

# Highly accurate energy consumption forecasting model based on parallel LSTM neural networks

Ning Jin <sup>a</sup>, Fan Yang <sup>a</sup>, Yuchang Mo <sup>b</sup>, Yongkang Zeng <sup>a</sup>, Xiaokang Zhou <sup>c,d</sup>, Ke Yan <sup>e,\*</sup>, Xiang Ma <sup>a</sup>

<sup>a</sup> Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, College of Information Engineering, China Jiliang University, Hangzhou 310018, China

<sup>b</sup> Fujian Province University Key Laboratory of Computational Science, School of Mathematical Sciences, Huaqiao University, Quanzhou 362021, China

<sup>c</sup> Faculty of Data Science, Shiga University, Hikone 5228522, Japan

<sup>d</sup> RIKEN Center for Advanced Intelligence Project, RIKEN, Tokyo 1030027, Japan

<sup>e</sup> College of Design and Engineering, National University of Singapore, 117566, Singapore

## ARTICLE INFO

### Keywords:

Long short term memory  
Energy consumption  
Time series data analysis  
Forecasting  
Singular spectrum analysis

## ABSTRACT

The main challenges of the energy consumption forecasting problem are the concerns for reliability, stability, efficiency and accuracy of the forecasting methods. The existing forecasting models suffer from the volatility of the energy consumption data. It is desired for AI models that predict irregular sudden changes and capture long-term dependencies in the data. In this study, a novel hybrid AI empowered forecasting model that combines singular spectrum analysis (SSA) and parallel long short term memory (PLSTM) neural networks is proposed. The decomposition with the SSA enhanced the performance of the PLSTM network. According to the experimental results, the proposed model outperforms the state-of-the-art models at different time intervals in terms of both prediction accuracy and computational efficiency.

## 1. Introduction

Energy consumption forecasting (ECF) is an important application of artificial intelligence (AI) to support smart grid and smart city constructions [1]. Reliable, efficient and accurate ECF improves the transmission efficiency of the smart grid, secures energy market trading, and reduces energy wastage [2–4]. The Internet of things (IoT) technology introduces a pattern recognition process through sensor data collected from individual households for ECF [5,6]. The AI-enhanced data-driven ECF provides an important clue to the governments, power plants and residences towards the sustainability of energy usage [7].

As part of the advanced metering infrastructure (AMI) program for the smart grid construction, ECF for individual household's energy consumption data has increasingly attracted many attentions due to the high volatile time series data impacted by human behaviors [8,9]. It is usually difficult for traditional methods, such as physical model based methods, to perform predictions [10]. In contrast, along with the fast development of AI, deep learning technologies, such as the long short term memory neural networks, have been widely applied to the ECF problem for individual households.

The current main issue existed in the literature is that data-driven

ECF models do not provide sufficiently accurate and reliable forecasting results, while irregular fluctuations and sudden changes present in the sensor data [11]. In addition, errors in the raw sensor data also affect the forecasting performance [12]. Noise removal and volatility handling become the main tasks for ECF for individual households, which are also the two main issues to be tackled in this study.

In this study, a hybrid deep learning model are designed to perform ECF for individual households with noisy and volatile data samples. The singular spectrum analysis (SSA) is utilized to eliminate the signal noise. Moreover, SSA extracts the important features from the raw data by decomposing the original time series data (signal) into sub-signals. Multiple sub-signals were fed into a parallel long short term memory neural network (PLSTM) that consists of multiple LSTM neural networks. Each LSTM neural network handles one sub-signal. And all LSTM neural networks work parallelly, producing the forecasting results for each sub-signal. The final ECF result is obtained by combining all forecasted results from all LSTM neural networks.

In summary, a hybrid data-driven ECF model was designed and implemented combining the SSA with a PLSTM structure. The innovation of this work is that we design a novel parallel neural network architecture and verified the performance of the proposed method

\* Corresponding author.

E-mail address: [keddiyan@gmail.com](mailto:keddiyan@gmail.com) (K. Yan).

compared with existing methods with different ECF time interval and different households' data. Based on the experimental results collected, the accuracy of the proposed model is significantly higher than the existing models. The ECF error rates of the proposed model for five households located in UK with time interval of 5 min is reduced by 34.34%, 31.46%, and 57.48% in average under the MAE, MAPE, and RMSE metrics. In addition, the efficiency of the training process was improved by 23.36%, 45.96%, and 64.24% for the dataset of household 5 at different time intervals in average, compared with existing methods. The main contributions of the current study can be concluded as:

- (1) **A neural network structure integrating multiple parallel LSTM neural networks.** The proposed PLSTM structure is composed of a number of peer LSTM neural networks. All peer LSTM neural networks are structured to run in parallel for efficiency improvement. Each LSTM is employed to training the corresponding sub-signal generated by the SSA. The integrated approach enables a more precise prediction of using multiple sub-signals representing multiple single features, which potentially improves the final prediction accuracy.
- (2) **A novel AI-empowered forecasting framework combining SSA and PLSTM neural networks.** For better prediction performance, the singular spectrum analysis (SSA) is utilized eliminating the signal noise by signal decomposition. The outputs of SSA are fed into a series of long short term memory (LSTM) neural networks in a parallel manner. It is worth to mention that the SSA is used for data denoising and feature extraction.
- (3) **Outperformance over existing models for ECF of individual households.** The compared existing models include the Decision Tree (DT), Random Forest (RF), Multi-layer Perceptron (MLP), Support Vector Regression (SVR), LSTM, Nested LSTM (NLSTM), Empirical Wavelet Transform LSTM (EWT-LSTM), Empirical Mode Decomposition LSTM (EMD-LSTM), Variational Mode Decomposition LSTM (VMD-LSTM) and Stationary Wavelet Transform LSTM (SWT-LSTM) models. MAE, MAPE, RMSE, and R-square were used for performance evaluation. The experimental results showed that our proposed model outperformed all existing methods mentioned above.

## 2. Related works

Time series data prediction methods can be generally categorized into model-based or data-driven (AI-based) methods [13,14]. In [15], it is noted that the data-driven methods, such as the convolutional neural networks (CNNs) and long short term memory (LSTM) neural networks, are more suitable for energy consumption forecasting [16,17]. The data-driven methods are also more popular for modern forecasting problem, such as the forecasting problems related to the Covid-19 pandemic. Gitto et al. [18] utilized data-driven methods to forecast the hospital bed demand. Podder et al. [19] applied ARIMA to model and forecast to fighting against Covid-19. Arslan et al. [20] carried out their study in three dimensions forecasting total number of Covid-19 infected people.

Data-driven models are mainly divided into two types: singular models and hybrid models. Singular models include decision tree [21], random forest [22], support vector regression [23], multilayer perceptron [24], convolutional neural network [25], recurrent neural network [26], and long and short term memory neural network [27]. And hybrid models combine singular models for better prediction performances.

From a survey of the literature, the hybrid models generally produce higher prediction accuracy than the singular model. In [28], author utilizes coupling SSA and least square SVM for ECF. Yan et al. [29] claimed that the LSTM neural network has the better performance capturing the dependencies among data samples compared to SVM. Wei et al. [30] proposed a hybrid model combining ISSA and LSTM predicting daily natural gas consumption, but the author focuses on the superiority of the model in different climate zone instead of different

time span. Meanwhile, we are fully aware that data decomposition plays a significant role for forecasting result improvement. Sun et al. [31] decomposed the economic factors for energy consumption forecasting. Yuan et al. [32] proposed to predict building electricity load by SSA and WNN. However, this article does not explore the different forecasting performance over various time horizons. Yan et al. [29] utilized different time intervals for ECF and employed SWT to decompose the original data. However, data decomposition by SWT might not be broadly applicable, while the SWT prefers to separate the Gaussian noise, instead of filtering out the impulse noise. In contrast, the denoising by singular spectrum analysis (SSA) is capable of correctly separating noise from the high signal-to-noise ratio time series. In addition, SSA also eliminates the Gaussian noise. Neeraj et al. [33] proposed to use SSA-LSTM for forecasting on power load data set. However, in the step of recombination after grouping, the construction of non-singular matrix only utilizes to denoise, rather than further use the decomposition feature to training. The important function of SSA for feature extraction was ignored. Furthermore, more researchers convey their interests to visualization analysis of the ECF problem. Bharati et al. [34,35] demonstrated that visualization analysis is conducive to manage energy consumption. We attempt to apply relevant technology to realize the visualization analysis in the future. A summary of all surveyed works listed above is depicted in Table 1.

According to the literature survey, hybrid AI-based methods generally provide better forecasting performance for time series data compared to singular methods, which combine signal processing methods with neural networks. However, main obstacles exist for such approaches, including: 1) The original time series data has large volatility and sudden changes, resulting in low prediction accuracy for nonlinear data. 2) The data-driven models training processes require a high cost of time. 3) There are various signal processing algorithms and neural networks. The combinations of inappropriate decomposition methods and unmatched neural networks do not provide prominent prediction performance. 4) The traditional usage of SSA only includes denoising. The function of feature extraction in SSA is not fully utilized.

In this study, a hybrid model combining SSA and multiple LSTM neural networks is proposed. In the data preprocessing part, we selected SSA decomposition for both denoising and feature extraction. It is observed that the ECF data include a mass of impulse noise and the noise is distributed over the whole spectrum. Therefore, SSA is applied separating the Gaussian noise and impulse noise from ECF signal. After SSA, the data is divided into secular trend items, a slice of fluctuation items and noise items. In the model training part, we employed paralleled LSTM that matches the number of subsequences after decomposition. The LSTM neural network is proficient in processing non-linear and non-stationary time series. The original dataset is characterized by large fluctuations. The long-term dependence of the original ECF time series is able to be captured by LSTM. The final prediction results of the proposed method are more accurate than the state-of-the-art methods. The details of the proposed method are described in Section 3. The experimental details and results are shown in Section 4.

## 3. Methodology

This section describes the details of the hybrid forecasting model that we proposed. In the first place, it explains the indispensable process of using SSA algorithm for data processing. Secondly, we introduced the internal structure and workflow of the LSTM neural network, as well as the reasons why it is available and powerful. Finally, we introduced the interaction between SSA and PLSTM and the overall planning in this study. The overall flow chart of the proposed AI model is shown in Fig. 1.

### 3.1. Singular spectrum analysis

An adaptive signal decomposition algorithm named singular spectrum analysis (SSA) is adopted for data pre-processing. The ultimate

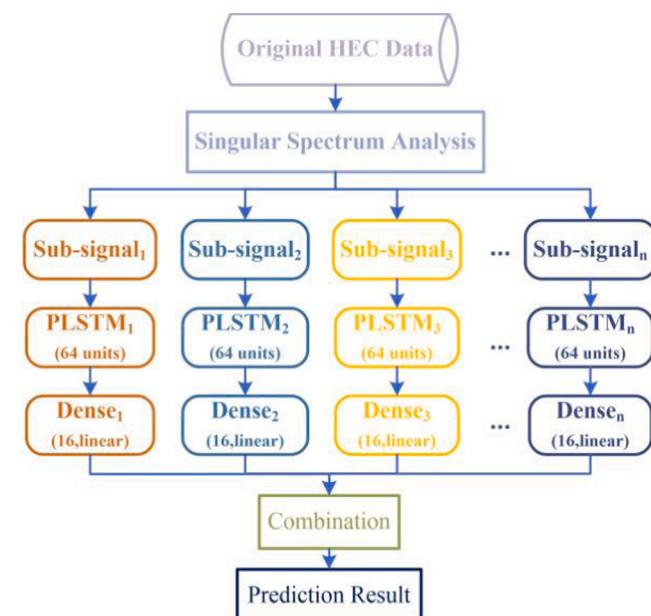
**Table 1**

Important forecasting methods surveys in this work: a method comparison.

