Hybrid Short-term Load Forecasting Method Based on Empirical Wavelet Transform and Bidirectional Long Short-term Memory Neural Networks

Xiaoyu Zhang, Stefanie Kuenzel, Nicolo Colombo, and Chris Watkins

***Abstract*—Accurate short-term load forecasting is essential to modern power system and smart grids. The utility can better implement demand-side management and operate power system stably with a reliable load forecasting system. The load demand contains a variety of different load components, and different loads operate with different frequencies. The conventional load forecasting methods, e.g., linear regression (LR), auto-regressive integrated moving average (ARIMA), deep neural network, ignore the frequency domain and can only use time-domain load demand as inputs. To make full use of both time-domain and frequency-domain features of the load demand, a load forecasting method based on hybrid empirical wavelet transform (EWT) and deep neural network is proposed in this paper. The proposed method first filters noises via wavelet-based denoising technique, and then decomposes the original load demand into several sub-layers to show the frequency features while the time-domain information is preserved as well. Then, a bidirectional long short-term memory (LSTM) method is trained for each sub-layer independently. For better tuning the hyperparameters, a Bayesian hyperparameter optimization (BHO) algorithm is adopted in this paper. Three case studies are designed to evaluate the performance of the proposed method. From the results, it is found that the proposed method improves the prediction accuracy compared with other load forecasting models.**

***Index Terms*—Load forecasting, empirical wavelet transform (EWT), recurrent neural network, data denoising, Bayesian hyperparameter optimization (BHO).**

NOMENCLATURE

A. Indices

m	Index of sample
n	Index of sub-layers
t	Index of time

B. Variables

$\mathcal{X}$	Sample domain
α	Magnitude threshold of empirical wavelet transform (EWT)
δ	Frequency distance threshold of EWT
η	Total number of input data in long short-term memory (LSTM)
$\sigma(\cdot)$	Activation function of neural network
ϵ_m	White Gaussian noise
σ_{noise}	Intensity of noise
$\varphi_h(\cdot)$	Function of low-pass filter
$\varphi_g(\cdot)$	Function of high-pass filter
$\rho_T(\cdot)$	Thresholding function of discrete wavelet transform (DWT)
ω_n	Support boundary
$a(k)$	Approximation coefficient
$\tilde{\boldsymbol{C}}_t$	Candidate cell state at current time step
$\boldsymbol{C}_t$	Cell state at current time step
$\boldsymbol{C}_{t-1}$	Cell state at previous time step
$d(k)$	Detail coefficient
d	Sequence size of each example
$f^*(t)$	Original data
$f^*(m)$	Original data in discrete form
$f(t)$	Denoised data
$f(m)$	Denoised data in discrete form
f_{sample}	Sampling frequency
F_ω	Frequency spectrum
$\boldsymbol{F}_t$	Data vector of forget gate at time t
h	Total number of hidden units in LSTM
$\vec{\boldsymbol{H}}_t$	Hidden vector for forward layer at time t
$\overleftarrow{\boldsymbol{H}}_t$	Hidden vector for backward layer at time t
$\boldsymbol{H}_t$	Total hidden vector layer at time t
$\boldsymbol{I}_t$	Data vector of input gate at time t
L	Total number of maxima

Manuscript received: May 8, 2021; revised: November 23, 2021; accepted: February 22, 2022. Date of CrossCheck: February 22, 2022. Date of online publication: XX XX, XXXX.

This work was supported by the Leverhulme Trust, U.K.

X. Zhang is with School of Artificial Intelligence, Anhui University, Hefei 230601, China, and he is also with the Department of Electronic Engineering, Royal Holloway, University of London, Egham Hill, Egham TW20 0EX, UK (e-mail: xiaoyu.zhang@rhul.ac.uk).

S. Kuenzel (corresponding author) is with the Department of Electronic Engineering, Royal Holloway, University of London, Egham Hill, Egham TW20 0EX, UK (e-mail: Stefanie.Kuenzel@rhul.ac.uk).

N. Colombo and C. Watkins are with the Department of Computer Science, Royal Holloway, University of London, Egham Hill, Egham TW20 0EX, UK (e-mail: Nicolo.Colombo@rhul.ac.uk; C.J.Watkins@rhul.ac.uk).

DOI: 10.35833/MPCE.2021.000276

M	Total sampling number of denoised data in discrete form
$\mathcal{M}$	Surrogate function
N	Number of sub-layers
$\boldsymbol{O}_t$	Data vector of output gate at time t
q	Total number of unit outputs in bidirectional LSTM (BLSTM)
S	Total decomposition level of DWT denoising
$\mathcal{S}(\cdot)$	Acquisition function
SS_{RES}	Sum squared regression error
SS_{TOT}	Sum squared total error
Thr	Thresholding value
T_n	Transition area width
T_L	Look-back steps

I. Introduction

LOAD forecasting is vital for power systems, especially for real-time energy management. The prediction results will influence the plans of utility providers, including deciding the amount of energy to be generated and purchased at the next stage. However, load forecasting is challenging due to the uncertainty and complexity of load demands. Conventional load forecasting techniques such as linear regression (LR) and auto-regressive integrated moving average (ARIMA) only extract time-domain features of the load demands. Since a variety of load components with different frequencies are contained in the load curve, the load demand is highly nonlinear and non-stationary. These characteristics of the original load demand make the prediction of conventional models less accurate. In addition, artificial intelligence (AI) based load forecasting methods, especially the recurrent neural network (RNN), have achieved desirable accuracy in recent years [1], [2]. RNN models have memory units that can learn not only the current input features but also the information from the past. This characteristic is highly suitable for forecasting tasks. Although RNN can map nonlinear features like conventional approaches, it cannot learn frequency-domain information. Hence, a hybrid short-term load forecasting (STLF) method that can extract both time-domain and frequency-domain features of load demands with high adaptivity should be proposed.

In recent years, the prediction performance and reliability of STLF models have been improved significantly with the development of AI techniques [3]. Modern STLF models can be divided into single STLF models and hybrid STLF models.

Single STLF models can be divided into learning-based models including regression-based models, deep learning based models, and machine learning based models. Conventional regression-based models include LR [4], [5], gradient boosting regression (GBR) [6], and ARIMA [7]. Deep learning based models utilize multiple hidden layers to evaluate the nonlinear correlations between their inputs and outputs. Convolutional neural network (CNN) [2], long short-term memory (LSTM) [8], and extreme learning machine (ELM) [9] have been employed to the STLF tasks and could achieve high prediction accuracy. Among all deep learning based models, LSTM and its variant bidirectional LSTM (BLSTM) attract the most attention of researchers for the superior performance in processing sequence data. The memory cell enables the predictor to better understand sequence information and utilize knowledge learnt from the past to make prediction for future. In addition, the probabilistic STLF methods such as quantile regression neural network [1] and sparse penalized quantile regression [10] are introduced in related works. Compared with normal point-to-point predictions, probabilistic STLF methods could predict the area where the future load may locate and better capture the load variation.

Hybrid STLF models have attracted more and more attention in recent years for their high adaptive and precise prediction accuracy. These models usually consist of two or more single methods to better extract the features of inputs and increase the prediction accuracy. Specifically, hybrid deep learning based models that combine the micro-clustering (MC) techniques are introduced in [8], [11]-[15]. Normally, the electric load clustering consists of four steps [3], i.e., pre-processing, clustering and centroid, constructing the representative load curves, and assessing the clustering performance. Whilst traditional MC-based STLF methods [8] cluster the load curves over the period and ignore the load variations of different periods, [11] and [12] propose an STLF model that combines the BLSTM with MC technique smoothly. The load demand data of each hour are clustered into several categories by implementing either supervised or unsupervised MC methods, and then a specific BLSTM model is trained for each cluster. As a result, the MC-based BLSTM methods give better predictions for the hours with more spikes [12]. However, the methods discussed above encounter a bottleneck as only the time-domain information of the load is utilized, while the rich frequency-domain information is overlooked. The hybrid methods that combine decomposition techniques and deep learning based models can utilize both time-domain and frequency-domain information. Decomposition methods include empirical mode decomposition (EMD) [16], variational mode decomposition (VMD) [17], [18], seasonal and trend decomposition using loess (STL) [19], and empirical wavelet transform (EWT) [20]. The EMD-based STLF methods are introduced in [16]. As an adaptive nonlinear decomposition method, EMD decomposes the original signal into a series of intrinsic mode functions (IMFs) using Hilbert-Huang transform [21], and each IMF is an amplitude modulation-frequency modulation (AM-FM) signal. However, as a purely data-driven method, EMD lacks the mathematical definition, so it is difficult to understand the decomposition results. Moreover, the decomposed signals will diverge at the endpoints and are highly sensitive to noises. The VMD-based STLF methods are presented in [17], [18]. As an alternative algorithm of EMD, VMD is a non-recursive and adaptive decomposition estimation method to decompose the original signal into several mode functions with specific bandwidth in the frequency domain [22]. In [17], a hybrid STL-VMD-LSTM STLF method is introduced to extract both seasonal and frequency features of the elec-

tric load. A hybrid VMD adaptive neuro fuzzy inference system (ANFIS) forecasting model is proposed in [23], where the model takes advantage of both mode decomposition and fuzzy logic principles. The last decomposition algorithm EWT combines the strength of the wavelet's mathematical definition with the flexibility of EMD [20], and a detailed introduction of the EWT technique will be introduced in Section II.

Although, as illustrated, there is a wealth of work available in the literature, the existing STLF models still have some knowledge gaps that can be filled. Firstly, the hybrid deep learning with EMD or VMD in the literature either lacks mathematical definition or has low adaptivity, so a new hybrid STLF method that takes advantages of both EMD and VMD should be proposed. Secondly, electric spikes and other noises would influence the training process and the prediction accuracy, so a proper denoising technique should be selected to process the original data.

