecasting. *Appl Intell* 51:2485–2499
29. He Y, Yang Q, Wang S et al (2019) Electricity consumption probability density forecasting method based on LASSO-Quantile Regression Neural Network. *Appl Energy* 233–234:565–575
30. He Y, Li H, Wang S et al (2020) Uncertainty analysis of wind power probability density forecasting based on cubic spline interpolation and support vector quantile regression. *Neurocomputing* 430:121–137
31. Ye F, Zhang L, Zhang D et al (2016) A novel forecasting method based on multi-order fuzzy time series and technical analysis. *Inf Sci* 367–368:41–57
32. Md M, Alam S, Rehman LM, Al-Hadhrami JP, Meyer (2014) Extraction of the inherent nature of wind speed using wavelets and FFT. *Energy Sustain Dev* 22:34–47
33. Ujjwal Kumar K, De Ridder (2010) GARCH modelling in association with FFT-ARIMA to forecast ozone episodes. *Atmos Environ* 44(34):4252–4265
34. Huang NE, Shen Z, Long SR et al (1998) The empirical mode decomposition method and the Hilbert spectrum for non-stationary time series analysis. *Proc R Soc A: Math Phys Eng Sci* 454:903–995
35. Zhang X, Wang J (2018) A novel decomposition-ensemble model for forecasting short-term load-time series with multiple seasonal patterns. *Appl Soft Comput* 65:478–494
36. Moreno SR, Mariani VC, dos Santos Coelho L (2021) Hybrid multi-stage decomposition with parametric model applied to wind speed forecasting in Brazilian Northeast. *Renew Energy* 164:1508–1526
37. Moreno SR, Gomes Ramon, da Silva Viviana, Mariani Cocco, dos Santos Coelho Leandro (2020) Multi-step wind speed forecasting based on hybrid multi-stage decomposition model and long short-term memory neural network. *Energy Convers Manag* 213:112869
38. He F, Zhou J, Mo L, Feng K, Liu G (2020) Day-ahead short-term load probability density forecasting method with a decomposition-based quantile regression forest. *Appl Energy* 262:114396
39. Jatin Bedi D (2020) Energy load time-series forecast using decomposition and autoencoder integrated memory network. *Appl Soft Comput* 93:106390
40. Gilles J (2013) Empirical wavelet transform. *IEEE Trans Signal Process* 61(16):3999–4010
41. Salkuti SR (2018) Short-term electrical load forecasting using hybrid ANN-DE and wavelet transforms approach. *Electr Eng* 100:2755–2763
42. Qiu X, Ren Y, Suganthan PN et al (2017) Empirical Mode Decomposition based ensemble deep learning for load demand time series forecasting. *Appl Soft Comput* 54:246–255
43. Qiu X, Suganthan PN, Amaratunga GAJ (2018) Ensemble incremental learning Random Vector Functional Link network for short-term electric load forecasting. *Knowl-Based Syst* 145:182–196
44. Li Y, Wu H, Liu H (2018) Multi-step wind speed forecasting using EWT decomposition, LSTM principal computing, RELM subordinate computing and IEWT reconstruction. *Energy Convers Manag* 167:203–219
45. He Y, Wang Y (2021) Short-term wind power prediction based on EEMD-LASSO-QRNN model. *Appl Soft Comput* 105:107288
46. da Silva RG, Dal Molin Ribeiro MH et al (2021) A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting. *Energy* 216:119174
47. Jaseena KU, Binsu C, Kovoov (2021) Decomposition-based hybrid wind speed forecasting model using deep bidirectional LSTM networks. *Energy Convers Manag* 234:113944
48. Shao Z, Fu C, Yang SL et al (2017) A review of the decomposition methodology for extracting and identifying the fluctuation characteristics in electricity demand forecasting. *Renew Sustain Energy Rev* 75:123–136
49. Pincus SM (1911) Approximate entropy as a measure of system complexity. *Proc Natl Acad Sci USA* 88(6):2297–2301
50. Daubechies I, Heil C (1992) TTen lectures on wavelets. *Cbms-nsf Regional Conference Series in Applied Mathematics: Society for Industrial & Applied Mathematics*
51. Friedman M (1939) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Publ Am Stat Assoc* 32(200):675–701

52. Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. *Ann Math Stat* 11:86–92
53. Sheskin DJ (2000) *Handbook of parametric and nonparametric statistical procedures*. Chapman & Hall, London

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Zhiyuan Liao is currently pursuing the B.Sc. degree with the School of Information Engineering, Nanchang University, Nanchang, China. His current research interests include artificial intelligence and other computational intelligence techniques.



Jiehui Huang is currently pursuing the B.Sc. degree with the School of Information Engineering, Nanchang University, Nanchang, China. He is also a member of Chinese Society of Optimization, Overall Planning and Economic Mathematics. His current research interests include artificial intelligence and swarm intelligence algorithm.



Yuxin Cheng is currently pursuing the B.Sc. degree with the School of Information Engineering, Nanchang University, Nanchang, China. His current research interests include artificial intelligence and other computational intelligence techniques.



Chunquan Li received the B.Sc., M.Sc., and Ph.D. degrees from Nanchang University, Nanchang, China, in 2002, 2007, and 2015, respectively. He has been with the School of Information Engineering, Nanchang University, since 2002, where he is currently a Professor and a Ganjiang Distinguished Professor. He has published over 50 research articles. He was also a Visiting Professor with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. His current interests

include computing intelligence, haptics, virtual surgery simulation, robotics, and their applications to biomedical engineering.



Peter X. Liu received the B.Sc. and M.Sc. degrees from Northern Jiaotong University, China, in 1992 and 1995, respectively, and the Ph.D. degree from the University of Alberta, Canada, in 2002. He has been with the Department of Systems and Computer Engineering, Carleton University, Canada, since 2002, and he is currently a Professor and the Canada Research Chair. He is also with the School of Information Engineering, Nanchang University, as an Adjunct Professor. His interests include interactive networked systems and teleoperation, haptics, micro-manipulation,

robotics, intelligent systems, context-aware intelligent networks, and their applications to biomedical engineering. He has published more than 280 research articles. He was a recipient of the 2007 Carleton Research Achievement Award, the 2006 Province of Ontario Early Researcher Award, the 2006 Carty Research Fellowship, the Best Conference Paper Award from the 2006 IEEE International Conference on Mechatronics and Automation, and the 2003 Province of Ontario Distinguished Researcher Award. He is a Licensed Member of the Professional Engineers of Ontario (P. Eng), a Senior Member of IEEE, and a Fellow of the Engineering Institute of Canada. He serves as an Associate Editor for several journals, including the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE/ASME TRANSACTIONS ON MECHATRONICS, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and the IEEE ACCESS.