Energy 203 (2020) 117858

Contents lists available at ScienceDirect

Energy

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

A novel ensemble method for hourly residential electricity consumption forecasting by imaging time series

Guoqiang Zhang*, Jifeng Guo

Northeast Forestry University Information and Computer Engineering Institute, Heilongjiang, Harbin, 150000, China

ARTICLE INFO

Article history:
Received 9 January 2020
Received in revised form
10 May 2020
Accepted 11 May 2020
Available online 13 May 2020

Keywords:
Electricity consumption forecasting
Conditional generative adversarial networks (CGANs)
Feature transform
Gramian angular fields (GAFs)
Improved dragonfly algorithm (IDA)

ABSTRACT

In this paper, a novel ensemble method is proposed to forecast the hourly consumption of residential electricity. Firstly, variational mode decomposition (VMD) is applied to decompose weather conditions (relative humidity and temperature, etc.), residential building data (manually operated appliances relevant to residents' lifestyle, dishwasher, heating heat-pump, and television, etc.), and electricity price into several band-limited intrinsic mode functions (BLIMFs). Then the incremental kernel principal component analysis (IKPCA) is applied to extract the incremental kernel principal components (IKPCs) from the BLIMFs. Next, IKPCs are encoded as images by the Gramian Angular Fields (GAFs). Secondly, a novel ensemble method based on conditional generative adversarial networks (CGANs), is applied to simulate the variability in people's electrical behavior and weather forecast errors. Moreover, the elitist search strategy of the multi-population genetic algorithm (MPGA) is introduced to realize the communication among each sub-CGAN. And then all sub-CGANs are integrated by the Huffman coding (HC). Thirdly, an improved dragonfly algorithm (IDA) is developed to optimize the weights of HC. The experimental results show that the forecasting results of the proposed ensemble method are obviously better than those of other standard and state-of-the-art methods tested in this paper.

1. Introduction

Load forecasting plays a key role in security and generation scheduling of electric power industry. Load forecasting with high accuracy can not only improve energy transaction efficiency but reduce operating cost in electric power industry. It is suggested that the accuracy changes in load forecasting plays a key role for operating costs in electric power industry companies [1]. However, residential electricity load demand is affected by lots of uncertain factors, such as weather conditions, residential building data, and electricity price. These uncertain factors bring about increasing challenges to load forecasting [2–7].

So far, kinds of techniques have been employed for load forecasting. In the early stages, regression methods [8,9] and time series methods [10–12] were extensively applied to forecast the load. Overall, the main purpose of the regression methods is to understand the relationship between multiple input variables (electricity price, temperature, etc.) and the target variable (hourly electricity consumption) via a goodness-of-fit function. Obviously, regression models can provide a satisfactory performance only when the relationship between input variables and the target variable are well formulated with the help of some expert knowledge. However, it is difficult to construct an appropriate expert system. In other words, due to the insufficient non-linear fitting capability in regression methods, these methods are actually prevented from achieving a satisfactory forecasting accuracy [13]. The time series methods are usually aimed to resolve serially correlated errors and non-linearity. Nonetheless, a series of hyper-parameter tuning (lag sets, threshold sets, etc.) are required in the time series methods, which is critical to the overall forecasting performance of the model. Moreover, in order to obtain a proper fitting relationship between the input and target variables, a larger parameter space is needed, which will lead to longer calculation time and thus cause over-fitting [14].

In order to efficiently address the various non-linearity in electricity consumption data, various artificial intelligence (AI) technologies have been extensively applied to load forecasting, such as classical artificial neural networks (ANNs) [15–17], fuzzy theory method [18–20], support vector regression (SVR) [21,22]. The advantage of the classical ANN-based model is that it is able to

* Corresponding author.
E-mail addresses: limiyi061@qq.com, limiyi061@ieee.org (G. Zhang).