In this paper, a novel hybrid denoising EWT-BLSTM-Bayesian hyperparameter optimization (BHO) STLF method is proposed, which combines mode decomposition with BLSTM to better extract the time-domain and frequency-domain features of the electric load. The contributions of this paper are detailed as follows.

1) A hybrid STLF method that combines the EWT decomposition with a BLSTM deep neural network is proposed to make multi-step predictions.

2) A wavelet-based denoising technique is proposed to eliminate the electric spikes.

3) A BHO algorithm is proposed, which can find optimal hyperparameters with fast speed and adjust hyperparameters to different sub-layers.

The remainder of this paper is organized as follows. The proposed load forecasting method is demonstrated in Section II. The experiment setup is discussed in Section III. In Section IV, four case studies are implemented, which compare the proposed load forecasting method and other methods and then evaluate the parameters that achieve the best performance. The conclusion and future work are provided in Section V.

II. Proposed Load Forecasting Method

In this section, the overall prediction system and the corresponding methodologies are introduced. As presented in Fig. 1, the proposed method is divided into five steps and described as follows.

Step 1: the first step is data pre-processing and denoising. The original electric load is input to the STLF model, and data cleaning is applied to the original dataset to populate the missing features. Then, a max-min scaling function is applied to the original dataset to limit the range of data between 0 and 1. Finally, a discrete wavelet transform (DWT) data based denoising algorithm is applied to the data to remove the noise.

Step 2: a sliding window is introduced to enable the proposed method to make real-time forecasts. The length of the sliding window is denoted as W, which is chosen as one week in this paper. At the beginning of the training, the first W data is included in the window, and the load is predicted at $W+L_F$, where L_F is the forecasting step. Then, the sliding window will move smoothly step by step and repeat the training process.

Step 3: the denoised electric load is decomposed into N sub-layers via the EWT decomposition algorithm. As incidated in Fig. 1, an example with 9 sub-layers is presented to show the decomposed components S_1-S_9 from the orginal load curve.

Step 4: N BLSTM prediction models are constructed, and each BLSTM neural network model is trained for one sub-layer while BHO method is employed to find the optimal hyperparameters.

Step 5: the prediction results for all sub-layers are reconstructed to present the final load forecasting results. Repeat *Steps* 2-5 util reaching the end of testing dataset.

A. Signal Denoising with Wavelets

The original load data contain a significant amount of noise, which is generated from various sources such as the electric spikes of electric appliances and intermittent penetration of distributed generators. In addition, the measurement devices such as smart meters and supervisory control and data acquisition (SCADA) also produce electronic noise. The high-frequency noise in the measured feeder load demand is a severe issue that influences the performance of load forecasting. DWT could effectively analyze the non-stationary signals and reduce the high-frequency noise [24].

The theory of the DWT-based denoising technique is to decompose the original data into the high-frequency and low-frequency components, and a suitable threshold of the high-frequency components is determined for denoising purpose, finally, the signal is reconstructed again. Sampling the original data $f^*(t)$ with frequency f_{sample} can obtain the discrete signal $f^*(m)$, $m=1,2,\ldots,M$. The purpose of the signal denoising is to remove the noise and find the best estimation of the underlying signal $f(m)$:

$$f^*(m)=f(m)+\sigma_{noise}\epsilon_m \quad m=1,2,\ldots,M \tag{1}$$

A two-level DWT wavelet decomposition process is shown in Fig. 2. From the figure, a signal can be decomposed into two coefficients: approximation coefficient a and detailed coefficient d. At level 1, $f^*(m)$ is decomposed into a_1 and d_1, and a_1 is then decomposed into a_2 and d_2 further. The decompssition level S is set to be 2 in this paper. The denoising approach includes three steps: signal decomposition, denoising, and reconstruction.

1) Signal decomposition. The original load demand, which is the noisy signal, is decomposed via the DWT, as shown in Fig. 2. The original discrete signal is passed through a series of high-pass filters (HPFs) and low-pass filters (LPFs). The detail coefficients of level i $d_i(k)$ are given via a HPF and the approximation coefficients of level i $a_i(k)$ are given via a LPF. The decomposition functions of level are expressed as:

$$a_i(k)=\sum_{m=1}^{M} f^*(m)\varphi_g(2m-k) \tag{2}$$

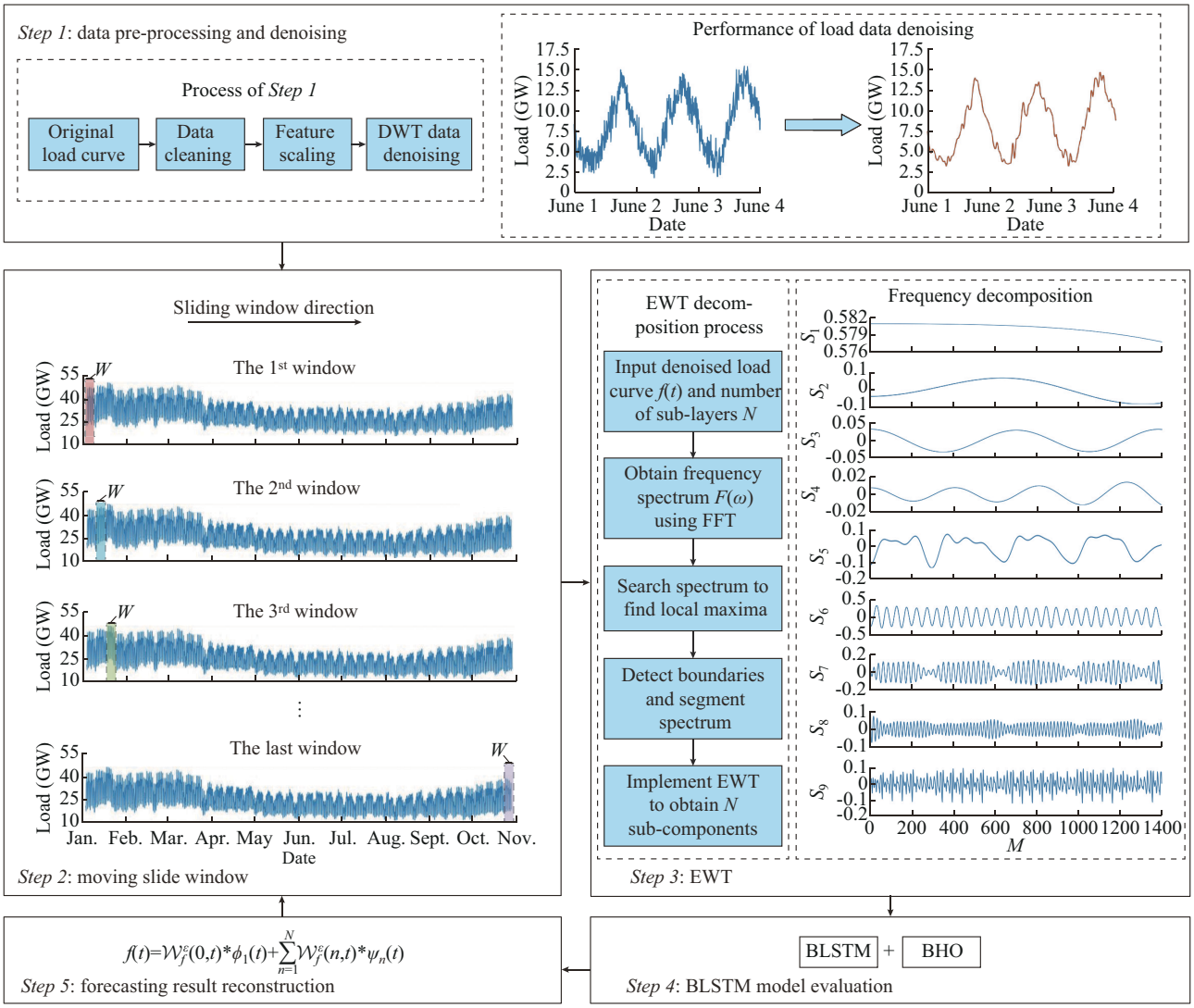

Fig. 1. Block-diagram of proposed hybrid STLF method.

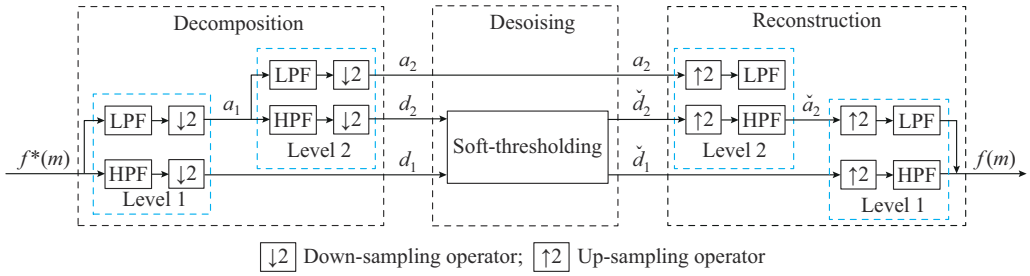

Fig. 2. Block diagram of signal denoising with wavelets.

$$d_i(k)=\sum_{m=1}^{M} f^*(m)\varphi_h(2m-k) \tag{3}$$

2) Denoising. It is essential to determine a suitable threshold *THR* for data denoising, and a thresholding function $\rho_T(x)$ is required. Thresholding can be divided into hard and soft thresholding. For hard thresholding, the values that exceed the threshold would be set to be 0. For soft thresholding, the magnitude of coefficients greater than the threshold is softened. In Fig.2, the symbol " ˇ " represents the coefficients that are softened. The noise level δ_{mad} is first estimated from the detail coefficients by median absolute deviation:

$$\delta_{\text{mad}}=\frac{Median\{d_i\}}{0.6745} \tag{4}$$

$$THR=\delta_{\text{mad}}\sqrt{\ln(M)} \tag{5}$$

After the threshold *THR* is determined, the soft thresholding function is applied to reduce the magnitude of the coefficient, which is defined as:

$$\rho_T(x)=\begin{cases}x-THR & x\geq THR\\ x+THR & x\leq -THR\\ 0 & |x|<THR\end{cases} \tag{6}$$

3) Reconstruction. The coefficients after the soft thresholding are reconstructed via inverse discrete wavelet transform (IDWT). In Fig. 1, *Step 1* compares the original load demand curve and the denoised load demand curve. It is observed that the noise and spikes from the original data are successfully cleared.