Reference	Method	Contributions	Limitations
[15]	Conventional AI-based models	A large amount of ECF papers were reviewed and analyzed.	Only one evaluation metric is used.
[16]	Enhanced bagged echo state network	Achieved a better forecasting performance than the existing models.	The search algorithm and the forecasting result require further improvements.
[17]	ARIMA and seasonal ARIMA	Forecasted the streamflow with lack of information.	The generalization of the proposed model is not sufficient.
[18]	Growth models	Solved the problem of managing ICU capacity.	The size of the training dataset is small.
[19]	ARIMA and ExtraTrees classifier	Predicted whether a patient needs to be admitted to ICU or semi-ICU.	The results highly depend on the reliability of dataset.
[20]	SEIR-based simulator	It utilized three dimensions assumptions and forecasting algorithms.	It is hard to draw definite conclusions or make clear statements about the natural course of the disease.
[21]	Decision tree	The proposed method was compatible with large and complicated datasets.	Overfitting occurs with a smaller dataset.
[22]	Random forests	Improved the operational severe weather forecasting throughout the Day 1–3 period.	The superiority of the prediction results was hard to maintain under various scenarios.
[23]	Support vector regression	Modeling performance in terms of RMSE was outstanding without sacrificing too much generalization performance.	The model cannot be constructed until the full dataset is available.
[24]	Multiple layer perceptron, FIR and Elman network	The efficiency of the learning algorithm was largely improved.	The full structures of Elman and FIR were not implemented.
[25]	Convolutional neural network-based method	Outperformed other algorithms by an average accuracy improvement of 42.91%.	The running time was compromised.
[26]	Recurrent neural network	Yielded performance improvements without incurring longer training times.	The operations of back-propagations were required.
[27]	LSTM	It is capable of accurately predicting future longitudinal and lateral trajectories for vehicles on highway.	Delayed responses were observed.
[28]	SSA and single least square support vector machine	Reduced the computational burden.	The oceanic/atmospheric circulation factors were not considered as model inputs.
[29]	Stable wavelet transform and LSTM	Remarkable margins on performance comparison over the state-of-art methods.	The feature extraction step of the proposed method is not sufficiently discussed.
[30]	ISSA and LSTM	The denoising ability of SSA was improved.	The superiority of the model was not justified on different time intervals.
[31]	X12 and STL	Accurately forecasted the small-scale users who were easily	Unable to control the rate of change of seasonal components.

**Table 1 (continued)**

Reference	Method	Contributions	Limitations
[32]	Cuckoo search and WNN	disturbed by random factors.	The forecast precision was enhanced by the wavelet disintegration.
[33]	SSA and LSTM	Effectively improved the prediction accuracy.	The important function of SSA for feature extraction was ignored.
[35]	Visualization tools and regression analysis	Found that temperature and weather contribute significantly to ECF.	The analysis results might not be universally applicable to other applications.

**Fig. 1.** The flow chart of the proposed model.

purpose of SSA is to denoise the original raw ECF data and decompose the ECF into sub-signals that are more regular (Fig. 2).

As shown in Fig. 2, the SSA effectively reduces the difficulties of forecasting by the signal decomposition process [36]. The decomposed sub-signals are evidently more regular than the original time series data, which potentially raise the forecasting performance of the LSTM neural networks.

The four main procedures of SSA include: construction of trajectory matrix, singular value decomposition, sequence grouping and diagonal averaging, which are explained in detail in the following four subsections.

### 3.1.1. Construction of trajectory matrix

Given that the primitive continuity of time series, the trajectory matrix  $X$  is usually fixed. The core of  $X$  is the sliding window length  $\psi$  where  $2 \leq \psi \leq \frac{N}{2}$ . One-dimensional raw data  $X_{ori} = \{x_1, x_2, \dots, x_N\}$  is sliced according to  $\psi$ . Consequently, the original time series is divided into several time-lag vectors  $X_i = (x_i, \dots, x_{i+\psi-1})^T$ . The trajectory matrix  $X$ , also known as the Hankel matrix, is represented in formula (1). Parameters  $\psi$  and  $\beta$  are requested to satisfy the condition:  $\psi < \beta, \mathcal{N} = \beta + \psi - 1$ .

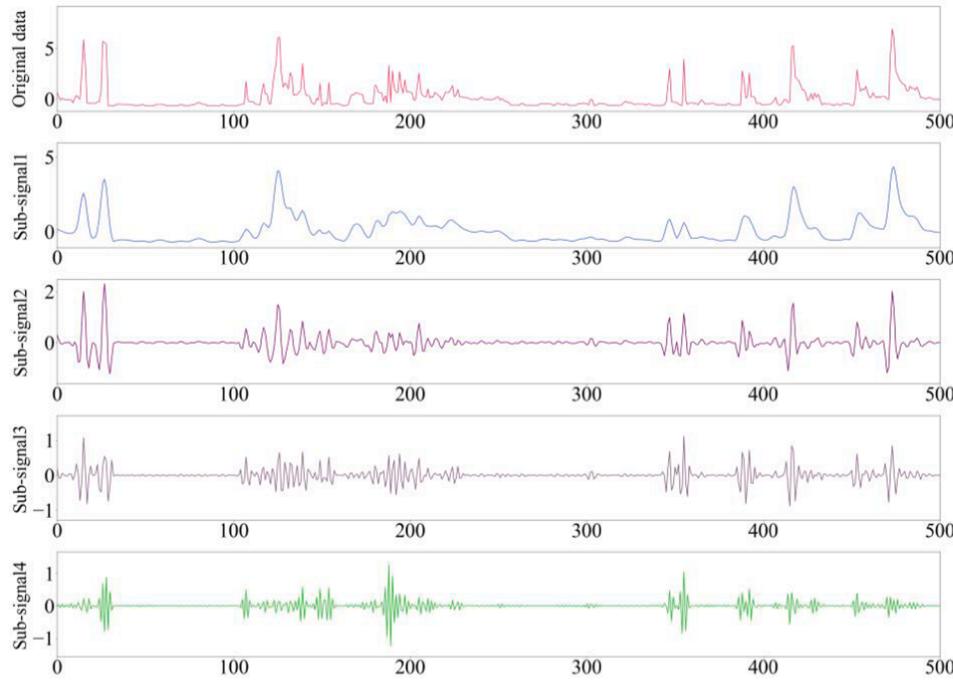


Fig. 2. Several subsequences of the original time series decomposed by SSA.

$$X = [X_1, \dots, X_\beta] = (X_{ij})_{i,j=1}^{\psi, \beta} \begin{bmatrix} x_1 & x_2 & \dots & x_\beta \\ x_2 & x_3 & \ddots & x_{\beta+1} \\ \vdots & \ddots & \ddots & \vdots \\ x_\psi & x_{\psi+1} & \dots & x_{\beta'} \end{bmatrix} \quad (1)$$

### 3.1.2. Singular value decomposition

**SVD** is the most crucial step of **SSA**. The vital purpose of this part is to denoise the original signal using the singular values  $\sqrt{\lambda_i}$ , where  $i \in \{1, 2, \dots, \psi\}$ . The principal component terms include the left eigenvector  $U_i$ , the right eigenvector  $V_i$ , the left singular matrix  $U$ , right singular matrix  $V$  (Eq. (2)) and diagonal matrix  $\Sigma$  (Eq. (3)). A matrix  $\mathcal{D}$  is obtained by diagonalizing the symmetric matrix  $X^T X$ . The singular values  $\sqrt{\lambda_i}$  is obtained by taking the square root of  $\mathcal{D}$ . The singular values  $\sqrt{\lambda_i}$  constructs the diagonal matrix  $\Sigma$ , where  $\Sigma$  conforms to  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_\psi$  and  $\lambda_\psi \rightarrow 0$ . It is worth noting that the left and right singularity matrices are eigenvectors of symmetric matrices  $XX^T$  and  $X^T X$ . The resultant matrix  $X$  (Eq. (5)) is consist of elementary matrix  $\mathcal{T}_i$  (Eq. (4)).

$$V_i = \frac{X^T U_i}{\sqrt{\lambda_i}} \quad (2)$$

$$\Sigma = \text{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_\psi}) \quad (3)$$

$$\mathcal{T}_i = \sqrt{\lambda_i} U_i V_i \quad (4)$$

$$X = \mathcal{T}_1 + \mathcal{T}_2 + \dots + \mathcal{T}_d, d = \max(i, \lambda_i > 0) \quad (5)$$

### 3.1.3. Sequence grouping for denoising

$p$  subsets (Eq. (6)) that are classified by the elementary matrix  $X_i$ . And the final matrix  $X$  is shown in Eq. (7). Singular values are employed as the basis for grouping. Smaller singular values are recognized as noise. Therefore, the first  $r$  larger singular values  $\{\lambda_1, \lambda_2, \dots, \lambda_r\}$  represent the main information and the rest of singular value  $\{\lambda_{r+1}, \dots, \lambda_i\}$  stand for the noise part. We discard the rest for better feature extraction and training. The value of  $r$  is determined by contribution rate  $\alpha$  (8).

$$X_I = X_{i_1} + \dots + X_{i_p} \quad (6)$$

$$X = X_{I_1} + \dots + X_{i_m} \quad (7)$$

$$\alpha = \frac{\lambda_i}{\sum_{i=1}^d \lambda_i} \quad (8)$$

### 3.1.4. Diagonal averaging

In this subsection, we convert each grouping matrix in Eq. (7) into a time series of length  $N$ .  $y_{ij}$  is an element in matrix  $Y$  with  $\psi$  rows and  $\beta$  columns, where  $1 \leq i \leq \psi, 1 \leq j \leq \beta$ . The matrix  $Y$  is transformed to  $Y = \{y_1, y_2, \dots, y_N\}$  in Eq. (9), where  $d^* = \min(\psi, \beta), f^* = \max(\psi, \beta)$ . The original time series  $X_{ori}$  is decomposed into  $m$  sub-series (Eq. (10)).

$$Y_k = \begin{cases} \frac{1}{k} \sum_{m=1}^k y_{m,k-m+1}, & 1 \leq k < d^* \\ \frac{1}{d^*} \sum_{m=1}^{d^*} y_{m,k-m+1}, & d^* \leq k < f^* \\ \frac{1}{N-k+1} \sum_{m=k-f^*+1}^{N-f^*+1} y_{m,k-m+1}, & f^* \leq k \leq N \end{cases} \quad (9)$$

$$y_t = \sum_{k=1}^m \hat{y}_t^k \quad (10)$$

### 3.2. Long short term memory

In RNN, the serious exploding gradient and vanishing gradient are occurred due to the long-distance among the nodes. The emergence of LSTM neural network aims to remedy the drawbacks of recurrent neural network (RNN). LSTM is published by Schmidhuber et al. [37]. Specifically, LSTM optimizes the repetitive modules of RNN, which converts from the original single neural network layer to four interactive modules. Therefore, LSTM is well-suited to process affairs with long-term dependencies. And LSTM is proficient to predict time series with large fluctuations and obvious mutations [38].