https://doi.org/10.1016/j.energy.2020.117858
0360-5442/

model the complicated nonlinear relationship between input variables (electricity price, temperature, etc.) and the target variable (hourly electricity consumption), which saves the need for quantitative correlation between input and target variables or complex mathematical formulations. Unfortunately, classical ANN-based model sometimes can suffer from overfitting and local optimization, possibly giving rise to unsatisfactory forecasting accuracy. In addition, owing to the tedious and adjustment process of trial-and-error, it is often difficult to determine whether the obtained ANN-based model is optimal [23]. Fuzzy theory methods can provide a better way to incorporate the cross-effects between input and target variables, via fuzzy membership function. However, the methods of fuzzy theory are vulnerable to local minima and subjective choice of the structure of the model [24]. Unlike most of the classical ANNs, the principle of structural risk minimization is introduced into SVR to minimize the upper limit of generalization error rather than minimizing training error. The research results of [21,22] show that there are three parameters playing an import role on the forecasting performance of SVR model. Specifically, they include the parameter which is used to weigh training errors and big weights, the parameter which is used to control Gaussian kernel function, and that for controlling the width of the insensitive loss function. It can be observed that better forecast accuracy can be obtained through adjusting these three parameters. However, SVR needs to map the input samples into a high-dimensional space to differentiate the samples. When there are a large number of input samples, the high-dimensional space will be larger, which will result in longer calculation time [25].

To overcome the drawbacks of the AI methods mentioned above, various deep learning methods are applied to electricity consumption forecasting, for instance, long short-term memory (LSTM) [26], and deep belief networks (DBN) [27,28]. In Ref. [26], the electricity consumption of residents was forecasted by means of manual operating appliances (dishwasher, heat pump, washing machine, TV, etc.) related to residents' life style, and satisfactory results were obtained. The success can be attributed to the powerful memory unit of LSTM, which can control the number of important information to remember and unimportant features to forget during training. In Ref. [27,28], multiple restricted Boltzmann machines (RBMs) are adopted to extract the input features, which can effectively reduce the information loss between different variables.

It should be pointed that no individual forecasting model can perform well for all datasets [14,29]. The main phenomenon is that the model performs well on this dataset. In the meantime, it may perform poorly on another dataset. Therefore, more and more hybrid methods [30–33] and ensemble methods [34,35] have been employed for load forecasting. The test results all verify that the forecasting performance of an ensemble and hybrid model is superior to that of a single model. Nevertheless, there is a major drawback in these hybrid and ensemble methods, namely they are trained only with static samples. In fact, there is certain variability in people's electrical behavior [26]. Moreover, it may bring some errors to the weather forecast due to complex environmental changes [36]. Therefore, a dynamic sample mechanism needs to be incorporated to improve the adaptability of the forecasting model. Moreover, the conditional generative adversarial networks (CGANs) [37], consist of generator model (G) and discriminator model (D). The advantage of CGAN is that its generator model can generate new realistic synthesized samples according to the input training samples. It justifies why CGAN is proposed to address the dynamic sample mechanism. G is applied to simulate the variability in people's electrical behavior and weather forecast errors, and to generate realistic synthesized data (RSD) by using the training set, while D is applied to distinguish RSD and real data set. CGAN is a deep learning model based on image processing. Generally speaking, it works best when its generation model and discrimination model are both deep learning models. However, CGAN, which is developed based on image processing [37], has been applied in a variety of fields involving image processing [38–40]. Due to the influence of the network structure, the input features are encoded into images. Moreover, convolutional neural networks (CNNs) [41] have achieved satisfactory results in the field of image processing [42,43]. Therefore, CNNs are selected to serve as the generation model and discrimination model of CGAN to address the novel input feature (imaging time-series) proposed in this paper.

Recently, many feature decomposition techniques have been extensively employed for load forecasting, including wavelet [44,45], empirical mode decomposition (EMD) [46], and variational mode decomposition (VMD) [47], etc. The advantage of feature decomposition is that a series of sub-components (with more stable variances and fewer outliers) can be obtained, which is helpful to forecasting performance [48,49]. VMD is an improved algorithm based on EMD, aiming to obtain better robustness to measurement noise relying on the non-recursive and variational modal decomposition to decompose the original input variables. Thus, it is helpful to reduce redundant information in the original variables. Limited by personal ability, we only compared wavelet, EMD and VMD in ***CASE III***. The test results all verify that the forecasting performance of a model with feature decomposition is superior to that of a model without feature decomposition. In this paper, the feature decomposition techniques mentioned above are compared, and VMD is chosen to decompose the hourly load time series according to the comparison results. However, excessive input variables may decrease the forecasting performance [50], while removing redundant variables can improve the forecasting performance [51]. The influence of excessive input variables and redundant variables are tested in ***CASE II***. Therefore, the band-limited intrinsic mode functions (BLIMFs) decomposed by VMD [52] are extracted by the incremental kernel principal component analysis (IKPCA) [53]. The Gramian Angular Fields (GAFs) [54] can effectively encode time series as images. Moreover, GAFs have achieved satisfactory processing results in many fields [54,55]. Hence, the incremental kernel principal components (IKPCs) extracted by IKPCA are encoded as images by GAFs.