B. EWT

After the data are denoised via DWT, the denoised data $f(t)$ are decomposed into N sub-layers via EWT. In [25], the target of EWT is to extract multiple sub-layers by constructing adaptive wavelets. The EWT decomposition process is performed in the following steps.

Step 1: apply fast fourier transform (FFT) to the denoised data $f(t)$ to obtain the frequency spectrum $F(\omega)$.

Step 2: search $F(\omega)$ to find N local maxima $\partial=\{\partial_n\}_{n=1,2,\ldots,N}$ and the corresponding frequencies $\omega=\{\omega_n\}_{n=1,2,\ldots,N}$ by using the magnitude threshold α and frequency distance thresholds δ. α is set to be 3% of the fundamental magnitude to detect the significant frequencies, and δ is set to be 8 Hz to avoid the overestimation [26].

Step 3: segment the frequency spectrum $[0, f_{sample}/2]$ into N segments, and the boundary Ω_n is the central line between two neighbouring local maxima, which can be calculated as:

$$\Omega_n=\frac{\omega_n+\omega_{n+1}}{2} \tag{7}$$

Step 4: build N wavelet filters, including one empirical scaling function (LPF) and N–1 empirical wavelets (band-pass filters), The scaling and wavelet functions, i.e., $\hat{\phi}_n(\omega)$ and $\hat{\psi}_n(\omega)$, are defined in (8) and (9), respectively.

$$\hat{\phi}_n(\omega)=\begin{cases}1 & |\omega|<(1-\gamma)\omega_n\\ \cos\left(\frac{\pi}{2}\beta\frac{|\omega|-(1-\gamma)\omega_n}{2\gamma\omega_n}\right) & (1-\gamma)\omega_n\leq|\omega|\leq(1+\gamma)\omega_n\\ 0 & |\omega|>(1-\gamma)\omega_n\end{cases} \tag{8}$$

$$\hat{\psi}_n(\omega)=\begin{cases}1 & (1+\gamma)\omega_n\leq|\omega|<(1-\gamma)\omega_{n+1}\\ \cos\left(\frac{\pi}{2}\beta\frac{|\omega|-(1-\gamma)\omega_{n+1}}{2\gamma\omega_{n+1}}\right) & (1-\gamma)\omega_{n+1}\leq|\omega|\leq(1+\gamma)\omega_{n+1}\\ \sin\left(\frac{\pi}{2}\beta\frac{|\omega|-(1-\gamma)\omega_n}{2\gamma\omega_n}\right) & (1-\gamma)\omega_n\leq|\omega|<(1+\gamma)\omega_n\\ 0 & \text{Otherwise}\end{cases} \tag{9}$$

where the arbitrary function $\beta(x)$ and the ratio γ are defined as:

$$\begin{cases}\beta(x)=0 & x<0\\ \beta(x)+\beta(1-x)=1 & 0\leq x\leq 1\\ \beta(x)=1 & x>1\end{cases} \tag{10}$$

$$\gamma<\min_n\left(\frac{\omega_{n+1}-\omega_n}{\omega_{n+1}+\omega_n}\right) \tag{11}$$

Step 5: perform scaling and wavelet functions shown in (12) and (13), respectively, to extract the approximate and detailed coefficients.

$$\mathcal{W}_f^{\varepsilon}(0,t)=\left\langle f,\phi_1\right\rangle=\int f(\tau)\bar{\phi}_1(\tau-t)\mathrm{d}\tau=\left(\hat{f}(\omega)\bar{\hat{\phi}}_1(\omega)\right)^{\vee} \tag{12}$$

$$\mathcal{W}_f^{\varepsilon}(n,t)=\left\langle f,\psi_n\right\rangle=\int f(\tau)\bar{\psi}_n(\tau-t)\mathrm{d}\tau=\left(\hat{f}(\omega)\bar{\hat{\psi}}_n(\omega)\right)^{\vee} \tag{13}$$

where $\langle,\rangle$ denotes the inner product; the symbols $\hat{}$ and $^{\vee}$ denote the Fourier transform and its inverse, respectivey; and $\bar{\phi}_1(\tau-t)$ and $\bar{\psi}_n(\tau-t)$ are the conjugate complex numbers of $\phi_1(\tau-t)$ and $\psi_n(\tau-t)$, respectively.

Step 6: compute the sub-band signals. The approximation sub-band signal $f_0(t)$ and the n^{th} detail sub-band signal $f_n(t)$ can be computed by (14) and (15), respectively.

$$f_0(t)=\mathcal{W}_f^{\varepsilon}(0,t)*\phi_1(t) \tag{14}$$

$$f_n(t)=\mathcal{W}_f^{\varepsilon}(n,t)*\psi_n(t) \tag{15}$$

where $*$ denotes the convolution operation.

The EWT reconstruction, which is also called inverse empirical wavelet transform (IEWT), is used to reconstruct the sub-layers to $f(t)$. $f(t)$ can be reconstructed via the reconstruction function as:

$$\begin{aligned}f(t)&=f_0(t)+\sum_{n=1}^{N}f_n(t)=\\ &\mathcal{W}_f^{\varepsilon}(0,t)*\phi_1(t)+\sum_{n=1}^{N}\mathcal{W}_f^{\varepsilon}(n,t)*\psi_n(t)=\\ &\left(\hat{\mathcal{W}}_f^{\varepsilon}(0,\omega)\hat{\phi}_1(\omega)+\sum_{n=1}^{N}\hat{\mathcal{W}}_f^{\varepsilon}(n,\omega)\hat{\psi}_n(\omega)\right)^{\vee}\end{aligned} \tag{16}$$

C. BLSTM

LSTM model was firstly proposed in 1997 [27]. As shown in Fig. 3(a), in LSTM, the hidden state in traditional RNN is replaced by the memory cell $\boldsymbol{C}_t\in\mathbf{R}^{h\times 1}$ and three gates, i.e., the input gate $\boldsymbol{I}_t\in\mathbf{R}^{h\times 1}$, the forget gate $\boldsymbol{F}_t\in\mathbf{R}^{\eta\times h}$, and the output gate $\boldsymbol{O}_t\in\mathrm{R}^{h\times 1}$. The output of the previous time step $\boldsymbol{h}_{t-1}\in\mathbf{R}^{\eta\times 1}$ and the input sequence of the current time step $\boldsymbol{X}_t\in\mathbf{R}^{\eta\times 1}$ are adopted as the input of the gates. These gates are controlled by the sigmoid activation function $\sigma(\cdot)$, where the information is reserved when the activation output is close to 1, and the information is eliminated when the activation output approaches 0. As for the memory cell $\boldsymbol{C}_t$, a candidate memory cell $\tilde{\boldsymbol{C}}_t\in\mathbf{R}^{h\times 1}$ is computed at first. The only difference between $\tilde{\boldsymbol{C}}_t$ and these gates is that $\tilde{\boldsymbol{C}}_t$ utilizes a Tanh activation function $\tanh(\cdot)$ ranging from −1 to 1. Finally, the memory cell $\boldsymbol{C}_t$ is generated by combining $\tilde{\boldsymbol{C}}_t$ and $\boldsymbol{I}_t$ and combining the previous memory cell $\boldsymbol{C}_{t-1}$ with $\boldsymbol{I}_t$ with $\boldsymbol{F}_t$, where $\boldsymbol{I}_t$ decides how much data from $\tilde{\boldsymbol{C}}_t$ is useful, and $\boldsymbol{F}_t$ decides how much information from the old memory cell is retained. The detail formulas are presented as:

$$\boldsymbol{I}_t=\sigma(\boldsymbol{W}_{xi}\boldsymbol{X}_t+\boldsymbol{W}_{hi}\boldsymbol{h}_{t-1}+\boldsymbol{b}_i) \tag{23}$$

$$\boldsymbol{F}_t=\sigma(\boldsymbol{W}_{xf}\boldsymbol{X}_t+\boldsymbol{W}_{hf}\boldsymbol{h}_{t-1}+\boldsymbol{b}_f) \tag{24}$$

$$\boldsymbol{O}_t=\sigma(\boldsymbol{W}_{xo}\boldsymbol{X}_t+\boldsymbol{W}_{ho}\boldsymbol{h}_{t-1}+\boldsymbol{b}_o) \tag{25}$$

$$\tilde{\boldsymbol{C}}_t=\tanh(\boldsymbol{W}_{xc}\boldsymbol{X}_t+\boldsymbol{W}_{hc}\boldsymbol{h}_{t-1}+\boldsymbol{b}_c) \tag{26}$$

$$\boldsymbol{C}_t = \boldsymbol{F}_t \odot \boldsymbol{C}_{t-1} + \boldsymbol{I}_t \odot \tilde{\boldsymbol{C}}_t \tag{27}$$

$$\boldsymbol{h}_t = \boldsymbol{O}_t \odot \tanh(\boldsymbol{C}_t) \tag{28}$$

where $\odot$ represents the element-wise multiplication; $\boldsymbol{W}_{xi}$, $\boldsymbol{W}_{xf}$, $\boldsymbol{W}_{hi}$, $\boldsymbol{W}_{xo}$, $\boldsymbol{W}_{xc} \in \mathbf{R}^{h\times\eta}$ and $\boldsymbol{W}_{hf}$, $\boldsymbol{W}_{hf}$, $\boldsymbol{W}_{ho}$, $\boldsymbol{W}_{hc} \in \mathbf{R}^{h\times h}$ are the weight matrices; and $\boldsymbol{b}_i, \boldsymbol{b}_f, \boldsymbol{b}_o, \boldsymbol{b}_c \in \mathbf{R}^{h\times 1}$ are the bias vectors.