The detail of the inner workings of LSTM is plotted in Fig. 3. The input and output of the current network are  $x_t$  and  $h_t$ , respectively.  $C_t$  is the state that learned from the current moment. It is noting that there are

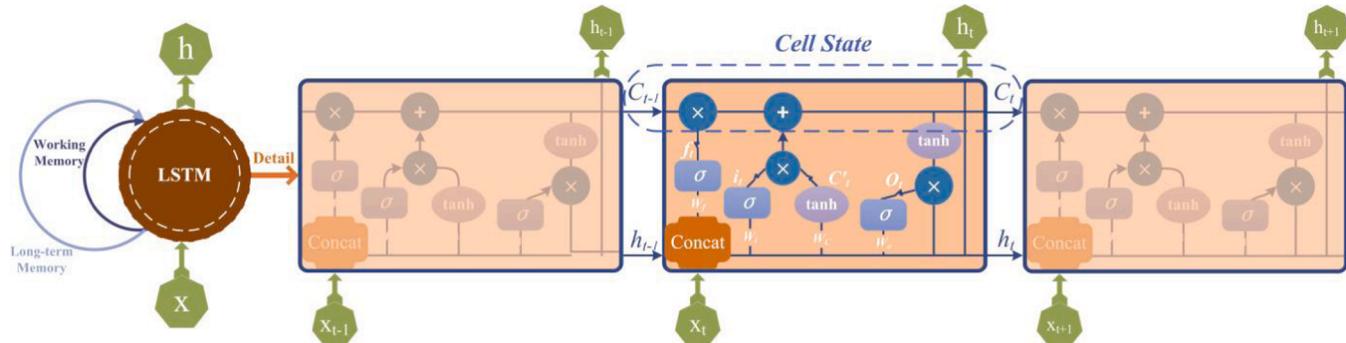


Fig. 3. The detailed structure of LSTM neural network.

three gates in the LSTM structure: input gate, output gate and forget gate.

Input gate, denoted as  $i_t$ , is responsible for how much current input data transfers to the cell state. With the control of input gate, a multitude of current irrelevant content avoid entering memory. The forget gate decided how much cell state at previous time saves to the current state, namely  $f_t$ , which reserves information from a long time ago. Stay or discard the current state is controlled by the output gate  $O_t$ . These elements determine the output cell state  $C_t$ .

At each processing, LSTM repeats the operation that the current input and the previous cell state will feed in the current state and output to the next cell state. The formulas for cell states and gating units are shown as follows, where  $W$  means the corresponding weight matrix, and  $b$  implies the related bias items. It is worth noting that  $\sigma$  and  $\tanh$  are activation functions that play an on-off role. The formula for the above elements is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (11)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (12)$$

$$C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (13)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot C'_t \quad (14)$$

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (15)$$

$$h_t = O_t \odot \tanh(C_t) \quad (16)$$

### 3.3. Hybrid forecasting model

Moving on to the AI model that we proposed. The detail of the AI model algorithm is presented in Algorithm 1.

#### Algorithm 1: Proposed AI model

**Input:** 5 raw ECF time series  $Y_{ori,1 \dots 5}$ , where  $ori \in \{1, 2, \dots, N\}$ . Training set size  $S_{train}$ . Testing set size  $S_{test}$ .

**Standardization:** Standardized ECF dataset conforms to mean value 0 and standard deviation 1 by utilizing Z-Score:  $F_{std} = \frac{y - \bar{y}}{std}$ ,  $Y_{std} = F_{std}(Y_{ori})$

Employ SSA to decomposed the standardization time series into multiple sub-signals.

Construct trajectory matrix:  $Y_{ik} = F_{embed}(Y_{std})$ , where  $i \in \{0, 1, 2, 3\}$ ,

$k \in \{0, 1, \dots, n-3\}$

Singular value composition:  $Y_{tra} = U \sum V^T = \sum_{i=0}^r \sqrt{\lambda_i} u_i (\bar{v}_i)^T$

Group:  $Y_{tra} = Y_{i_1} + \dots + Y_{i_p}$

Reconstruct:  $Y_{ik} = F_{rec}(Y_{tra})$

Get training set, validation set and testing set:

$Y_{ik} = \{TrainX_{ik}, TrainY_{ik}, VaX_{ik}, VaY_{ik}, TestX_{ik}, TestY_{ik}\}$

**Training:**  $i$  different parallel LSTM neural networks are prepared for  $TrainX_{ik}$  for  $i = 0$  to  $n$  do

Set Loss function:  $MSE = \frac{1}{t} \sum_{k=1}^t (y_k - \bar{y})^2$

Training data:  $TrainX_k$ , where  $k \in \{0, 1, \dots, t-7\}$

(continued on next column)

(continued)

#### Algorithm 1: Proposed AI model

Inverse standardization:  $Y_{ik} = F_{istd}(Y_{ik})$

Output: The prediction of  $Y_{ori,1 \dots 5}$  and evaluation result.

The experiments demonstrated that the hybrid model possesses higher accuracy and stronger generalization ability. And the whole process mainly includes three steps:

Step 1: Utilize Z-Score to standardize the five original ECF time series into five series with an average value of 0 and a standard deviation of 1. Each series is decomposed into several sub-signals by implementing SSA process. The excess decomposition layers in the subsequence are not available for the feature extraction.

Step 2: The sub-signals obtained from step 1 are divided into multiple training sets, validation sets and testing sets. These 20 sub-signals are the input of the PLSTM neural network. The network is optimized by learning the fitting process between 20 training sub-signals and the raw data of five ECF. After combining the outcomes of each four sub-signals we are able to acquire five final outcomes. After training, utilize the ECF testing data set to obtain the final result.

Step 3: Inverse standardized the final result we gained from step 2. It is purposed to return the data to its original size, and then perform the evaluation functions to assess the five households' prediction performance in terms of goodness of fit and error severity.

The schematic diagram of the proposed AI framework is shown in Fig. 4.

The potential improvements of the proposed method over the existing approaches lie on two aspects. First, the SSA is used for both denoising and feature extraction, while the conventional methods usually only use SSA for denoising for the training dataset. Second, the parallel LSTM neural network structure potentially improves the forecasting results over centralized singular LSTM neural network due to the parallel trainings on extracted features (decomposed sub-signals). In addition, the parallelized structure will also greatly shorten the training time, providing a more efficient and robust learning structure for the ECF problem.

## 4. Experimental process and results

### 4.1. Data description

UK-Dale which released in April 2017 is adopted as the statistical basis of the experiments [39]. It was collected by domestic appliance-level and the individual appliance-level in five households from 22:28:15 on November 9, 2012 to 18:35:53 on April 26, 2017. Each of the five household has a different number of occupants. Each occupant has a significant difference in lifestyle. The total number of meters is 101. The maximum time was recorded in the first households, totaling 786 days.

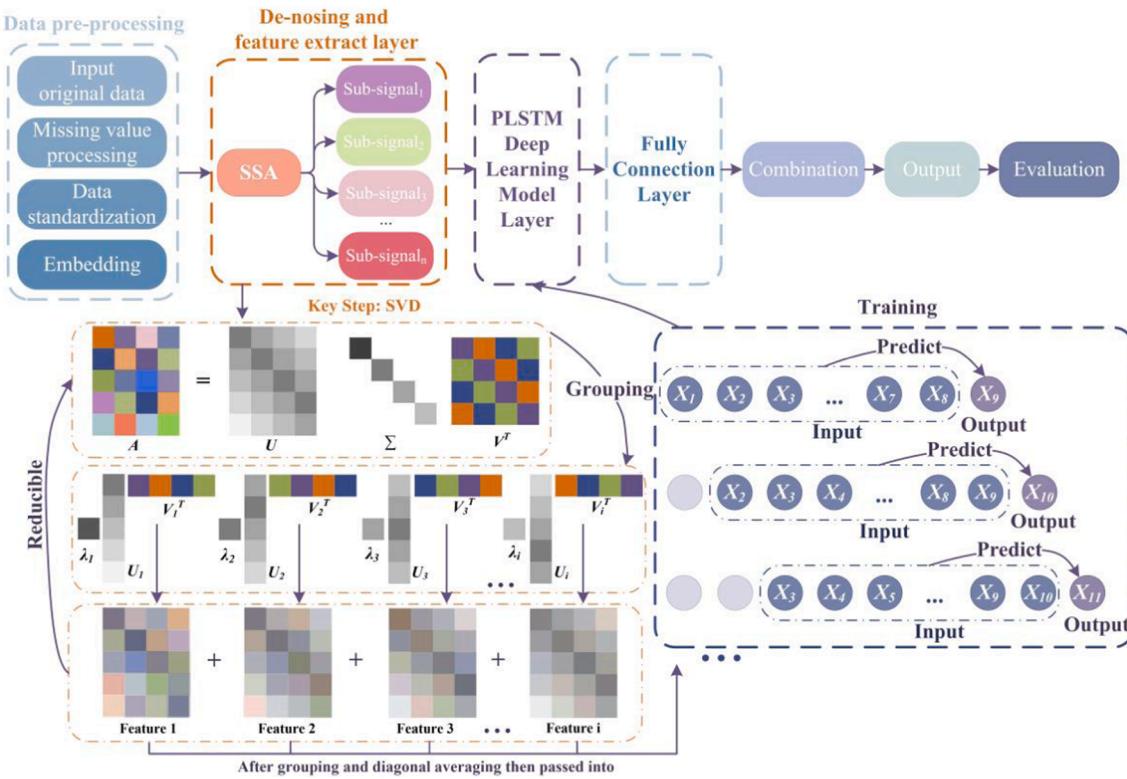


Fig. 4. The schematic diagram of the proposed method.

In order to energy saving and fire prevention, the appliance was not powered once in a while, so any gap of more than two minutes is considered as zero. In our study, equivalent time sampling was taken every 6 s. Each time step is 5 min long. So as to reflect both the short-term and the long-term trend of the dataset, a total of 1, 2, 4, 6 and 12 step sizes were utilized in the training.

On this basis, the data is divided into 87.5% training set and 12.5% testing set. In addition, 5% of the training set is the validation set. To be specific, taking 5 min as a time step, the range of the training set is the first 8400 units, of which 420 data are the validation set. The last 1200 units are the testing set.