Huffman coding (HC) [56] combining applicability and simplicity excels in computing the minimum of the weighted path length (WPL), with the advantage of assigning different weight coefficients to each sub-model that needs to be integrated, just as some experts are voting for each sub-model [57]. HC has the advantage of starting with nodes (sub-models) with minimal forecasting error and less need for too many nodes (sub-models) compared with the existing ensemble methods tested in ***CASE IV***. The advantage of HC determines that it can enhance forecasting accuracy while cutting the training and forecasting time. Therefore, HC was chosen to integrate each sub-CGAN. Furthermore, to reduce the training time of the integrated CGANs, the elitist search strategy of the multi-population genetic algorithm (MPGA) [58] is introduced to realize the cooperative learning among each sub-CGAN in the ensemble model. In other words, each sub-CGAN serves as a sub-population of MPGA. The training error of CGAN serves as the fitness value. For the generator models (G), in the horizontal direction, every N_{it} iteration, each sub-CGAN learns weights from other sub-CGANs according to the difference degree of the average error. In N_{it} iterations, if the training error of a sub-CGAN does not decrease for N_c consecutive times, the sub-CGAN generates new weights at a certain probability. However, in the vertical direction, the best weights in each sub-CGAN are selected and saved to Elitist Population G. The principles for cooperative learning in the discriminator models (D) are the same as those in the generator

models except that the best weight in each sub-CGAN is saved to Elitist Population D. Moreover, to improve the convergence ability of HC, an improved dragonfly algorithm (IDA) has been developed to optimize the weight coefficients of HC.

The contributions and advantages of this paper are as follows:

1) A novel input feature of encoding time series as images is proposed, that is, the BLIMFs decomposed by VMD are extracted by IKPCA, and then the extracted results (IKPCs) are encoded as images by GAFs. The proposed input feature has two advantages. Firstly, the newly decomposed subcomponents have fewer outliers and more stable variances [45,59]. Secondly, IKPCA extracting can remove excessive and redundant information [51].
2) The generator model (G) of CGAN is applied to simulate the variability in people's electrical behavior and weather forecast errors [60], and to generate realistic synthesized data (RSD) according to training samples; at the same time, the discriminator model (D) is chosen to distinguish the RSD and real data sample. The advantage of CGAN is that its sample simulation strategy not only contributes to the robustness and generalization ability of the forecasting model but solves the issue of insufficient training samples for holiday electricity consumption forecasting.
3) CGAN is integrated by HC. Moreover, an improved dragonfly algorithm (IDA) has been developed to optimize the weight coefficients of HC, thereby strengthening the convergence ability of HC. The advantage of an ensemble model is that it can take full advantage of each sub-model. Furthermore, HC can identify the contributions of all sub-models.
4) The elitist search strategy of MPGA is introduced to realize the cooperative learning among each sub-CGAN in the ensemble model. Each sub-model improves its own convergence ability through cooperative learning, which can improve the overall forecasting performance of the overall model.

2. Principles of the proposed method

2.1. Processing of input features

Variational mode decomposition (VMD) [52] is a new non-stationary signal adaptive decomposition estimator method. Its purpose is to obtain better robustness to measurement noise using non-recursive and variational modal decomposition to decompose the original input variables. Therefore, it is helpful to decrease redundant information in the original time series of residential electricity consumption. Let the real valued input variables of meteorological factors, electricity price, and residential building data be F_i ($i = 1,2, \ldots, n$, i means the serial number of original variables of residential electricity consumption; 1 represents temperature, while 2 indicates the electricity price, etc.), the detailed decomposition steps can be summarized as follows:

Step1: The real valued input variables F_i are decomposed into a discrete number of sub-signals (band-limited intrinsic mode functions, BLIMFs), $\{u_k(t)\}, k = 1, 2, \ldots K$. For each BLIMF, the Hilbert transform is introduced to calculate the unilateral frequency spectrum (UFS) by Eq. (1), and then, UFS is shifted to a "baseband" by Eq. (2). In other words, the principle of implementation is mixing the respective center frequency ω_k and its corresponding exponential term together.

$$UFS = \left[\delta(t) + \frac{j}{\pi t} * u_k(t)\right] \tag{1}$$

$$baseband = \left[\delta(t) + \frac{j}{\pi t} * u_k(t)\right] e^{-j\omega_k t} \tag{2}$$