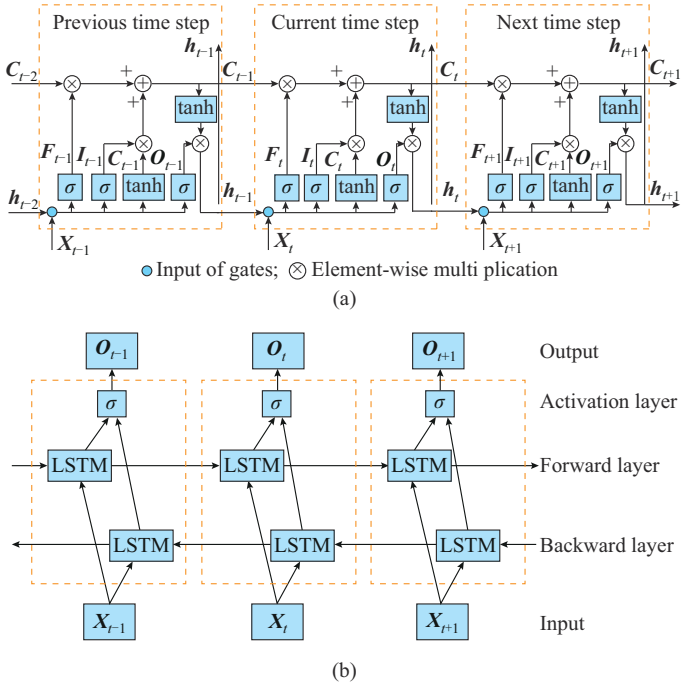

Fig. 3. LSTM and BLSTM models. (a) LSTM. (b) BLSTM.

The main disadvantage of conventional LSTM model is that it can only utilize the information from the past. To overcome this drawback, the BLSTM is proposed in 1997 [28]. As shown in Fig. 3(b), unlike the unidirectional LSTM, BLSTM can utilize both previous and future information with two separate LSTM layers, i.e., a forward LSTM layer that passes information from the past to future and a backward LSTM layer that passes information from the future to past. As the data collected by the smart meter are sequence data in time domain, the BLSTM model is especially suitable to process such kind of sequence data for following reasons. Firstly, the amount of input information that a BLSTM model can reach is larger than standard LSTM model, and the rich information makes BLSTM have much higher data representation capability [29]. Secondly, the BLSTM models do not follow the recursive procedure, and this characteristic enables these models make predictions on stochastic and intermittent data with high accuracy [11].

In a BLSTM structure, given a minibatch input $\boldsymbol{X}_t \in \mathbf{R}^{d\times\eta}$, the forward and backward hidden states at time step t, i.e., $\vec{\boldsymbol{H}}_t \in \mathbf{R}^{h\times\eta}$ and $\overleftarrow{\boldsymbol{H}}_t \in \mathbf{R}^{h\times\eta}$, can be expressed as:

$$\vec{\boldsymbol{H}}_t = \phi(\boldsymbol{W}_{xh}^f \boldsymbol{X}_t + \boldsymbol{W}_{hh}^f \vec{\boldsymbol{H}}_{t-1} + \boldsymbol{b}_h^f) \tag{29}$$

$$\overleftarrow{\boldsymbol{H}}_t = \phi(\boldsymbol{W}_{xh}^b \boldsymbol{X}_t + \boldsymbol{W}_{hh}^b \overleftarrow{\boldsymbol{H}}_{t-1} + \boldsymbol{b}_h^b) \tag{30}$$

where the superscripts f and d represent the forward and backward hidden states; $\boldsymbol{W}_{xh}^f, \boldsymbol{W}_{xh}^b \in \mathbf{R}^{h\times d}$ and $\boldsymbol{W}_{hh}^f, \boldsymbol{W}_{hh}^b \in \mathbf{R}^{h\times h}$ are the weights of the model; and $\boldsymbol{b}_h^f, \boldsymbol{b}_h^b \in \mathbf{R}^{h\times\eta}$ are the biases of the model. Then, by integrating the forward and backward hidden states, the hidden state is obtained as $\boldsymbol{H}_t \in \mathbf{R}^{2h\times\eta}$. Finally, $\boldsymbol{H}_t$ is fed to the output layer to compute the output of BLSTM block $\boldsymbol{O}_t \in \mathbf{R}^{\eta\times q}$:

$$\boldsymbol{H}_t = [\vec{\boldsymbol{H}}_t^{\mathrm{T}} \quad \overleftarrow{\boldsymbol{H}}_t^{\mathrm{T}}]^{\mathrm{T}} \tag{31}$$

$$\boldsymbol{O}_t = \boldsymbol{W}_{hq} \boldsymbol{H}_t + \boldsymbol{b}_q \tag{32}$$

where $\boldsymbol{W}_{hq} \in \mathbf{R}^{q\times 2h}$ is the weight of the model; and $\boldsymbol{b}_q \in \mathbf{R}^{q\times\eta}$ is the bias of the output layer.

D. BHO

Training and optimizing a deep learning model is a complex process that involves a great number of hyperparameters and regularization terms. Hyperparameter optimization is essential for training neural networks as it aims to find the hyperparameters that return the best accuracy or performance given a dataset. However, the hyperparameter tuning process is normally a "black box" function, which requires the examiners to keep querying the model and obtain the feedback of model performance. The hyperparameter optimization problem for a "lack box" function $G(x)$ can be formalized as:

$$x_M = \arg\min_{x\in X} G(x) \tag{33}$$

where x_M is the optimal hyperparameter set; and X is the candidate set. The target of the function is to find x_M to minimize $G(x)$. Grid search is the most fundamental hyperparameter tuning method [30], where a space is defined for each hyperparameter at first, and then the algorithm exhaustively searches this space sequentially and trains a model for every possible combination of hyperparameter values. The drawback of grid search is that the number of training models increases exponentially when hyperparameters increase.

Compared with the above methods, a novel BHO was proposed in 2011 [31]. Instead of searching the hyperparameter space blindly, BHO creates a prior distribution model, and then the model is optimized with given information to better fit the actual distribution. And it can use the results from the previous iteration to decide the next candidate value of hyperparameter. Hence, the BHO is much more efficient and less time-consuming as it selects the optimal hyperparameter in an informed manner and better utilizes the past information.

The central methodology of the BHO method is to construct a surrogate probability model to select hyperparameters to minimize the original objective function. Providing a sample domain $\mathcal{X}$, the true objective function $G(x)$ to be optimized is approximated with a surrogate function $\mathcal{M}$. $\mathcal{M}$ is initialized with a small data group from $\mathcal{X}$, and an acquisition function $\mathcal{S}$ is adopted to choose the next point to query. A variety of surrogate functions $\mathcal{M}$ are introduced in [32], including Gaussian processes (GPs), random forests, and tree-structured parzen estimators (TPEs) [33]. In this work, GP is employed as the surrogate function. The GP is a stochastic process that is a collection of random variables in time or space domain, such that each linear finite-dimension-

al restriction is a joint Gaussian distribution [34]. A GP is restricted by a mean $\mu(x)$ and a covariance function $k(x,x')$, while $\mu(x)$ is assumed to be zero in most situations, and $k(x,x')$ determines the smoothness of $G(x)$. $k(x,x')$ is regarded as the kernel of GP and needs to be symmetric, continuous, and positive, and the square exponential function in (34) is employed as the kernel in most cases.

$$k(x,x')=l\cdot\exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) \tag{34}$$

where l and σ are the positive parameters.

As for $\mathcal{S}$, it determines the next point to query by selecting the most promising candidate. Normally, three acquisition functions are widely used, which are the maximum probability of improvement (MPI) [35], expected improvement (EI) [36], and upper confidence bound (UCB) [37]. The disadvantage of MPI is that it only chooses the points with highly confident to query, hence there is little improvement of the model. EI overcomes the limitation of MPI by maximizing the expected improvement of the best value at the current stage. In such way, if the new value performances much better, the model improves a lot; if the new value performs much worse, the model maintains the same as before. In this work, EI is chosen as $\mathcal{X}$. The formula of EI is expressed as:

$$\begin{aligned}ACQU_{EI}(x;\{x_n,y_n\},\theta)=&\\ \sigma(x;\{x_n,y_n\},\theta)\gamma(x)\Phi(\gamma(x))+N(\gamma(x);0,1)&\end{aligned} \tag{35}$$

$$\gamma(x)=\frac{f(x_{\text{best}})-\mu(x;\{x_n,y_n\},\theta)}{\sigma(x;\{x_n,y_n\},\theta)} \tag{36}$$

where x_{best} is the best value at the current stage; θ denotes the parameters of GP model; x_n and y_n are the available samples; $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal; $\mu\{x;\{x_n,y_n\},\theta\}$ denotes the predictive mean function; and $\sigma(x;\{x_n,y_n\},\theta)$ denotes the predictive variance function.

The detailed BHO with GP is illustrated in Algorithm 1.