#### 4.2. Experimental setup

##### 4.2.1. Software and hardware configuration

The experiments are completed on the Tensorflow1.15.0, Keras 2.3.1 and Python 3.6.2 in Win10 system. The hardware configurations are Intel i7-9700 K CPU and NVIDIA GeForce GTX 2060 GPU. All the source code and required data sets of the experiment can be obtained for free from <https://github.com/Fairy-09/HECO>. In addition, we have shown the environment configuration required for the experiment in the “README.md” file. All the results shown in this article (Fig. 2, Fig. 7, Fig. 8, and Fig. 8) are drawn by matplotlib 3.3.1 in python according to the experimental results. The remaining images are drawn using Microsoft Visio 2010.

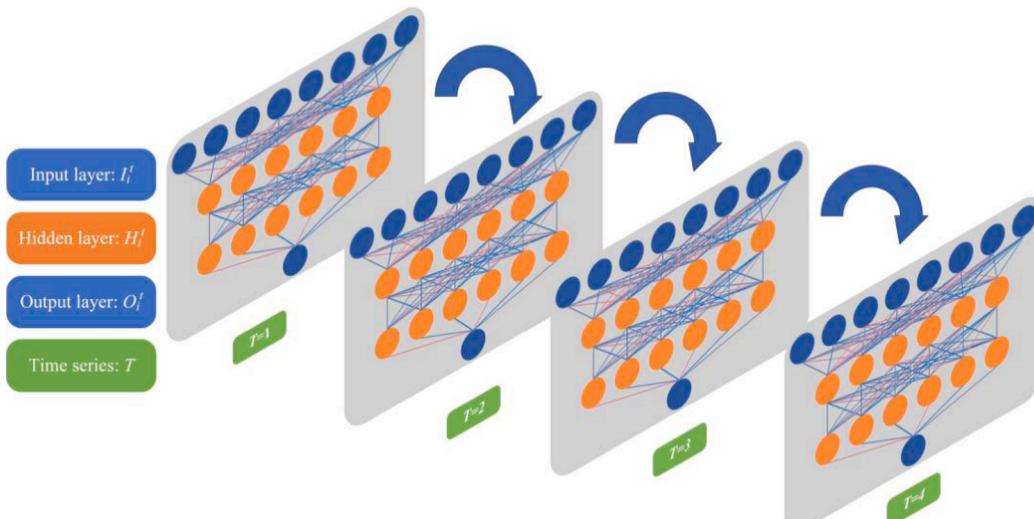


Fig. 5. The details of the proposed network.

#### 4.2.2. Detailed information about all compared models

In the experiment, we selected several classic singular models and latest hybrid models for comparisons, including Decision Tree (DT), Random Forest (RF), Multi-layer Perceptron (MLP), Support Vector Regression (SVR), LSTM, Nested LSTM (NLSTM), Empirical Wavelet Transform LSTM (EWT-LSTM), Empirical Mode Decomposition LSTM (EMD-LSTM), Variational Mode Decomposition LSTM (VMD-LSTM) and Stationary Wavelet Transform LSTM (SWT-LSTM). The detailed structure of the proposed model is shown in Fig. 5. The significant hyperparameter settings of all the models mentioned above are shown in Fig. 6.

DT and RF are traditional supervised learning models and have a long history. DT is constructed according to the principle of minimizing loss functions. RF is composed of multiple DTs. Various DTs are trained on different training sets separately. Finally, the results of each learner are combined as for the final result of RF. SVR is an application of SVM to regression problems, which greatly depends on the selection of kernel functions. According to the experimental results, SVR with the radial basis function (RBF) kernel is effective in ECF among these three machine learning models (DT, RF and SVR).

In comparison to traditional machine learning models, deep learning models have strong learning and generalization capability. The structure of full connection with activation function ReLU in MLP alleviates gradient disappearance. It is appropriate to dispose the nonlinear dataset [40]. Taking into accounting the characteristics of the total data, 64 hidden neurons were utilized in each four feedforward network layers, and 16 units were employed in Dense layers for LSTM and NLSTM. An adaptive learning rate method called root mean square prop (RMSprop) was utilized as the optimizer in LSTM [41]. The formula of the selected optimizer is shown in Eqs. (17) and (18).  $r$  is the accumulation of squared gradient  $g$ . The attenuation coefficient  $\rho$  is adopted to control the number of acquired historical information.  $\delta$  is a constant which maintained the stability. Adjust the learning rate  $\epsilon$  according to the element operation and then obtain the updated independent variable

θ. RMSprop is suitable for the recurrent network architecture and accelerates the optimization speed [42]. The above hyperparameter selections make LSTM and NLSTM neural network not prone to overfitting and have long-term dependence.

$$r_t = \rho r_{t-1} + (1 - \rho)g \odot g \quad (17)$$

$$\theta = \theta - \frac{\epsilon}{\sqrt{\delta + r}} \odot g \quad (18)$$

Models that combine LSTM with signal processing are also adopted in the comparative study. Four signal processing methods were presented in this experiment, including the empirical wavelet transform (EWT) [43], the intrinsic mode functions (IMF) [44], the variational mode decomposition (VMD) [45] and stationary wavelet transform (SWT) [46]. All above-mentioned signal processing methods decomposed the original time series data as an overall signal. The decomposed sub-signals were processed by the same PLSTM structure used in the current study. The MAE, MAPE and RMSE results of EWT-LSTM, IMF-LSTM, VMD-LSTM and SWT-LSTM are shown in Tables 2–5. It is evident that the proposed method outperforms all compared methods mentioned above.

All of the above state-of-the-art models are widely utilized and represent the landmark contributions in the subject research of time series prediction. Comparing these excellent models indicates the robustness of the proposed model under different conditions. The main work of the proposed signal processing algorithm in our study is to decompose the ECF data into four signal components, which contain one major information sub-signal and three minor information sub-signals. In this way, data feature extraction is realized, laying a good foundation for neural network learning. Then the processed signal components are put into four paralleled LSTM for training. The first 8 values are implemented to predict the value of the ninth number. The ratio of dropout is set to avert overfitting. Taking one training session as an example, the original dataset was split into 8400 training data and 1200

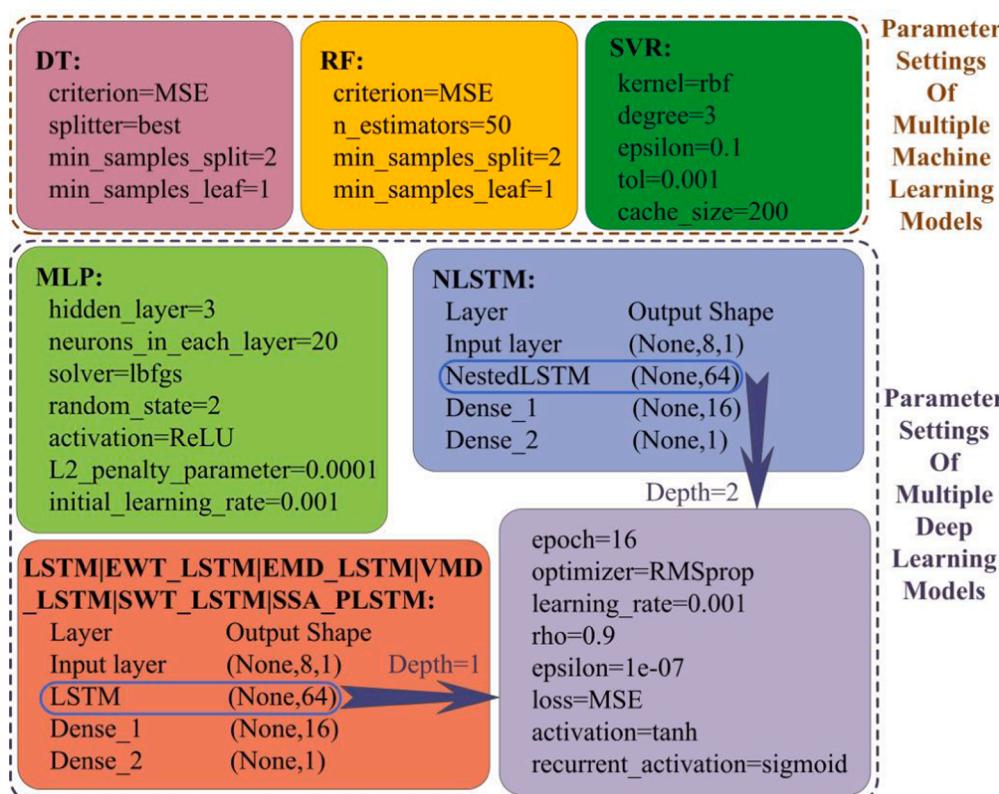


Fig. 6. The hyperparameter settings of all models.

**Table 2**

MAE, MAPE, RMSE of five households when sampling frequency was 5 min.

Data	MAE					MAPE (%)					RMSE				
	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5
<i>Decision-Tree</i>	0.0147	0.0126	0.0172	0.0141	0.0118	30.88	29.76	37.55	34.13	14.50	0.0320	0.0329	0.0447	0.0321	0.0342
<i>Random-Forest</i>	0.0104	0.0102	0.0128	0.0107	0.0093	20.96	24.10	28.25	28.09	11.69	0.0226	0.0242	0.0317	0.0230	0.0256
<i>MLP</i>	0.0106	0.0104	0.0129	0.0102	0.0087	21.31	22.67	28.20	25.57	9.26	0.0227	0.0257	0.0313	0.0228	0.0281
<i>SVR</i>	0.0093	0.0088	0.0123	0.0094	0.0097	16.04	16.23	28.05	23.06	10.92	0.0230	0.0249	0.0323	0.0218	0.0294
<i>LSTM</i>	0.0108	0.0090	0.0124	0.0093	0.0079	22.79	18.57	26.36	22.86	8.72	0.0226	0.0232	0.0323	0.0214	0.0246
<i>NLSTM</i>	0.0098	0.0084	0.0121	0.0090	0.0085	19.23	16.34	27.01	20.49	9.16	0.0219	0.0228	0.0317	0.0213	0.0275
<i>EWT-LSTM</i>	0.0114	0.0101	0.0157	0.0112	0.0110	26.86	23.71	51.02	33.43	16.35	0.0217	0.0235	0.0315	0.0222	0.0244
<i>EMD-LSTM</i>	0.0109	0.0096	0.0140	0.0082	0.0096	28.78	28.72	50.45	21.47	17.34	0.0176	0.0173	0.0240	0.0162	0.0193
<i>VMD-LSTM</i>	0.0074	0.0069	0.0093	0.0064	0.0063	16.85	17.96	31.65	19.48	9.90	0.0130	0.0126	0.0170	0.0117	0.0129
<i>SWT-LSTM</i>	0.0043	0.0039	0.0052	0.0048	0.0038	9.03	8.55	12.78	16.19	5.02	0.0090	0.0094	0.0128	0.0094	0.0106
<i>Proposed</i>	<b>0.0028</b>	<b>0.0020</b>	<b>0.0025</b>	<b>0.0028</b>	<b>0.0023</b>	<b>6.67</b>	<b>5.56</b>	<b>7.37</b>	<b>10.55</b>	<b>4.06</b>	<b>0.0043</b>	<b>0.0037</b>	<b>0.0051</b>	<b>0.0045</b>	<b>0.0040</b>

**Table 3**

MAE, MAPE, RMSE of the five households when sampling frequency was 10 min.