Step2: The Gaussian smoothness is introduced to estimate the bandwidth, and then the constrained variational problem can be obtained:

$$\begin{cases} \min\limits_{\{u_k\},\{\omega_k\}} \left\{ \sum\limits_k \left\| \partial_t \left[\delta(t) + \frac{j}{\pi t} * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\} \\ s.t. \sum\limits_k u_k = f \end{cases} \tag{3}$$

Here $\{u_k\} = \{u_1, \ldots, u_K\}$, $\{\omega_k\} = \{\omega_1, \ldots, \omega_K\}$,

Step3: To determine the number of decomposition of the original electricity consumption time series, the penalty factor α and Lagrangian multiplier $\lambda(t)$ are introduced to convert Eq. (3) into a non-constrained variational problem:

$$\begin{aligned} L(\{u_k\},\{\omega_k\},\lambda) = \alpha \sum_k \left\| \partial_t \left[\delta(t) + \frac{j}{\pi t} * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \\ + \left\| f(t) - \sum_k u_k(t) \right\|_2^2 + \langle \lambda(t), f(t) - \sum_k u_k(t) \rangle. \end{aligned} \tag{4}$$

Step4: The augmented Lagrangian saddle point is calculated by the alternate direction method of multipliers (ADMM):

$$\begin{aligned} u_k^{n+1} = \underset{u_k, u_k \in X}{\operatorname{argmin}} \Big\{ \alpha \| j\omega[(1 + \operatorname{sgn}(\omega + \omega_k)) u_k(\omega + \omega_k)] \|_2^2 \\ + \left\| f(\omega) - \sum_i u_i(\omega) + \frac{\lambda(\omega)}{2} \right\|_2^2 \Big\} \end{aligned} \tag{5}$$

Replace ω with $\omega - \omega_k$ in Eq. (5), the non-negative frequencies and its solution can be calculated by Eq. (6) and Eq. (7), respectively:

$$\begin{aligned} u_k^{n+1} = \underset{u_k, u_k \in X}{\operatorname{argmin}} \left\{ \int_0^\infty 4\alpha(\omega - \omega_k)^2 |u_k(\omega)|^2 \right. \\ \left. + 2 \left| f(\omega) - \sum_i u_i(\omega) + \frac{\lambda(\omega)}{2} \right|^2 d\omega \right\} \end{aligned} \tag{6}$$

$$u_k^{n+1}(\omega) = \frac{f(\omega) - \sum\limits_{i \neq k} u_i(\omega) + \frac{\lambda(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k)^2} \tag{7}$$

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega |u_k(\omega)|^2 d\omega}{\int_0^\infty |u_k(\omega)|^2 d\omega} \tag{8}$$

$$\lambda^{n+1}(\omega)=\lambda^{n}(\omega)+\tau\left[f(\omega)-\sum_{k}u_k^{n+1}(\omega)\right] \tag{9}$$

According to the same process, the updated center frequency ω_k can be calculated by Eq. (8), repeat Eqs. (6)–(8) until

$$\sum_{k}\left\|u_k^{n+1}-u_k^{n}\right\|_2^2\Big/\left\|u_k^{n}\right\|_2^2<\varepsilon \tag{10}$$

When this process has been finished, the decomposed results (BLIMFs) of the original electricity consumption time series can be obtained. Here $u_k^{n+1}(\omega)$ means the Wiener Filtering of $f(\omega)-\sum\hat{u_i}(\omega)$, ω_k^{n+1} means the corresponding gravity center of the current Mode's power spectrum.

However, after being decomposed by VMD, excessive input variables may be generated [50], which can decrease the forecasting performance [51] (test in ***CASE II***). The incremental kernel principal component analysis (IKPCA) is a dimensionality reduction algorithm, which can not only effectively extract the feature relationships between different electricity consumption time series decomposed by VMD but also diminish the loss of initial information. Therefore, the decomposed results (BLIMFs) of the original electricity consumption time series decomposed by VMD are extracted by IKPCA. Let $M_{a:b,c:d}$ represent a matrix with the rows a^{-} th to b^{-th}, and the columns c^{-th} to d^{-th}, $1_{r,c}$, $0_{n,c}$ and I_n mean a $n\times n$ one matrix, a $n\times c$ zero matrix and a $n\times n$ identity matrix, respectively. The detailed extracted steps are as follows according to Ref. [53].