Algorithm 1: BHO with GP

1: **for** $n=1,2,...,$ **do**
2: Find the new x_{n+1} by maximizing acquisition function
3: $x_{n+1}=\arg\max_X \mathcal{S}(x|\mathcal{D}_n)$
4: Query objective function to obtain $y_{n+1}=G(x_n)+\varepsilon_n$
5: Argument data $D_{n+1}=\{D_n,(x_n,y_n)\}$
6: Update GP model
7: **end for**

III. Experimental Setup

A. Data Description and Pre-preparation

Historical demand data collected by the National Electricity Transmission System (NETS), UK, are adopted as the dataset [38]. The dataset contains hourly historical demand data, wind generation, solar generation, and inter-connector flow in the entire British area between 2005 and 2020. In this paper, the National Demand dataset (ND-dataset), which is the sum of generation based on national grid operational generation metering, and the England and Wales Demand dataset (EWD-dataset), which is the demand consumption in England and Wales in UK, are selected as the prediction targets. The load demands of the ND-dataset and the EWD-dataset between 2015 and 2016 are used for the simulation. The dataset is split into training and testing sets (90% for training and 10% for testing). The original dataset is then denoised via DWT as described in Section II-A. Then, the denoised dataset is normalized via max-min normalization, so all data are limited in the range of [−1, 1].

B. Open Access Software Platform and Package

To implement the proposed simulation case study, a variety of open access packages and libraries based on Python 3.7 and TensorFlow 2 are adopted. PyWavelets [39], PyEMD [40], EWTPY [41], and VMDPY [41] are used for implementing DWT, EMD, EWT, and VMD. A BHO package, i.e., Hyperopt [42], is used for hyperparameter tuning.

C. Benchmarks

To better evaluate the prediction performance of the proposed method, other state-of-the-art STLF methods, especially hybrid methods, are taken into considerations. Firstly, two single STLF methods are considered, i. e., 1D CNN-LSTM and 1D CNN-gated recurrent unit (GRU), which are developed in [8] and [43], respectively. Both the 1D CNN layer and LSTM/GRU layer are efficient in extracting time-series features. The two hybrid STLF methods that combine mode decomposition techniques with RNN are selected as benchmark models: the VMD-LSTM method proposed in [17], [44], [45] and the EMD-LSTM method proposed in [16], [23].

D. Performance Metrics

To assess the performance of the proposed predictor, the following four performance metrics are adopted, i. e., mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE), and R^2. The detailed formulas are expressed as:

$$MAE=\frac{\sum_{m=1}^{M}\left|y_m-\hat{y}_m\right|}{M} \tag{37}$$

$$MAPE=\frac{\sum_{m=1}^{M}\left|(y_m-\hat{y}_m)/y_m\right|}{M}\times 100\% \tag{38}$$

$$RMSE=\sqrt{\frac{\sum_{m=1}^{M}(y_m-\hat{y}_m)^2}{M}} \tag{38}$$

$$R^2=1-\frac{SS_{RES}}{SS_{TOT}}=1-\frac{\sum_{m=1}^{M}(y_m-\hat{y}_m)^2}{\sum_{m=1}^{M}(y_m-\dot{y})^2} \tag{39}$$

where $\hat{y}_m$ is the prediction value; and $\dot{y}$ is the mean value.

IV. Results and Discussion

To evaluate the proposed load forecasting method, the aforementioned ND-dataset and EWD-dataset are tested. Three case studies are designed in this section, i.e., the influence of number of sub-layers N, the influence of look-back steps T_L and forecasting steps T_F, and a comparison between the proposed model and relevant works.

A. Influence of Number of Sub-layers

Referring to the EWT decomposition technique introduced in Section II, the original time-varying load demand is decomposed into N sub-layers by the EWT, which are defined as S_1 - S_N in this paper. The number of N has a significant impact on the final forecasting performance. The range of N increases from 5 to 13. Both the ND-dataset and the EWD-dataset are used in the comparison experiment. The performance of the proposed method with different numbers of N is summarized in Tables I and II and Fig. 4.

TABLE I
Prediction Performance of Proposed Method Using ND-dataset with Different N

N	*MAE* (TW)	*MAPE* (%)	*RMSE* (TW)	R^2
5	0.583	2.011	0.712	0.983
6	0.393	1.367	0.556	0.990
7	0.372	1.251	0.532	0.991
8	0.508	1.648	0.614	0.987
9	0.244	0.820	0.342	0.996
10	0.484	1.670	0.578	0.988
11	0.277	0.990	0.380	0.996
12	0.794	2.588	0.873	0.975
13	0.459	1.581	0.555	0.990

TABLE II
Prediction Performance of Proposed Method Using EWD-dataset with Different N

N	*MAE* (TW)	*MAPE* (%)	*RMSE* (TW)	R^2
5	0.369	1.080	0.473	0.979
6	0.401	1.213	0.510	0.978
7	0.239	0.722	0.314	0.992
8	0.230	0.680	0.296	0.992
9	0.324	0.958	0.385	0.987
10	0.301	0.895	0.371	0.988
11	0.279	1.044	0.418	0.993
12	0.190	0.558	0.257	0.994
13	0.229	0.681	0.297	0.992

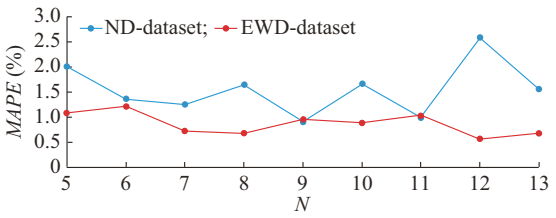

Fig. 4. MAPEs of proposed method using ND-dataset amd EWD-dataset with different N.

From Tables I and II, it is observed that the MAE, MAPE, and RMSE are relatively large when N is too tiny (near 5) or too large (near 13), as shown in Fig. 4. For the ND-dataset, among all N values, the dominant value is $N=9$, followed by $N=11$, where the RMSEs are 0.342 TW and 0.380 TW, respectively. For the EWD-dataset, the proposed method achieves the best performance with the smallest values of MAE, MAPE, and RMSE when $N=12$.

Once the optimal number of decomposition layers is determined, the denoised load demand data are decomposed by EWT to obtain the sub-components. Then, N LSTM predictors are trained simultaneously to predict each sub-component. The predictions for decomposed sublayers given the validation set are shown in Fig. 5. The load demand is decomposed into N sub-layers by the EWT, which provides the best performance of the selected datasets ($N=9$ for the ND-dataset and $N=12$ for the EWD-dataset). For high-frequency component sub-layers (S_1 - S_4), the proposed method has good prediction performance. While the high-frequency components have high fluctuations, most prediction errors come from the prediction for these components.

B. Impact of Look-back Steps and Forecasting Steps

In this case study, the multi-step prediction performance of the proposed method is investigated. Two parameters of the LSTM method, i.e., the look-back steps T_L and forecasting steps T_F, are roughly tuned to look for the parameters to achieve the best performance. In the previous study, only 1-step ahead (half-hour) is predicted, while T_F means the model will make predictions several steps ahead (the interval is half-hour for each step), as illustrated in Fig. 6. The look-back step T_L defines how many previous time steps are used to predict the future load demand.

We vary T_L and T_F from 3 to 9 using an interval of 1 and apply the load forecasting method using every combination of T_L and T_F. The heatmap shown in Fig. 7 indicates the prediction performance, i.e., MAE, using ND-dataset and EWD-dataset with different T_L and T_F combinations. From Fig. 7, it can be observed that the prediction accuracy raises with the increase of look-back step T_L until reaching a threshold. Moreover, the proposed method has a better prediction ability with a smaller value of forecasting step T_F. When T_F increases, the prediction error climbs steadily.

From the figure, the proposed method reaches the best performance when it makes one-step prediction with looking seven steps back, as the MAE are only 0.226 TW and 0.183 TW for ND-dataset and EWD-dataset, respectively. The prediction method performs the worst when the proposed method only utilizes the previous data from three steps before. It is shown that the MAE reaches 0.594 TW when $T_L=3$ and $T_F=9$ for both ND-dataset and EWD-dataset, which confirms that lacking previous information will prevent the predictor from achieving high prediction accuracy. It is also observed that the estimation error increases again when the look-back step T_L is larger than 7, and the main reason for this result is summarized as follows. The LSTM method will first compress the input sequence into a fixed-length vector, which is used for predicting the future load. However, a long input sequence makes the represent vector a bad sum-

mary to the input sequence, which reduces the model performance [46]. An attention-based LSTM will break the limitation and enhance the load forecasting further. An example of the multi-step prediction performance is shown in Fig. 8.