Data	MAE					MAPE (%)					RMSE				
	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5
<i>Decision-Tree</i>	0.0327	0.0302	0.0453	0.0360	0.0275	36.30	34.14	49.62	51.98	18.23	0.0605	0.0698	0.1040	0.0741	0.0710
<i>Random-Forest</i>	0.0243	0.0222	0.0366	0.0240	0.0237	26.97	25.76	41.30	36.98	16.70	0.0438	0.0468	0.0759	0.0466	0.0559
<i>MLP</i>	0.0266	0.0232	0.0387	0.0245	0.0253	29.21	26.00	46.06	34.39	14.10	0.0503	0.0504	0.0819	0.0483	0.0760
<i>SVR</i>	0.0216	0.0196	0.0350	0.0205	0.0239	20.39	17.71	38.62	26.88	13.77	0.0459	0.0482	0.0792	0.0436	0.0651
<i>LSTM</i>	0.0245	0.0206	0.0368	0.0207	0.0217	28.24	21.77	43.65	28.53	13.06	0.0427	0.0463	0.0806	0.0431	0.0566
<i>NLSTM</i>	0.0241	0.0188	0.0399	0.0202	0.0209	26.46	18.67	56.72	27.22	12.20	0.0432	0.0459	0.0784	0.0422	0.0561
<i>EWT-LSTM</i>	0.0226	0.0265	0.0422	0.0218	0.0266	24.38	34.76	67.73	33.19	19.65	0.0421	0.0482	0.0761	0.0413	0.0570
<i>EMD-LSTM</i>	0.0222	0.0233	0.0466	0.0157	0.0236	29.83	31.20	97.96	22.35	19.81	0.0341	0.0400	0.0719	0.0284	0.0411
<i>VMD-LSTM</i>	0.0151	0.0163	0.0271	0.0139	0.0182	19.66	21.35	50.13	22.01	14.83	0.0232	0.0275	0.0440	0.0228	0.0320
<i>SWT-LSTM</i>	0.0098	0.0089	0.0159	0.0097	0.0093	11.71	9.68	20.16	13.92	6.76	0.0172	0.0198	0.0329	0.0199	0.0209
<i>Proposed</i>	<b>0.0042</b>	<b>0.0051</b>	<b>0.0080</b>	<b>0.0058</b>	<b>0.0036</b>	<b>5.83</b>	<b>7.47</b>	<b>13.40</b>	<b>9.21</b>	<b>2.79</b>	<b>0.0064</b>	<b>0.0084</b>	<b>0.0133</b>	<b>0.0092</b>	<b>0.0082</b>

**Table 4**

MAE, MAPE, RMSE of the five households when sampling frequency was 20 min.

Data	MAE					MAPE (%)					RMSE				
	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5
<i>Decision Tree</i>	0.0762	0.0609	0.1235	0.0718	0.0827	46.04	32.50	71.88	49.44	29.25	0.1335	0.1336	0.2380	0.1433	0.1779
<i>Random Forest</i>	0.0529	0.0486	0.1057	0.0507	0.0572	30.47	28.98	64.12	36.34	19.16	0.0902	0.1012	0.1902	0.1032	0.1295
<i>MLP</i>	0.0668	0.0501	0.1224	0.0585	0.0602	38.20	28.29	72.60	36.79	17.12	0.1098	0.0985	0.2242	0.1290	0.1611
<i>SVR</i>	0.0481	0.0425	0.0962	0.0447	0.0528	23.30	19.21	52.71	27.15	15.47	0.0901	0.1029	0.1796	0.1010	0.1345
<i>LSTM</i>	0.0539	0.0469	0.1022	0.0445	0.0532	32.77	24.70	73.84	28.50	16.31	0.0876	0.0998	0.1737	0.0995	0.1331
<i>NLSTM</i>	0.0543	0.0415	0.1066	0.0421	0.0504	32.21	20.96	81.42	25.72	14.73	0.0878	0.0954	0.1741	0.0986	0.1327
<i>EWT-LSTM</i>	0.0483	0.0540	0.1055	0.0464	0.0660	26.50	35.92	84.31	33.71	24.92	0.0842	0.0953	0.1712	0.0950	0.1317
<i>EMD-LSTM</i>	0.0388	0.0597	0.1014	0.0370	0.0599	24.53	43.55	99.04	32.75	23.87	0.0585	0.0881	0.1405	0.0689	0.1032
<i>VMD-LSTM</i>	0.0364	0.0385	0.0655	0.0380	0.0485	23.10	27.93	57.28	26.04	22.13	0.0517	0.0602	0.0988	0.0676	0.0721
<i>SWT-LSTM</i>	0.0234	0.0224	0.0431	0.0204	0.0258	13.41	13.52	29.98	14.84	9.27	0.0382	0.0447	0.0766	0.0446	0.0549
<i>Proposed</i>	<b>0.0088</b>	<b>0.0105</b>	<b>0.0241</b>	<b>0.0114</b>	<b>0.0134</b>	<b>5.81</b>	<b>7.48</b>	<b>18.59</b>	<b>7.973</b>	<b>5.57</b>	<b>0.0145</b>	<b>0.0165</b>	<b>0.0405</b>	<b>0.0225</b>	<b>0.0232</b>

**Table 5**

MAE, MAPE, RMSE of the five households when sampling frequency was 30 min.

Data	MAE					MAPE (%)					RMSE				
	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5	Hse_1	Hse_2	Hse_3	Hse_4	Hse_5
<i>Decision Tree</i>	0.1099	0.0916	0.1423	0.1196	0.1923	42.93	40.92	55.65	57.07	62.42	0.1751	0.1717	0.2577	0.2081	0.2952
<i>Random Forest</i>	0.0852	0.0684	0.1583	0.0802	0.1364	32.99	29.05	64.22	41.03	40.87	0.1323	0.1175	0.2564	0.1324	0.2216
<i>MLP</i>	0.1334	0.0746	0.2061	0.0929	0.1203	47.14	30.74	79.70	44.68	28.69	0.3162	0.1462	0.3931	0.1572	0.2427
<i>SVR</i>	0.0776	0.0659	0.1489	0.0666	0.0834	25.58	22.84	51.39	28.68	16.40	0.1319	0.1274	0.2608	0.1221	0.2062
<i>LSTM</i>	0.0835	0.0705	0.1601	0.0668	0.0842	34.62	28.69	74.24	28.92	18.49	0.1237	0.1228	0.2465	0.1224	0.1926
<i>NLSTM</i>	0.0829	0.0673	0.1544	0.0733	0.0754	32.79	26.11	68.91	31.17	14.79	0.1263	0.1224	0.2411	0.1289	0.1950
<i>EWT-LSTM</i>	0.0767	0.0745	0.1422	0.0628	0.0911	29.34	32.63	68.66	30.08	22.76	0.1196	0.1156	0.2283	0.1089	0.1772
<i>EMD-LSTM</i>	0.0563	0.0787	0.1163	0.0535	0.0724	24.28	40.81	63.03	33.10	20.25	0.0822	0.1144	0.1608	0.0738	0.1292
<i>VMD-LSTM</i>	0.0705	0.0556	0.0922	0.0461	0.0717	27.85	26.96	48.73	26.57	20.57	0.0963	0.0821	0.1325	0.0693	0.1141
<i>SWT-LSTM</i>	0.0332	0.0338	0.0696	0.0291	0.0382	13.64	15.10	30.45	13.43	9.172	0.0515	0.0572	0.1137	0.0504	0.0788
<i>Proposed</i>	<b>0.0177</b>	<b>0.0202</b>	<b>0.0332</b>	<b>0.0212</b>	<b>0.0167</b>	<b>7.898</b>	<b>10.79</b>	<b>14.74</b>	<b>11.30</b>	<b>4.82</b>	<b>0.0261</b>	<b>0.0271</b>	<b>0.0520</b>	<b>0.0298</b>	<b>0.0317</b>

testing data. Then, split 420 validation data from the training data. It is noted that the validation data is utilized to detect the overfitting in training results, rather than update the weight. Ultimately, the different batches of training results are combined to establish the final prediction results.

#### 4.2.3. Evaluation metric

By convention, we believe that a preeminent model has stability and accuracy. The Mean Square Error (MSE) is continuous and derivable, so it facilitates the convergence of the gradient descent algorithm. So MSE is chosen as the loss function and model evaluation parameter.

$$MSE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (18)$$

In order to examine the superiority of this method from multiple perspectives, it is necessary to comprehensively evaluate various superiority models. Therefore, we have set up three evaluation schemes for the error level, the order of magnitude, and the overall fitting degree. Three evaluation metrics are: Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). The closer the values of these three evaluation indexes are approach to nil, the more perfect the algorithm will be.

$$MAE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (19)$$

$$RMSE(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (20)$$

$$MAPE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (21)$$

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = \frac{SSR}{TSS} = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (22)$$

where  $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$  refers to the real value;  $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$  is the predicted value;  $\bar{y}$  is the mean value.

Additionally, a criterion of conformity degree between the actual value and predicted value is also set as the indicator to evaluate the prediction performance, which is called R Squared ( $R^2$ ).  $R^2$  is a coefficient of determination in a range from 0 to 1. The determination coefficient is defined as the ratio of Sum of Squares of the Regression (SSR) to Total Sum of Squares (TSS). When  $R^2$  is close to 1, it verifies the formidable fitting competence of the model. Simultaneously, it is vital to note that when the  $R^2$  value of the model is less than the mean, we treated it as 0.

## 5. Experimental results

In this section, we expose the experimental results under multiple evaluation indicators. According to the experimental results, the proposed algorithm effectively screens out the diverse features in the

original ECF data and alleviates the impact of abnormal and invalid data on the prediction results. Judging from the values of three error evaluation indexes and one fitting evaluation index, the prediction performance of the comparison model is slightly different in multiple households. The results are all placed in Tables 2–5, where exhibit the quantized value of multiple algorithms in five households. Each table takes a different time step. Fig. 7. exploits the polar axis roses chart to display the  $R^2$  values of the proposed model and comparison models. It shows that the algorithm proposed by us outperforms all the comparison models. Compared with the comparison model, the MAE, MAPE and RMSE of the proposed algorithm drop sharply. The average  $R^2$  value of five households is 0.98, 0.98, 0.97, 0.98, 0.98. Five outcomes all close to 1 demonstrates that the proposed algorithm has a high goodness-of-fit.

According to the principia and experiment results, it can be seen that compared with other preprocessing methods, the meticulous mathematical theory of SSA is the basis of superior performance. SSA decomposed the raw data into subsequences that are easy to learn for deep learning algorithms. The results justified that SSA is capable of processing the data which is nonlinear and lacks regularity for a short period of time. Parallel training of LSTM for different subsequences is able to maximize the advantages of the SSA decomposition. In order to verify the advantages of paralleled LSTM in prediction, we further conducted the second experiment. In Fig. 8, we show the training time of different LSTM extensions combining with SSA, which include the bidirectional LSTM (BiLSTM) [49], nested LSTM [50] and stacked LSTM [51]. According to Fig. 8, the proposed method presents the shortest running time. SSA\_BiLSTM and SSA\_NLSTM run slightly slower. And SSA\_SLSTM took nearly three times as long to train. The different structures of these four models are described in Figs. 9–12, where  $X$  represents the original input,  $H$  represents the current output, and  $Y$  represents the final prediction result.