Step1: The decomposed electricity consumption time series (BLIMFs) matrix $\alpha=[x_1,\ \ldots,\ x_n]\in\mathbb{R}^{m\times n}$ is mapped into $A=[\Phi(x_1),\ \ldots,\Phi(x_n)]$ using a higher dimensional mapping function Φ, $x_i\in\mathbb{R}^m$ represents the i-th input vector;

Step2: Calculate the mean μ_A of A, and center A:

$$\begin{cases}\mu_A=A\left(\frac{1}{n}1_{n,1}\right)=Av,v=\left(\frac{1}{n}1_{n,1}\right)\\ \hat{A}=A(I_n-v1_{1,n})=Av',v'=(I_n-v1_{1,n})\end{cases} \tag{11}$$

Step3: Decompose the eigenvalue of the kernel matrix M:

$$\begin{cases}M=\hat{A}^T\hat{A}=(Av')(Av')^T(Av')=(v')^TA^TAv'\\ M=Q\Delta Q^T\end{cases} \tag{12}$$

Step4: Factorize the rank-r singular value of $\hat{A}$:

$$\hat{A}^r=\left[\hat{A}Q^r(\Delta^r)^{-\frac{1}{2}}\right]\left[(\Delta^r)^{\frac{1}{2}}\right]\left[(Q^r)^T\right]\equiv U^r\Sigma^r(V^r)^T \tag{13}$$

Step5: According to Eq. (11) and Eq. (13), the r most significant kernel principal components of A can be calculated:

$$U^r=Av'Q^r(\Delta^r)^{-\frac{1}{2}}=A\alpha,\alpha=v'Q^r(\Delta^r)^{-\frac{1}{2}} \tag{14}$$

Step6: To provide a reference for the increment of IKPCA, which is helpful to extract the decomposed electricity consumption time series, the mean μ_C of the given new data C and center C can be calculated:

$$\begin{cases}\mu_C=C\left(\frac{1}{n}1_{n,1}\right)=Cw,w=\left(\frac{1}{n}1_{n,1}\right)\\ \hat{C}=C(I_n-w1_{1,n})=Cw',w'=(I_n-v1_{1,n})\end{cases} \tag{15}$$

Step7: Calculate the mean μ_D of the overall data $D=[U^rC]$ and center D:

$$\begin{cases}\mu_D=\frac{n}{n+i}\mu_A+\frac{i}{n+i}\mu_C=[A\ C]\frac{1}{n+i}\begin{bmatrix}nv\\ iw\end{bmatrix}=\overline{A}\overline{v}\\ \hat{D}=D-\mu_D\end{cases} \tag{16}$$

Step8: Construct matrix $E^{\sim}$:

$$E^{\sim}=\left[\hat{C}\sqrt{\frac{ni}{n+i}}(\mu_A-\mu_C)\right]=[A\ C]\left[\begin{bmatrix}0_{n,i}\\ w'\end{bmatrix}\sqrt{\frac{ni}{n+i}}[\ v-w]\right]=\overline{A}r \tag{17}$$

Step9: Calculate the scaled covariance matrix of D:

$$S_D=\hat{D}(\hat{D})^T=\begin{bmatrix}A^{\hat{r}} & E^{\sim}\end{bmatrix}\begin{bmatrix}A^{\hat{r}} & E^{\sim}\end{bmatrix}^T\begin{bmatrix}\Sigma^r & L\\ 0_{i+1,r} & K\end{bmatrix}\begin{bmatrix}v^r & 0_{n,i+1}\\ 0_{i+1,r} & I_{i+1}\end{bmatrix}^T \tag{18}$$

where

$$\begin{cases}L=(U^r)^TE^{\sim}=\alpha^TA^T\overline{A}r\\ H=E^{\sim}-U^rL=[A\ C]\begin{bmatrix}r_{1:n,:}\\ r_{(n+1):(n+i),:}\end{bmatrix}-\alpha L=\overline{A}\beta\end{cases} \tag{19}$$

decompose the eigenvalue of the kernel matrix M_H of H:

$$\begin{cases}M_H=\beta^T\overline{A}^T\overline{A}\beta=\beta^T\overline{M}\beta\\ M_H=Q_H\Delta_H(Q_H)^T\end{cases} \tag{20}$$

then J and K can be calculated:

$$\begin{cases}J=\overline{A}\beta Q_H(\Delta_H)^{-\frac{1}{2}}\\ K=(\Delta_H)^{\frac{1}{2}}(Q_H)^T\end{cases} \tag{21}$$