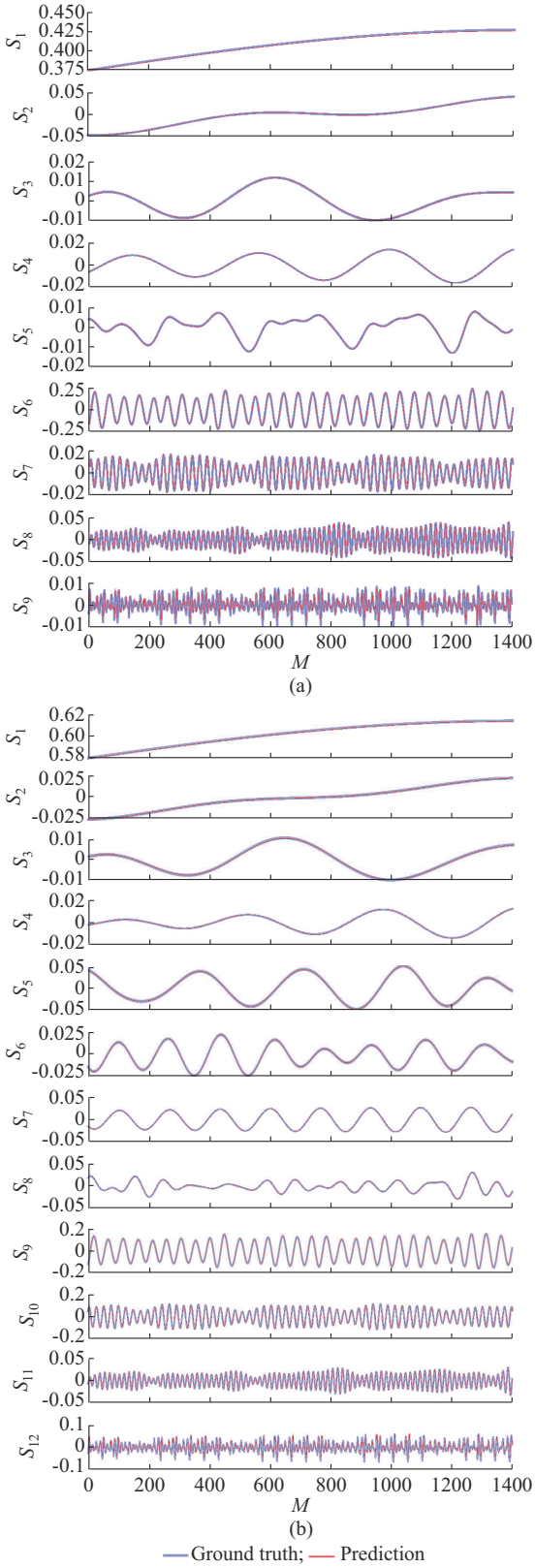

Fig. 5. Prediction for each sub-layer given validation set. (a) ND-dataset with $N=9$. (b) EWD-dataset with $N=12$.

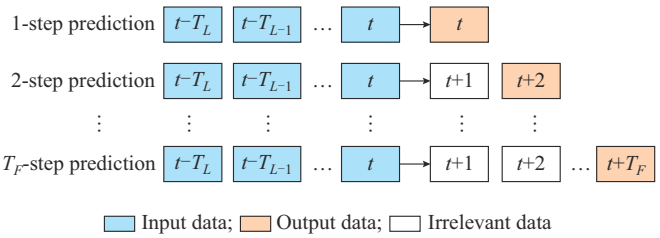

Fig. 6. Structure of multi-step load forecasting.

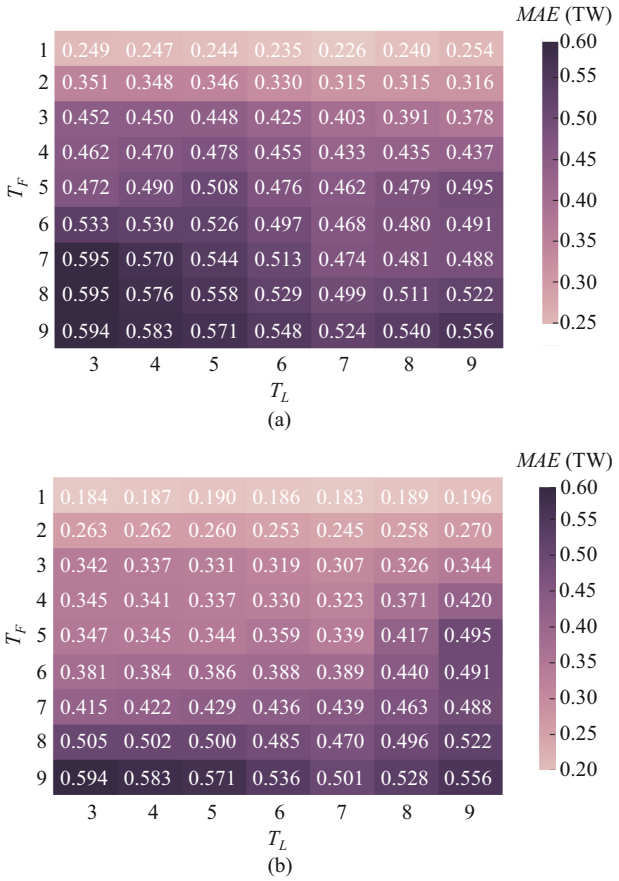

Fig. 7. Prediction performance with different T_L and T_F combinations. (a) ND-dataset. (b) EWD-dataset.

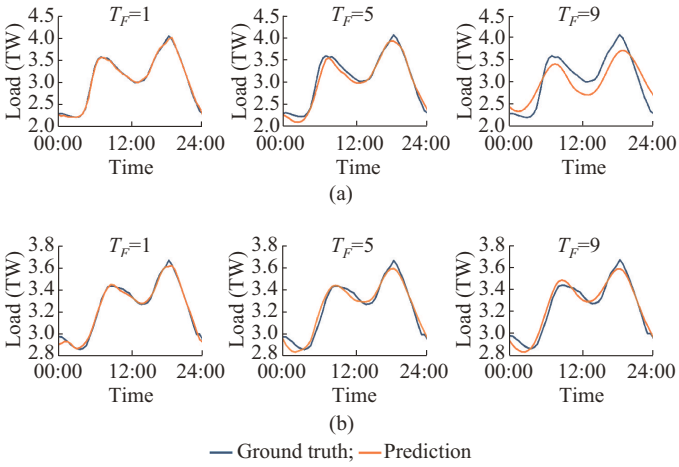

Fig. 8. Examples of multi-step prediction with $T_L=7$. (a) ND-dataset. (b) EWD-dataset.

C. Comparison of BHO with Grid Searching and Random Search

In this case study, the proposed BHO algorithm is compared with two hyperparameter tuning methods, i. e., grid search and random search. As the naive hyperparameter tuning apmethod, grid search simply searches the whole hyperparameters space, which is defined in Table III. Another tuning method, i.e., random search [47], tunes the hyperparameters by randomly selecting the combinations of possible parameters. As the proposed hybrid STLF method trains N BLSTM sub-models in parallel, the optimal hyperparameters should be evaluated for all sub-models. Moreover, the number of sub-layers and look-back step for each dataset are selected as the optimal values. In this paper, hyperparameters include learning rate, dropout rate, cell type, number of hidden layers, epoches, etc. The hyperparameter tuning ranges are shown in Table III.

TABLE III
HYPERPARAMETER TUNING RANGES

Hyperparameter	Range
Learning rate	10^{-5}-10^{-1}
Dropout rate	0.3-0.7
Cell type	GRU, LSTM
Number of hidden layers	1-5
Batch size	32, 64, 128, 256, 512, 1024
Optimizer	Adam, Nadam, RMSprop, Adagrad
Loss	MSE, MAPE, MAE, Huber
Activation function	ReLU, Sigmoid, Tanh
Epoches	20, 50, 100, 150, 200

Both the prediction accuracy and training time are compared in Tables IV and V. From the tables, it is found that although the traditional grid search achieves almost equal prediction accuracy as BHO, it is time-consuming, and it takes almost six times as long as other methods. The disadvantage of the grid search would be more obvious when the hyperparameter space is large or the structure of the neural network is complex. As for the random search method, it costs the shortest time, but the prediction accuracy decreases. As for the proposed BHO method, it takes the advantages of both grid search and random search, as it gives second-optimal results with much faster computation speed than the grid search.

TABLE IV
RESULTS OF DIFFERENT HYPERPARAMETER OPTIMIZATION METHODS USING ND-DATASET

Method	*MAE* (TW)	*MAPE* (%)	*RMSE* (TW)	R^2	Time (hour)
Grid search	0.241	0.803	0.322	0.996	63.32
Random search	0.325	0.931	0.485	0.992	12.24
BHO	0.244	0.820	0.342	0.996	13.12

D. Comparison of Proposed Method with Others

In this case study, the 1-step prediction performance of the proposed method is compared with other relevant forecasting methods. A detailed description of the methods adopted in this paper is listed below: ① 1D CNN-LSTM STLF method; ② 1D CNN-GRU STLF method; ③ EMD-LSTM STLF method; ④ VMD-LSTM STLF method; and ⑤ ND-EWT-BLSTM-BHO STLF method (proposed method). For the former two methods, the original time-varying load demand is adopted as the inputs of neural network models. While for the last three models, the original load demand data are decomposed via EMD, VMD, and EWT, respectively, and then the neural network is trained for each sub-layer.

TABLE V
RESULTS OF DIFFERENT HYPERPARAMETER OPTIMIZATION METHODS USING EWD-DATASET

Method	*MAE* (TW)	*MAPE* (%)	*RMSE* (TW)	R^2	Time (hour)
Grid search	0.195	0.561	0.262	0.994	84.23
Random search	0.312	0.901	0.343	0.989	15.12
BHO	0.190	0.558	0.257	0.994	17.36

Tables VI and VII show the prediction performance of five methods considering the performance metrics, i. e., MAE, MAPE, RMSE, and R^2, of the predicted load demand given the ND-dataset and the EWD-dataset, respectively.