Fig. 9 shows the flowchart of SSA-PLSTM. SSA-PLSTM is able to remember useful information and forget unimportant information. The number of parallel LSTMs depends on the number of corresponding signal decomposition layers. Each LSTM maintains the same structure and hyperparameters. The proposed model can effectively and accurately predict sub-sequences with different features.

Fig. 10 shows the flowchart of SSA-BiLSTM. Conventional LSTM is unidirectional, while BiLSTM combines forward LSTM and backward LSTM. In the mid-point of the structure, we concatenate the forward and backward hidden vectors. The final prediction result is determined by the combination of the output results of the forward layer and the backward layer at the corresponding time.

Fig. 11 shows the structure of SSA-NLSTM. In NLSTM, the value of one memory unit is calculated by an LSTM unit. The external memory unit can freely choose to read and write related long-term information to the internal unit. NLSTM increases the depth of the neural network, so that the structure can remember the content of a longer time range.

Fig. 12 shows the structure of SSA-LSTM. SLSTM stacks LSTM to increase the depth of the neural network. The output sequence of the previous LSTM is utilized as the input of the next LSTM layer. The number of network layers of SLSTM can be selected manually, and the multiple network layers allows the model to capture more dependencies.

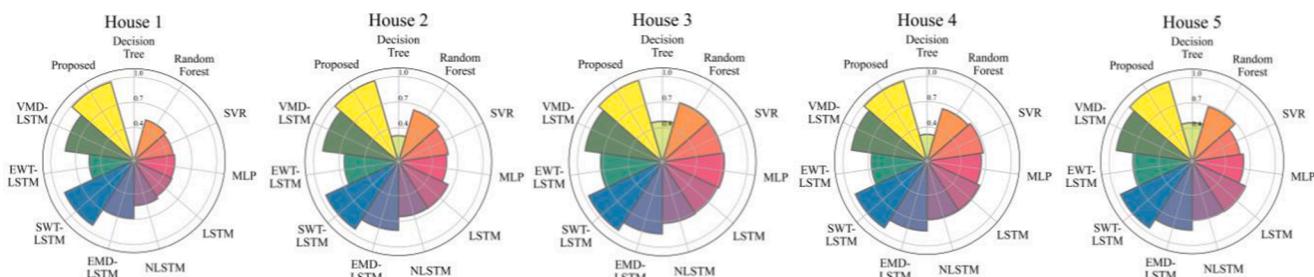
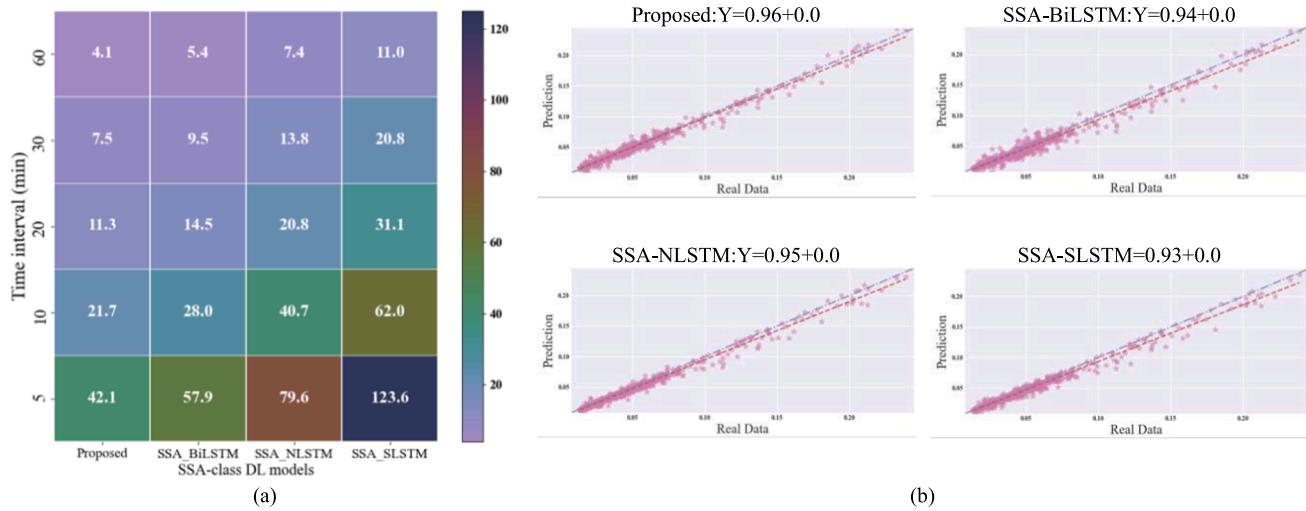
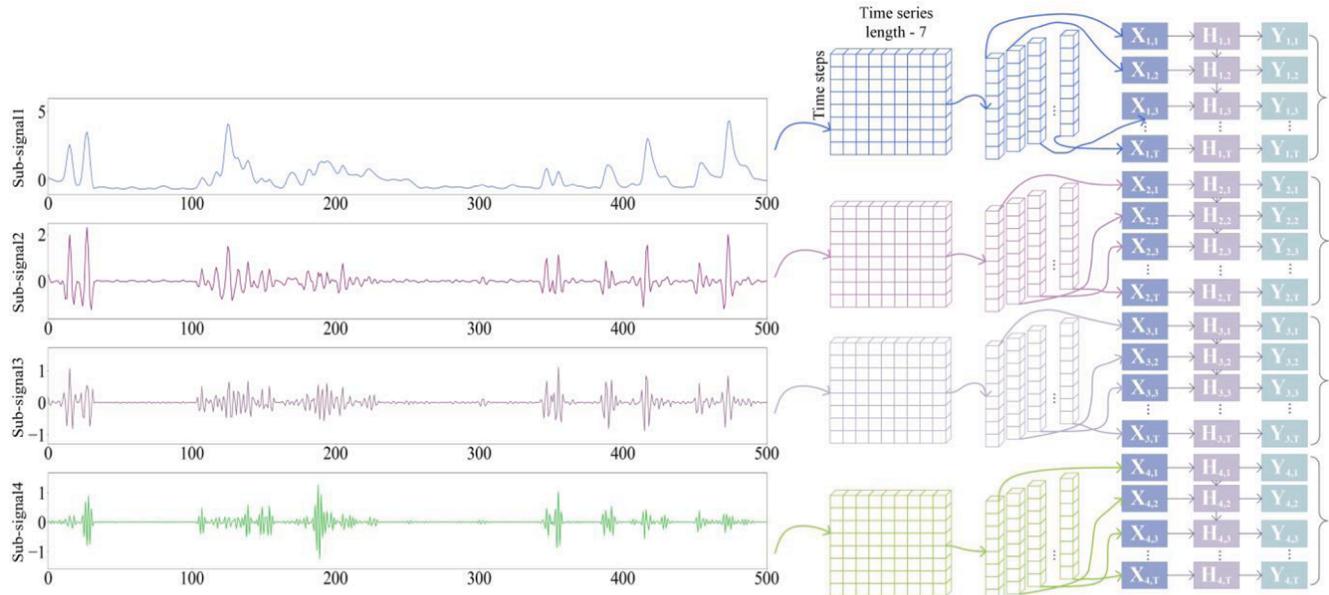


Fig. 7. The  $R^2$  value of the SSA-PLSTM and other comparison models.



**Fig. 8.** (a) plots the training time of four SSA-class models in different time intervals. (b) plots the linear regression of the comparison between different SSA-class ensemble models.



**Fig. 9.** The approach of SSA-PLSTM.

The four models introduced above are all cutting-edge deep learning models, which have obvious advantages compared to traditional models. According to the experimental results, the SSA-PLSTM outperforms the existing methods in terms of both accuracy and efficiency, which is evidently to be the most suitable method for ECF of individual households.

## 6. Aggregate results

In this section, we compare and analyze the performance of the state-of-art models with the proposed method using multiple real-world household datasets. The state-of-art models include DT, RF, MLP, SVR, NLSTM and various extensions of LSTM. It shows an obvious lagging phenomenon in the fitting diagram Fig. 13. However, the lagging phenomenon is much relieved with our method (Fig. 13).

In Fig. 13, the various LSTM extensions including EWT-LSTM, EMD-LSTM, VMD-LSTM, and SWT-LSTM also utilize signal processing algorithms to preprocess data. However, the performances of the above-

mentioned signal processing algorithms are worse than that of our method. In addition, the method of EMD empirically decomposes the high-frequency and low-frequency parts incompletely when decomposing data with large fluctuations. Too many rounds of decompositions of EMD do not help in the final prediction performance.

The prediction results of EWT-LSTM and EMD-LSTM are volatile in the place which is expected stable and smooth. Furthermore, a noticeable lag occurs. Lag state in the electricity consumption prediction is lacks of practical meaning. VMD-LSTM is sensitive to the variation of data, but the excessive punishment of boundary and internal jump by applying L2-norm impaired the prediction accuracy. All of the comparison models are preeminent, and the most cardinal obstacle that affects the prediction accuracy is that the feature learning is not awfully sufficient.

SWT has marvelous predictive performance among all the comparison models. According to the forecasting results of SWT-LSTM, it is observed that the models combining SWT and deep learning models greatly improve precision and accuracy. This method has eminent

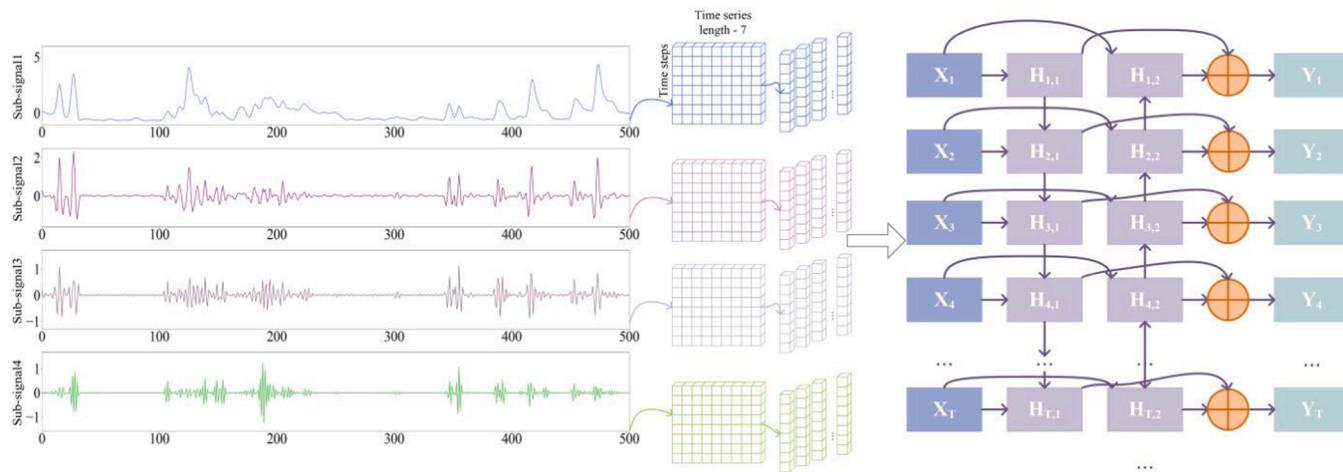


Fig. 10. The approach of SSA-BiLSTM.