Then, according to Ref. [54] the extracted electricity consumption time series (IKPCs) extracted by IKPCA can be encoded as images by GAFs as follows:

Step1: The IKPCs series $X=[x_1,\ \ldots,\ x_n]$ is rescaled into the scope of [0,1] by Eq. (22). The reason why X is normalized between 0 and 1 is to ensure that the encoded images can be converted into electricity consumption time series by Eq. (31) when the forecasting result is output.

$$\tilde{x_i}=\frac{x_i-\min(X)}{\max(X)-\min(X)} \tag{22}$$

Step2: Transform the rescaled series $X^{\sim}$ into polar coordinates:

$$\begin{cases} \varphi = \arccos\left(x_i^{\sim}\right), 0 \le x_i^{\sim} \le 1 \\ r = \dfrac{t_i}{N}, t_i \in \mathbb{N} \end{cases} \tag{23}$$

Here t_i means the time stamp and N means a constant factor with the function of regularizing the polar coordinate system span. And then the temporal correlation within different time intervals can be identified by Gramian Angular Difference/Summation Field (GADF/GASF):

$$\begin{cases} GADF = \left[\sin\left(\varphi_i - \varphi_j\right)\right] \\ \quad = \sqrt{I - \left(X^{\sim}\right)^2}' \cdot X^{\sim} - \left(X^{\sim}\right)' \cdot \sqrt{I - \left(X^{\sim}\right)^2} \\ GASF = \left[\cos\left(\varphi_i + \varphi_j\right)\right] \\ \quad = \left(X^{\sim}\right)' \cdot X^{\sim} - \sqrt{I - \left(X^{\sim}\right)^2}' \cdot \sqrt{I - \left(X^{\sim}\right)^2} \end{cases} \tag{24}$$

Here I means the unit row vector [1, 1, …,1].

Step3: Obtain quasi-Gramian matrice of GADF/GASF by calculating the inner product as follows:

$$\begin{cases} \langle x, y \rangle_{GADF} = \sqrt{1 - x^2} \cdot y - x \cdot \sqrt{1 - y^2} \\ \langle x, y \rangle_{GASF} = x \cdot y - \sqrt{1 - x^2} \cdot \sqrt{1 - y^2} \end{cases} \tag{25}$$

2.2. Principles of ensemble CGAN

Excessive input variables may decrease the forecasting performance [50], while removing redundant variables can enhance the forecasting performance [51] (test in ***CASE II***). Therefore, the time series of initial variables need to be pre-processed. As shown in Fig. 1, the original time series are first decomposed by variational mode decomposition (VMD), and then a series of sub-components (with more stable variances and fewer outliers) can be obtained, which helps enhance the forecasting performance [48,49]. In other words, the possible redundant information is eliminated. After that, the incremental kernel principal component analysis (IKPCA) is applied to extract the incremental kernel principal components (IKPCs) from the band-limited intrinsic mode functions (BLIMFs) decomposed by VMD. That is to say, the possible excessive information is eliminated. The advantage of IKPCA is that it not only good at extracting the interrelation between multiple variables but also could sustain the original information. Subsequently, the extracted results are encoded into images by Gramian Angular Difference/Summation Field (GADF/GASF). Due to the influence of the network structure, the input features are encoded into images. In other words, CGAN (details see Section 2, Chapter 5) is developed based on image processing [37]. Next, the encoded images serve as the input of CGAN, which implies the start of training process. HC is composed of internal and leaf nodes, labeled with a set of weights (W), which occur at the levels (L) of HC. To integrate each sub-CGAN for load forecasting, the levels represent the forecasting errors of each sub-CGAN, while the weights represent the weight coefficients on the occurrence of the forecasting error of the l-th sub-CGAN. Then HC begins to construct a tree with L and different weight coefficients (W) to calculate the minimum value of WPL, that is, the minimum value of the forecasting error of all integrated sub-CGANs. Therefore, the maximum value of the forecasting accuracy can be calculated using 1 minus the minimum value of the integrated forecasting error. The detailed integrated steps are as follows according to Ref. [56].

Step1: Initialize the weight coefficients.
Step2: Sort the weight coefficients in ascending order.
Step3: Substitute the two least-weight nodes with their sum (internal node).
Step4: Repeat ***Step 3*** until a single weight coefficient is left.
Step5: Calculate WPL using the following equation.

$$WPL = w_1 l_1 + w_2 l_2 + \ldots + w_n l_n \tag{26}$$

Here $\{l_1, l_2, \ldots, l_n\}$ represent the forecasting errors of all sub-CGANs, $\{w_1, w_2, \ldots, w_n\}$ represent the correlative weight coefficients.