TABLE VI
PREDICTION PERFORMANCE OF FIVE METHODS USING ND-DATASET

Method	T_L	*MAE* (TW)	*MAPE* (%)	*RMSE* (TW)	R^2
1D CNN-LSTM	1	0.719	2.732	1.035	0.966
	3	0.369	1.253	0.506	0.992
	5	0.347	1.065	0.475	0.993
	7	0.482	1.587	0.617	0.989
1D CNN-GRU	1	1.131	3.561	1.407	0.942
	3	0.362	1.325	0.501	0.992
	5	0.345	1.057	0.475	0.993
	7	0.343	1.052	0.484	0.993
VMD-LSTM	1	0.814	2.614	1.097	0.960
	3	0.309	1.061	0.388	0.994
	5	0.295	0.925	0.369	0.995
	7	0.243	0.807	0.305	0.996
EMD-LSTM	1	0.871	3.04	1.005	0.965
	3	0.421	1.384	0.535	0.990
	5	0.489	1.604	0.645	0.986
	7	0.526	1.774	0.689	0.985
Proposed method	1	0.507	1.485	0.651	0.961
	3	0.249	0.840	0.357	0.996
	5	0.244	0.820	0.342	0.996
	7	0.226	0.757	0.325	0.997

As shown in the tables, the proposed method outperforms other methods. Moreover, the spectral load forecasting methods including ND-EWT-LSTM-BHO, EMD-LSTM, and VMD-LSTM have better prediction accuracy than conventional deep learning methods including 1D CNN-LSTM and

1D CNN-GRU. 1D CNN-LSTM and 1D CNN-GRU methods have the worst estimation performance with the highest MAE, MAPE, and RMSE almost in all experiments. The prediction performances of VMD-LSTM and EMD-LSTM are quite similar, which is just below the proposed method. Figure 9 compares the prediction values with the testing set using the proposed and benchmark methods. The results predicted by the proposed method are the closest to the ground truth measurements with both the ND-dataset and the EWD-dataset. Moreover, the results estimated by the CNN-LSTM and CNN-GRU methods are farthest from the ground truth curve, showing that CNN-LSTM and CNN-GRU perform worst among all models.

TABLE VII
PREDICTION PERFORMANCE OF FIVE METHODS GIVEN EWD-DATASET

Method	T_L	*MAE* (TW)	*MAPE* (%)	*RMSE* (TW)	R^2
1D CNN-LSTM	1	0.866	2.857	1.170	0.945
	3	0.474	1.378	0.698	0.980
	5	0.359	1.062	0.606	0.986
	7	0.344	0.998	0.484	0.992
1D CNN-GRU	1	1.018	3.125	1.308	0.929
	3	0.552	1.562	0.809	0.973
	5	0.363	1.078	0.657	0.983
	7	0.377	1.113	0.656	0.983
VMD-LSTM	1	0.507	1.485	0.651	0.961
	3	0.315	0.918	0.396	0.985
	5	0.212	0.630	0.267	0.993
	7	0.245	0.712	0.301	0.992
EMD-LSTM	1	0.527	1.543	0.703	0.956
	3	0.331	0.961	0.400	0.986
	5	0.468	1.365	0.598	0.965
	7	0.605	1.827	0.715	0.956
Proposed method	1	0.570	1.655	0.748	0.950
	3	0.184	0.547	0.254	0.995
	5	0.190	0.558	0.257	0.994
	7	0.183	0.543	0.247	0.995

Figure 10 shows the high-density scatter plot of the ground truth and prediction values with different forecasting methods. The scatter plot shows the correlation relationship between the two variables. The higher the R^2 value, the stronger the correlation between the ground truth and prediction values, representing higher accuracy achieved by the forecasting method. For the proposed method, the scatter about the line is relatively small, and most points are on the regression line, with only several data values far from other data values. For other spectral methods, the R^2 of VMD-LSTM and EMD-LSTM methods also show a strong correlation with the ground truth curve, with R^2 values over 0.99. CNN-GRU shows the worst correlation from the scatter plot, with R^2 values of 0.992 and 0.973 for the two datasets, respectively.

E. Discussion

In this subsection, three case studies are presented. The main findings are summarized as follows.

1) The first case study aims to optimize the sub-layer number N to achieve the best performance. Referring to the simulation, it is observed that when N ranges between 9 and 12, the proposed method achieve the smallest RMSE, MAPE, and MAE values.

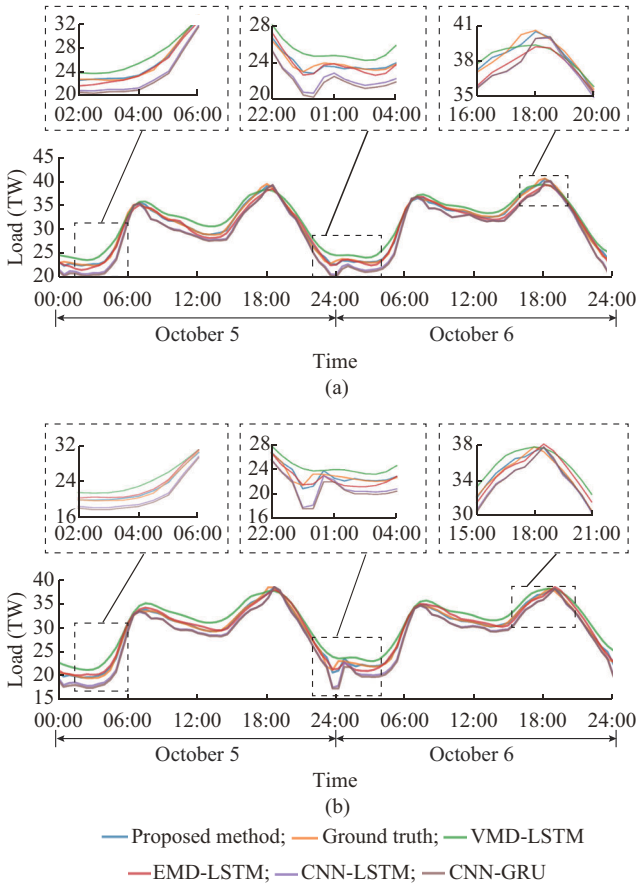

Fig. 9. Load forecasting performance of proposed and benchmark methods. (a) ND-dataset. (b) EWD-dataset.

2) For the second case study, the impacts of look-back steps T_L and forecasting steps T_F are studied. From the results presented in Fig. 8, it can be concluded that the proposed method can forecast demand load accurately even nine hours later in the future. It is observed that increasing the value of T_L improves the performance, while the prediction accuracy will not increase further once T_L reaches a certain limitation. The explanation for the result is that the demand load in previous time steps has a very strong correlation with the demand load in the future. However, if T_L is too large, the LSTM method would lose concentration and be unable to match the prediction with the previous values.

3) In the last case study, the optimized method is compared with other forecasting methods, i.e., 1D CNN-LSTM, 1D CNN-GRU, VMD-LSTM, and EMD-LSTM. From the results, it is shown that the proposed method improves RMSE by 28.01% and 48.97%, MAE by 34.11% and 46.80%, and MAPE by 28.92% and 45.59% for the ND-dataset and the EWD-dataset, respectively.

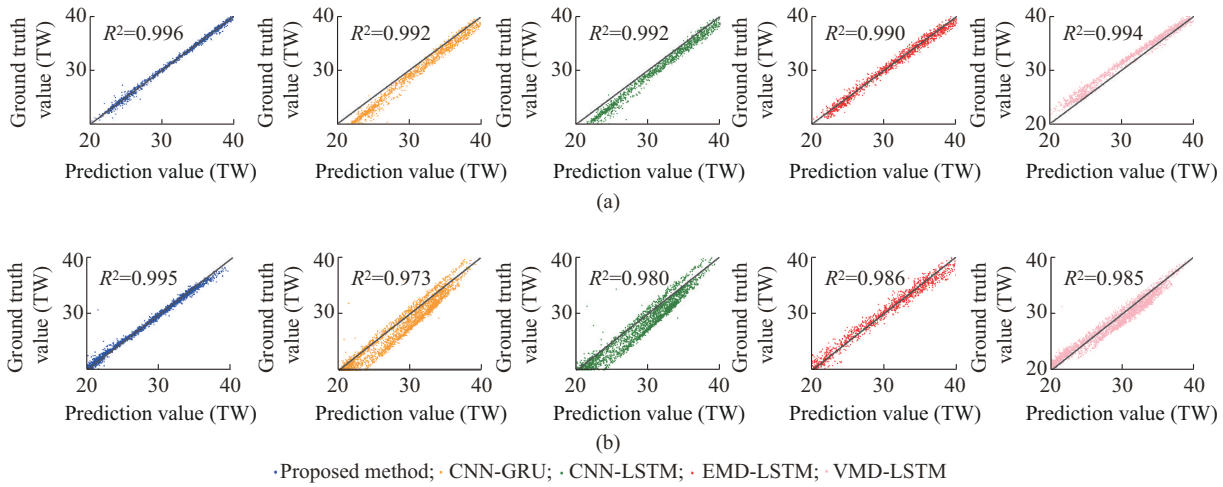

Fig. 10. High-density scatter plot of ground truth and prediction values of different load forecasting methods. (a) ND-dataset. (b) EWD-dataset.

V. Conclusion

Accurate load forecasting is extremely important for demand-side management and power planning. In this paper, a hybrid load forecasting method ND-EWT-BLSTM-BHO is proposed by extracting both time-domain and frequency-domain information to reduce the uncertainty of load forecasting. The method considers the wavelet-based denoising algorithm, EWT component decomposition technique, BLSTM algorithm, and BHO algorithm. The proposed method first filters noise such as electric spikes from the measured load demand data. Then, an EWT algorithm is adopted to decompose the data into N sub-layers to extract time-domain and frequency-domain features. N LSTM neural network models are trained for all sub-layers at the next step.

Additionally, a BHO algorithm tunes the hyperparameters to find the best combinations that achieve the best performance. Finally, the prediction results for all sub-layers are reconstructed and used to present the result of the load forecasting. The British load demand dataset is used for the simulation. The demands of ND-dataset and EWD-dataset are thoroughly investigated. In this paper, three case studies are demonstrated. The conclusion is that the proposed method has better performance than the existing component decomposition models.

The limitation of this paper is that the proposed method requires a high-computation server to process high-volume data, and such a server may be unavailable. In the future, a more computation-efficient method should be proposed to reduce the computation cost.

References

[1] D. Gan, Y. Wang, S. Yang *et al.*, "Embedding based quantile regression neural network for probabilistic load forecasting," *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 2, pp. 244-254, Mar. 2018.