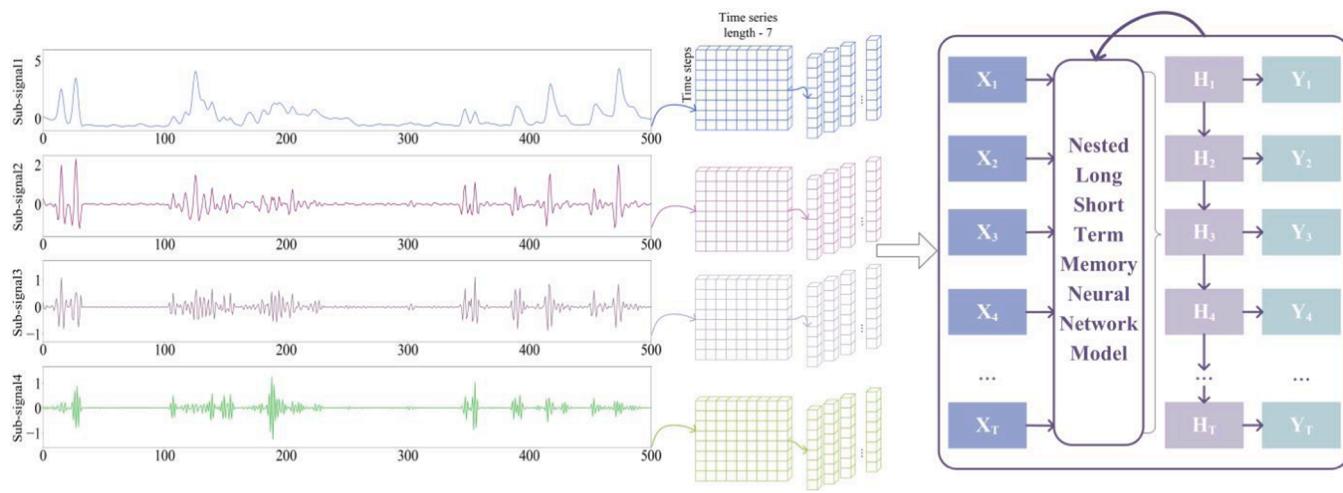


Fig. 11. The approach of SSA-NLSTM.

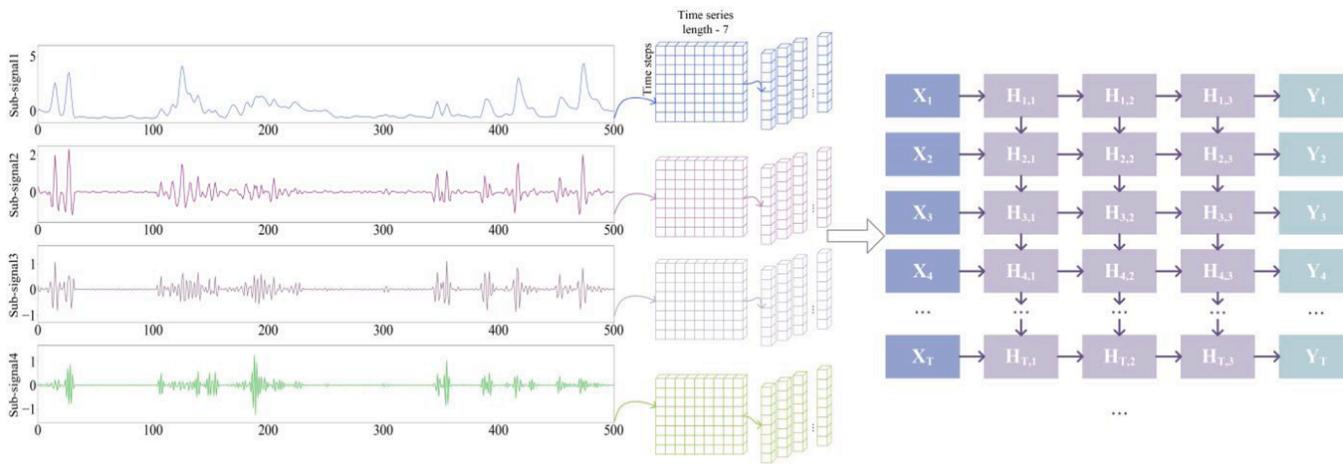
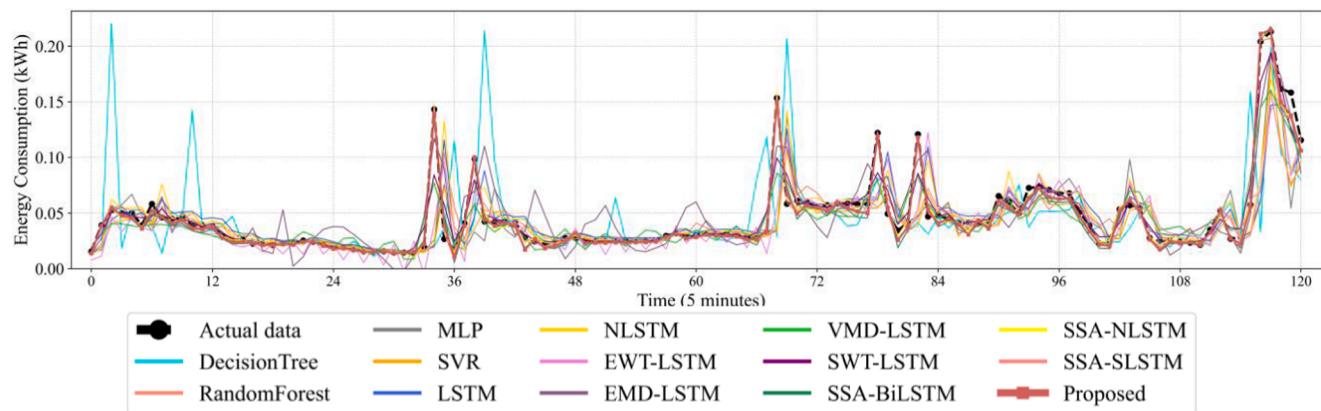


Fig. 12. The approach of SSA-SLSTM.

contributions in many domains. In many studies, SWT-LSTM demonstrated its capacity to process unstable, highly variable data [47,48]. It decomposed the raw data into high frequency and low frequency. Subsequently, the low-frequency component is decomposed abundantly. However, it caused the high-frequency component isn't adequately

handled. Insufficient decomposition confined further performance of SWT. As is plotted in Fig. 13., even if the mutation was caught, but there is still a big gap with the real data. Additionally, SWT is not adaptive. The wavelet basis function is selected to be utilized globally. This decomposition principle makes the wavelet still fluctuate slightly when



**Fig. 13.** The energy consumption forecasting results of all compared and proposed methods in this study.

the original data is stable.

In the smart grid system, the accurate prediction of signals crests plays a vital role in early warning. It prevents the collapse of the power grid system. In practice, the proposed algorithm greatly reduces the prediction error. On account of the signal processing algorithm work on the original data in advance, the model is proficient in master the data characteristic and has splendid learning ability Fig. 13 indicates that the combined model has better performance, and outperforms the compared methods, which justified that the proposed method has the capacity to process data with instability and volatility. Preserving considerable integrity of features before training is one of the reasons for the proposed model to ensure high precision. SSA removes Gaussian noise and impulse noise on the whole spectrum during denoising. The integrity of features and the effectiveness of denoising in the process enable SSA-PLSTM to maintain a good performance under different assessments. The generalization of the proposed model is also a key point for the proposed method to be applied to other domain areas, such as works done in [52–54].

## 7. Results discussion

From the experimental results (Figs. 8–13 and Tables 2–5), a series of detailed conclusions and discussions can be drawn as follows.

- 1) Different time steps are utilized to test the memory ability of multifarious models. To verify the performance of forecasting models in different time dimensions, the interval of 5 min, 10 min, 20 min and 30 min are used in our experiments. As the time step increases, higher requirements are put forward for the capacity of model learning and analysis. Apparently, the prediction errors of the models start to rise with the increase of time intervals. But the prediction result of SSA-PLSTM outperforms other benchmarked models at different time intervals. It demonstrates the powerful learning and non-linear training capacity in time series prediction. Since the proposed model adapts to the different time steps, the difficulty of sampling is greatly reduced. Additionally, the forecasting error of the proposed model is at a lower order of magnitudes from beginning to end. It helps companies stabilized the power supply, ensuring the normal life of individuals and groups. And we expect that our research conducive to energy transformation and electrical energy employment.
- 2) The universal applicability of the model reflects the learning and generalization ability to a great extent. We take the experiments in different scenarios. Firstly, the number of occupants is different. In the first household, there are 2 adults, 1 dog and 2 children. In the household 2 and 5, 2 adults live in it. 1 adult and 1 pensioner exist in household 4. Nobody lives in household 3. Secondly, the aggregate number of meters is different. There are 54, 20, 5, 6 and 26 electric

meters in the five households, respectively. Last, the duration of each household is not exactly the same. The data characteristics of different households are different, which makes the training of the model more difficult. The prediction results of first eight models are less different, and the error value of the rest experimental models decreases significantly.

## 8. Conclusion

In this article, we propose a hybrid AI model combining SSA and parallel long short term memory neural networks to forecast households' energy consumption. The individual household ECF is part of the AMI program and is crucial for the smart grid construction. The existing main issues existed in the literature include data noises and high volatility. In this study, we tackled these two main issues using a SSA decomposition process that extracts features as sub-signals and eliminates the data noise. The extracted features are treated separately with the same number of LSTM neural networks for forecasting. The PLSTM structure trains the LSTM neural networks concurrently and combines the forecasting results to produce the final output. The UK-DALE dataset is used for experimental verification. Thirteen existing forecasting models are included in the comparative study to show the superior performance of the proposed method.

The combination between signal processing and artificial intelligence brings higher prediction accuracy to time series forecasting. In addition, experimental results showed that the proposed SSA-PLSTM method is generally faster than the existing DL approaches. Moreover, the superior performance of the proposed method on different time horizons showed the generalization of the proposed method. We expect that the proposed model can be more widely adopted in other ECF research areas and other time series forecasting application scenarios.

The proposed model also has shortage and limitations. Although the model we proposed has achieved prominent prediction performance under the current data distribution, it is not a universal model applicable to different datasets. For example, after the prediction work of a certain household, the model needs to reload and re-train using the new datasets for new household energy consumption prediction. The focus point of our future work is to develop a universally applicable model for multiple households' energy consumption forecasting. In the terms of hardware, the other shortage of our work is that we deploy the training process on only one server. Thus, we intend to fully utilize the resource of computers with distributed learning and collaborative learning.

## Data Availability Statement

The source code and required data sets of the experiment can be obtained for free from <https://github.com/Fairy-09/HECO>.