2.3. Principles of cooperative learning among each sub-CGAN

To reduce the training time of the integrated CGANs, the elitist search strategy of MPGA [58] is introduced to realize the cooperative learning among each sub-CGAN in the ensemble model. As shown in Fig. 2, each sub-CGAN serves as a sub-population of MPGA. The training error of CGAN serves as the fitness value. For the generator models (G_1, G_2, …, G_n), in the horizontal direction, every N_{it} iteration, each sub-CGAN learns weights from other sub-CGANs by Eq. (27). This process is a crossover operation that simulates MPGA.

$$\begin{aligned} W'_{sub-CGAN(i)} &= c_1 W_{sub-CGAN(i)} \\ &+ (1 - c_1)\left[W_{sub-CGAN(i)} - W_{sub-CGAN(j)}\right] \end{aligned} \tag{27}$$

Here c_1 is a random number within the scope of [0,1].

The sub-CGAN i chooses sub-CGAN j as the object for cooperative learning according to the difference degree (DD, defined by Eq. (28)) of the average errors. This process is a migration operation that simulates MPGA.

$$\begin{cases} DD_{sub-CGAN(i,j)} = Error_{sub-CGAN(i)} - Error_{sub-CGAN(j)} \\ j = \underset{1 \le j \le sn}{\operatorname{argmax}}\left(DD_{sub-CGAN(i,j)}\right) \end{cases} \tag{28}$$

Moreover, in N_{it} iterations, if the training error of a sub-CGAN does not decrease for N_c consecutive times, the sub-CGAN generates new weights randomly by Eq. (29). This process is a mutation operation that simulates MPGA.

$$\begin{aligned} W'_{sub-CGAN(i)} &= c_2 W_{sub-CGAN(i)} \\ &+ (1 - c_2) Random_{matrix} \end{aligned} \tag{29}$$

Here c_2 is a random number in the scope of [0,1]. $Random_{matrix}$ means the randomly generated weight matrix.

However, in the vertical direction, the best weights in each sub-CGAN are selected and saved to Elitist Population G by an artificial selection. The principles for cooperative learning in the discriminator models (D_1, D_2, …, D_n) are the same as those in the generator models (G_1, G_2, …, G_n) except that the best weight in each sub-CGAN is saved to Elitist Population D.

2.4. The process of load forecasting based on the proposed method

Step1 Obtain input features. Decompose the initial input variables affecting the load by using VMD, extract IKPCs from the decomposed results (BLIMFs). The extracted IKPCs (in daily load data) are encoded as images by GAFs. The images are divided into training set and test set.
Step2 Initialize parameters. Assume the total number of dragonflies to be N_d, the max iteration of IDA to be T_{max}, the number of sub-groups of dragonflies to be N_s, and the number of sub-