[2] J. Wang, X. Chen, F. Zhang *et al.*, "Building load forecasting using deep neural network with efficient feature fusion," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 1, pp. 160-169, Jan. 2021.

[3] C. Si, S. Xu, C. Wan *et al.*, "Electric load clustering in smart grid: Methodologies, applications, and future trends," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 2, pp. 237-252, Mar. 2021.

[4] G. Dudek, "Pattern-based local linear regression models for short-term load forecasting," *Electric Power Systems Research*, vol. 130, pp. 139-147, Jan. 2016.

[5] B. Dhaval and A. Deshpande, "Short-term load forecasting with using multiple linear regression," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 4, pp. 3911-3917, Aug. 2020.

[6] M. Massaoudi, S. S. Refaat, I. Chihi *et al.*, "A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for short-term load forecasting," *Energy*, vol. 214, p. 118874, Jan. 2021.

[7] C.-M. Lee and C.-N. Ko, "Short-term load forecasting using lifting scheme and ARIMA models," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5902-5911, May 2011.

[8] W. Kong, Z. Y. Dong, Y. Jia *et al.*, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841-851, Jan. 2019.

[9] S. Li, L. Goel, and P. Wang, "An ensemble approach for short-term load forecasting by extreme learning machine," *Applied Energy*, vol. 170, pp. 22-29, May 2016.

[10] Y. Wang, D. Gan, N. Zhang *et al.*, "Feature selection for probabilistic load forecasting via sparse penalized quantile regression," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 5, pp. 1200-1209, Sept. 2019.

[11] H. Jahangir, H. Tayarani, S. S. Gougheri *et al.*, "Deep learning-based forecasting approach in smart grids with microclustering and bidirectional LSTM network," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 8298-8309, Jul. 2020.

[12] H. Jahangir, S. S. Gougheri, B. Vatandoust *et al.*, "Plug-in electric vehicle behavior modeling in energy market: a novel deep learning-based approach with clustering technique," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 4738-4748, Nov. 2020.

[13] M. Sun, Y. Wang, F. Teng *et al.*, "Clustering-based residential baseline estimation: a probabilistic perspective," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6014-6028, Nov. 2019.

[14] F. L. Quilumba, W.-J. Lee, H. Huang *et al.*, "Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 911-918, Mar. 2015.

[15] Y. Li, D. Han, and Z. Yan, "Long-term system load forecasting based on data-driven linear clustering method," *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 2, pp. 306-316, Mar. 2018.

[16] Neeraj, J. Mathew, R. K. Behera *et al.*, "EMD-Att-LSTM: a data-driven strategy combined with deep learning for short-term load forecasting," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 5, pp. 1229-1240, Sept. 2022.

[17] S. H. Kim, G. Lee, G.-Y. Kwon *et al.*, "Deep learning based on multi-decomposition for short-term load forecasting," *Energies*, vol. 11, no. 12, pp. 1-17, Dec. 2018.

[18] X. Shi, X. Lei, Q. Huang *et al.*, "Hourly day-ahead wind power prediction using the hybrid model of variational model decomposition and long short-term memory," *Energies*, vol. 11, no. 11, pp. 1-20, Nov. 2018.

[19] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and*

Practice. Melbourne: OTexts, 2018.
[20] H. Liu and Z. Long, "An improved deep learning model for predicting stock market price time series," *Digital Signal Processing*, vol. 102, p. 102741, Jul. 2020.
[21] N. E. Huang, "Introduction to the Hilbert-Huang transform and its related mathematical problems," in *Hilbert-Huang Transform and Its Applications*. Singapore: World Scientific, 2014.
[22] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531-544, Nov. 2013.
[23] Y. K. Semero, J. Zhang, and D. Zheng, "EMD-PSO-ANFIS-based hybrid approach for short-term load forecasting in microgrids," *IET Generation, Transmission & Distribution*, vol. 14, no. 3, pp. 470-475, Feb. 2020.
[24] P. Singh, G. Pradhan, and S. Shahnawazuddin, "Denoising of ECG signal by non-local estimation of approximation coefficients in DWT," *Biocybernetics and Biomedical Engineering*, vol. 37, no. 3, pp. 599-610, Jun. 2017.
[25] J. Gilles, "Empirical wavelet transform," *IEEE Transactions on Signal Processing*, vol. 61, no. 16, pp. 3999-4010, Aug. 2013.
[26] K. Thirumala, A. C. Umarikar, and T. Jain, "Estimation of single-phase and three-phase power-quality indices using empirical wavelet transform," *IEEE Transactions on Power Delivery*, vol. 30, no. 1, pp. 445-454, Feb. 2015.
[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.
[28] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, Nov. 1997.
[29] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602-610, Aug. 2005.
[30] M. E. Hodgson, "Searching methods for rapid grid interpolation," *The Professional Geographer*, vol. 41, no. 1, pp. 51-61, Mar. 1989.
[31] S. Koziel and X.-S. Yang. *Computational Optimization, Methods and Algorithms*. New York: Springer, 2011.
[32] A. H. Victoria and G. Maragatham, "Automatic tuning of hyperparameters using Bayesian optimization," *Evolving Systems*, vol. 12, pp. 217-223, May 2021.
[33] M. Zhao and J. Li, "Tuning the hyper-parameters of CMA-ES with tree-structured Parzen estimators," in *Proceedings of 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, pp. Xiamen, China, Jun. 2018, pp. 613-618.
[34] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. New York: Springer, 2003.
[35] I. Couckuyt, D. Deschrijver, and T. Dhaene, "Fast calculation of multi-objective probability of improvement and expected improvement criteria for Pareto optimization," *Journal of Global Optimization*, vol. 60, no. 3, pp. 575-594, Oct. 2014.
[36] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," *Advances in Neural Information Processing Systems*, vol. 25, pp. 2951-2959. Dec. 2012.
[37] N. Radović and M. Erceg, "Hardware implementation of the upper confidence-bound algorithm for reinforcement learning," *Computers & Electrical Engineering*, vol. 96, p. 107537, Dec. 2021.
[38] National Grid ESO. (2020, Dec.). Historic demand data. [Online]. Available: https://www.nationalgrideso.com/data-explorer
[39] G. Lee, R. Gommers, F. Waselewski *et al.*, "Pywavelets: a Python package for wavelet analysis," *Journal of Open Source Software*, vol. 4, no. 36, pp. 1-2, Apr. 2019.
[40] Laszuk D. (2017, May). Python implementation of empirical mode decomposition algorithm. [Online]. Available: https://github. com/laszukdawid/PyEMD
[41] V. R. Carvalho, M. F. Moraes, A. P. Braga *et al.*, "Evaluating five different adaptive decomposition methods for EEG signal seizure detection and classification," *Biomedical Signal Processing and Control*, vol. 62, p. 102073, Sept. 2020.
[42] J. Bergstra, D. Yamins, D. D. Cox *et al.*, "Hyperopt: a Python library for optimizing the hyperparameters of machine learning algorithms," in *Proceedings of the 12th Python in Science Conference*, Austin, USA, Jan. 2013, pp. 1-8.
[43] M. Sajjad, Z. A. Khan, A. Ullah *et al.*, "A novel CNN-GRU-based hybrid approach for short-term residential load forecasting," *IEEE Access*, vol. 8, pp. 143759-143768, Jul. 2020.
[44] G. Zhang, H. Liu, P. Li *et al.*, "Load prediction based on hybrid model of VMD-MRMR-BPNN-LSSVM," *Complexity*, vol. 2020, pp. 1-21, Jan. 2020.
[45] F. He, J. Zhou, Z. Feng *et al.*, "A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm," *Applied energy*, vol. 237, pp. 103-116, Mar. 2019.
[46] Y. Wang, M. Huang, X. Zhu *et al.*, "Attention-based LSTM for aspect-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, USA, Nov. 2016, pp. 606-615.
[47] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 2, pp. 1-2, Dec. 2012.

Xiaoyu Zhang received the B. Eng. degree from the North China Electric Power University, Beijing, China, in 2016, the M.S. degree with the distinction in power system from University of Birmingham, Birmingham, UK, in 2017, and the Ph.D. degree in electrical engineering from the Royal Holloway, University of London, London, UK, in 2022. His research interests include deep learning technology and data analytics in smart grids, smart grid privacy and security, and demand-side management.

Stefanie Kuenzel received the M. Eng. and Ph. D. degrees from Imperial College London, London, UK, in 2010 and 2014, respectively. She is currently the Head of the Power Systems Group and a Senior Lecturer with the Department of Electronic Engineering, Royal Holloway, University of London, London, UK. Her current research interests include renewable generation and transmission, including high-voltage direct current (HVDC) as well as Smart Meters.

Nicolo Colombo received the Ph. D degree in theoretical physics from the University of Mons, Mons, Belgium, in 2012. After a brief research appointment in the physics department of Texas A&M University, he made a career switch to computer science and worked as a Research Associate in the machine learning group of the Luxembourg Centre for Systems Biomedicine, Luxembourg, and the Statistical Science department of University College Londodn, London, UK. Since 2019, He is a Lecturer at the Royal Holloway University of London, London, UK. His current research interests include theoretical machine learning, optimization, neural networks, conformal prediction, and applications to biomedicine and transportation systems.

Chris Watkins received the Ph. D. degree from University of Cambridge, Cambridge, UK, in 1989. He is a world-class authority on reinforcement learning & evolutionary theory and Professor of Artificial Intelligence at Royal Holloway, University of London, London, UK. He coined the Q-Table algorithm approach that spurred the resurgence in reinforcement learning (this approach was at the heart of Google's recent successful AI projects). Prior to returning to academia, he was employed as a Quant at a hedge fund firm in London for several years. His current research interests include intelligent machines, bstract models of evolution, and statistical visualization.