## 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.

## Acknowledgement of Fundings

This work is partially supported by supported by the Ministry of Education (MOE) Singapore, Tier 1 funding under grant number R296000208133 and also supported in part by the National Natural Science Foundation of China under grant number 61972156 and Program for Innovative Research Team in Science and Technology in Fujian Province University.

## References

- [1] Q. Xiao, Y. Si, Time series prediction using graph model. 2017 3rd IEEE international conference on computer and communications (ICCC), IEEE, 2017.
- [2] Ke Yan, et al., Multi-step short-term power consumption forecasting with a hybrid deep learning strategy, *Energies* 11 (2018).
- [3] B. Zhang, Y. Du, E.G. Lim, L. Jiang, K. Yan, Design and simulation of Peer-to-Peer energy trading framework with dynamic electricity price, in: 2019 29th Australasian Universities Power Engineering Conference (AUPEC), IEEE, 2019, pp. 1–6.
- [4] V. Garg, N.K. Bansal, Smart occupancy sensors to reduce energy consumption, *Energy Build.* 32 (1) (2000) 81–87.
- [5] X. Zhou, X. Xu, W. Liang, Z. Zeng, Z. Yan, Deep learning enhanced multi-target detection for end-edge-cloud surveillance in smart IoT, *IEEE Internet Things J.* (2021).
- [6] X. Zhou, X. Yang, J. Ma, I. Kevin, K. Wang, Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT, *IEEE Internet Things J.* (2021).
- [7] P. Anand, C. Deb, K.e. Yan, J. Yang, D. Cheong, C. Sekhar, Occupancy-based energy consumption modelling using machine learning algorithms for institutional buildings, *Energy Build.* 252 (2021) 111478, <https://doi.org/10.1016/j.enbuild.2021.111478>.
- [8] T. Dietz, P.C. Stern, E.U. Weber, Reducing carbon-based energy consumption through changes in household behavior, *Daedalus* 142 (1) (2013) 78–89.
- [9] V. Masson, J. Hidalgo, A. Amossé, F. Belaid, E. Bocher, M. Bonhomme, et al., Urban Climate, Human behavior & Energy consumption: from LCZ mapping to simulation and urban planning (the MapUCE project). 9th International Conference on Urban Climate, 2015.
- [10] X. Lu, T. Lu, C.J. Kibert, M. Viljanen, Modeling and forecasting energy consumption for heterogeneous buildings using a physical–statistical approach, *Appl. Energy* 144 (2015) 261–275.
- [11] H. Zhou, Q. Liu, K. Yan, Y. Du, Deep Learning Enhanced Solar Energy Forecasting with AI-Driven IoT. *Wireless Communications and Mobile Computing*, 2021, 2021.
- [12] Y. Cao, X. Zhou, K. Yan, Deep Learning Neural Network Model for Tunnel Ground Surface Settlement Prediction Based on Sensor Data, *Mathe. Probl. Eng.* 2021 (2021).
- [13] Y. Wang, J. Wang, G.e. Zhao, Y. Dong, Application of residual modification approach in seasonal ARIMA for electricity demand forecasting: A case study of China, *Energy Policy* 48 (2012) 284–294.
- [14] V.G. Tran, V. Debusschere, S. Bacha, One week hourly electricity load forecasting using Neuro-Fuzzy and Seasonal ARIMA models, *IFAC Proc. Vol.* 45 (21) (2012) 97–102.
- [15] N. Wei, C. Li, X. Peng, F. Zeng, X. Lu, Conventional models and artificial intelligence-based models for energy consumption forecasting: A review, *J. Petrol. Sci. Eng.* 181 (2019) 106187, <https://doi.org/10.1016/j.petrol.2019.106187>.
- [16] H. Hu, L. Wang, L.u. Peng, Y.-R. Zeng, Effective energy consumption forecasting using enhanced bagged echo state network, *Energy* 193 (2020) 116778, <https://doi.org/10.1016/j.energy.2019.116778>.
- [17] H. Zamani Sabzi, S. Abudu, R. Alizadeh, L. Soltanishehat, N. Dilekli, J.P. King, Integration of time series forecasting in a dynamic decision support system for multiple reservoir management to conserve water sources, *Energy Sources Part A* 40 (11) (2018) 1398–1416.
- [18] Simone Gitto, et al., Forecasting national and regional level intensive care unit bed demand during COVID-19: The case of Italy, *Plos one* 16 (2) (2021).
- [19] Prajoy Podder, et al. Forecasting the Spread of COVID-19 and ICU Requirements. (2021).
- [20] S. Arslan, M.Y. Ozdemir, A. Ucar, Nowcasting and Forecasting the Spread of COVID-19 and Healthcare Demand in Turkey, a Modeling Study, *Front. Public Health* 8 (2020).
- [21] Y.-Y. Song, L.U. Ying, Decision tree methods: applications for classification and prediction, *Shanghai Arch. Psychiatry* 27 (2) (2015) 130.
- [22] A.J. Hill, G.R. Herman, R.S. Schumacher, Forecasting severe weather with random forests, *Mon. Weather Rev.* 148 (5) (2020) 2135–2161.
- [23] K. Kavaklıoglu, Modeling and prediction of Turkey's electricity consumption using Support Vector Regression, *Appl. Energy* 88 (1) (2011) 368–375.
- [24] Timo Koskela, et al., Time series prediction with multilayer perceptron, FIR and Elman neural networks, in: Proceedings of the World Congress on Neural Networks. INNS Press San Diego, USA, 1996.
- [25] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, *Sensors* 17 (4) (2017) 818, <https://doi.org/10.3390/s17040818>.
- [26] Samy Bengio, et al., Scheduled sampling for sequence prediction with recurrent neural networks, *arXiv preprint arXiv:1506.03099*, 2015.
- [27] F. Altché, A. de La Fortelle, An LSTM network for highway trajectory prediction. 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2017.
- [28] Q.B. Pham, T.-C. Yang, C.-M. Kuo, H.-W. Tseng, P.-S. Yu, Coupling Singular Spectrum Analysis with Least Square Support Vector Machine to Improve Accuracy of SPI Drought Forecasting, *Water Resour. Manage.* 35 (3) (2021) 847–868.
- [29] Ke Yan, et al., A hybrid LSTM neural network for energy consumption forecasting of individual households, *IEEE Access* 7 (2019) 157633–157642.
- [30] N. Wei, C. Li, X. Peng, Y. Li, F. Zeng, Daily natural gas consumption forecasting via the application of a novel hybrid model, *Appl. Energy* 250 (2019) 358–368.
- [31] T. Sun, T. Zhang, Y. Teng, Z. Chen, J. Fang, Monthly electricity consumption forecasting method based on X12 and STL decomposition model in an integrated energy system, *Mathe. Probl. Eng.* 2019 (2019) 1–16.
- [32] Z. Yuan, W. Wang, H. Wang, S. Mizzi, Combination of cuckoo search and wavelet neural network for midterm building energy forecast, *Energy* 202 (2020) 117728, <https://doi.org/10.1016/j.energy.2020.117728>.
- [33] N. Neeraj, et al., Long short-term memory-singular spectrum analysis-based model for electric load forecasting, *Electr. Eng.* (2020) 1–16.
- [34] S. Bharati, M.A. Rahman, R. Mondal, et al., Prediction of energy consumed by home appliances with the visualization of plot analysis applying different classification algorithm, in: *Frontiers in Intelligent Computing: Theory and Applications*, Springer, Singapore, 2020, pp. 246–257.
- [35] S. Bharati, P. Podder, M.R.H. Mondal, Visualization and prediction of energy consumption in smart homes, *Int. J. Hybrid Intell. Syst.* 16 (2) (2020) 81–97.
- [36] N. Golyandina, A. Korobeynikov, A. Zhigljavsky, *Singular spectrum analysis with R*, Springer, New York, 2018.
- [37] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
- [38] G. Memarzadeh, F. Keynia, Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm, *Electr. Power Syst. Res.* 192 (2021) 106995, <https://doi.org/10.1016/j.epsr.2020.106995>.
- [39] J. Kelly, W. Knotternbelt, The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes, *Sci. Data* 2 (1) (2015) 1–14.
- [40] A.F. Mashaly, A.A. Alazba, MLP and MLR models for instantaneous thermal efficiency prediction of solar still under hyper-arid environment, *Comput. Electron. Agric.* 122 (2016) 146–155.
- [41] Sebastian Ruder, An overview of gradient descent optimization algorithms, *arXiv preprint arXiv:1609.04747* (2016).
- [42] H. Brendan McMahan, Matthew Streeter, Delay-tolerant algorithms for asynchronous distributed online learning. (2014).
- [43] J. Gilles, Empirical wavelet transform, *IEEE Trans. Signal Process.* 61 (16) (2013) 3999–4010.
- [44] Norden E. Huang, et al., The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *Proc. Roy. Soc. London. Series A: Mathe., Phys. Eng. Sci.* 454 (1971) (1998) 903–995.
- [45] K. Dragomiretskiy, D. Zosso, Variational mode decomposition, *IEEE Trans. Signal Process.* 62 (3) (2014) 531–544.
- [46] G.P. Nason, B.W. Silverman, The stationary wavelet transform and some statistical applications, in: *Wavelets and Statistics*, Springer, New York, NY, 1995, pp. 281–299.
- [47] Tushar Tyagi, Parth Gupta, Prabhisek Singh, A Hybrid Multi-focus Image Fusion Technique using SWT and PCA. 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2020.
- [48] Suyi Li, Guangda Liu, Zhenbao Lin, Comparisons of wavelet packet, lifting wavelet and stationary wavelet transform for de-noising ECG. 2009 2nd IEEE International Conference on Computer Science and Information Technology, IEEE, 2009.
- [49] Sima Siami-Namin, Neda Tavakoli, Akbar Siami Namin, The performance of LSTM and BiLSTM in forecasting time series. 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019.
- [50] Xiaolei Ma, et al., Forecasting transportation network speed using deep capsule networks with nested LSTM models, *IEEE Trans. Intell. Transp. Syst.* 22 (8) (2021) 4813–4824.
- [51] Xunsheng Du, et al., Stacked LSTM deep learning model for traffic prediction in vehicle-to-vehicle communication. 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), IEEE, 2017.
- [52] K. Yan, Chiller fault detection and diagnosis with anomaly detective generative adversarial network, *Build. Environ.* (2021) 107982.
- [53] X. Zhou, X. Xu, W. Liang, Z. Zeng, S. Shimizu, L.T. Yang, Q. Jin, Intelligent Small Object Detection Based on Digital Twinning for Smart Manufacturing in Industrial CPS, *IEEE Trans. Ind. Inf.* (2021), <https://doi.org/10.1109/TII.2021.3061419>.
- [54] X. Zhou, W. Liang, S. Shimizu, J. Ma, Q. Jin, Siamese Neural Network Based Few-Shot Learning for Anomaly Detection in Industrial Cyber-Physical Systems, *IEEE Trans. Ind. Inf.* 17 (8) (2021) 5790–5798, <https://doi.org/10.1109/TII.2020.3047675>.