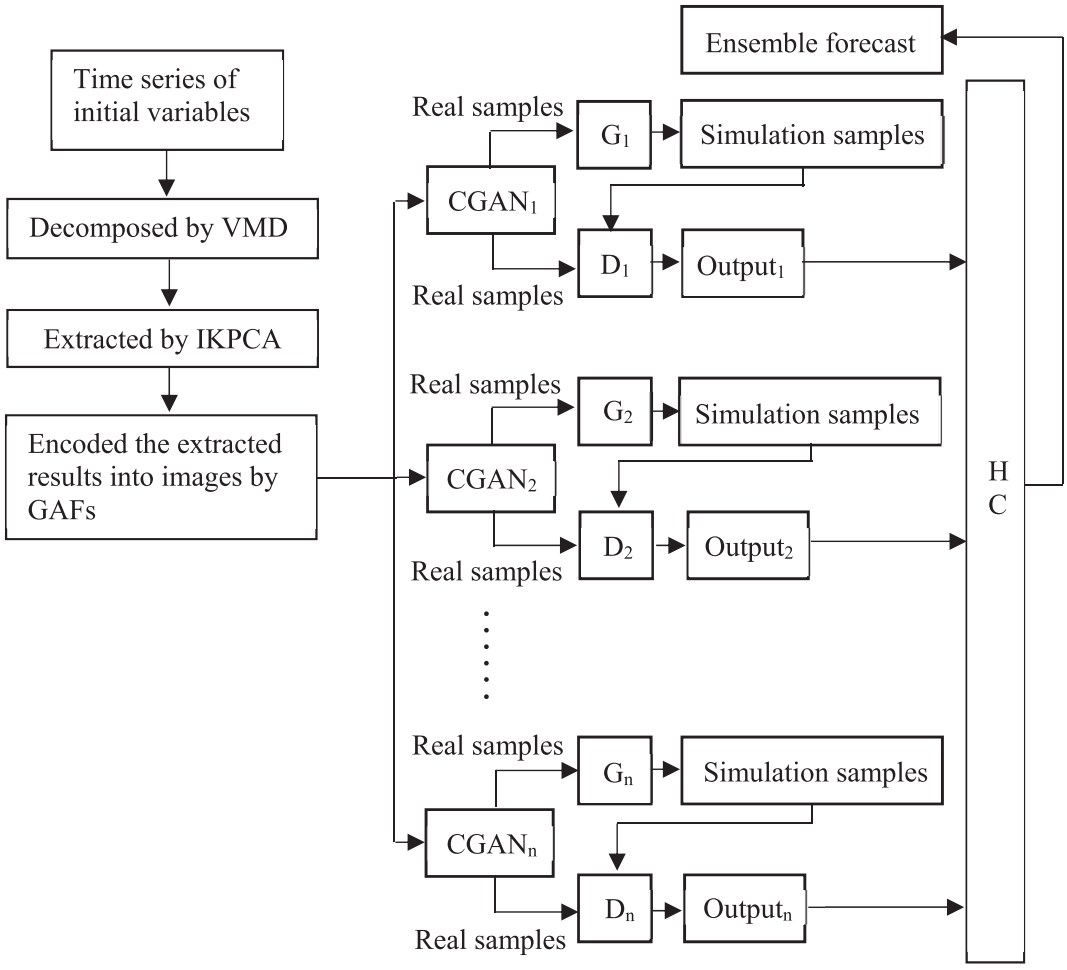

Fig. 1. Principles for ensemble CGANs.

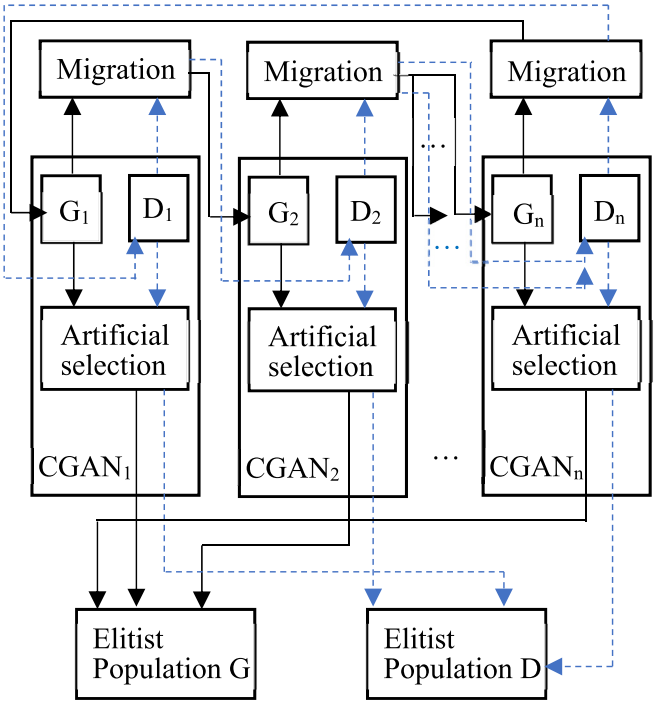

Fig. 2. Communication strategies among each sub-model.

CGAN to be n. The location of the dragonflies at the $i^{\text{-th}}$ iteration can be expressed as:

$$X = [w_1, w_2, \ldots, w_n,]^T \tag{30}$$

Here, w whose values are all within the scope of [0,1], represents the weight matrix of each sub-CGAN.

Step3 Train models. Perform the communication strategy in Section 2, Chapter 3 to realize the cooperative learning among each sub-CGAN in the ensemble model. According to Ref. [54], the output images of the discriminator model (D) of each sub-CGAN can be recovered as follows:

$$\cos\varphi = \sqrt{\frac{\cos(2\varphi)+1}{2}}, \varphi \in \left[0, \frac{\pi}{2}\right] \tag{31}$$

Then the forecasting errors of all sub-CGANs can be obtained compared with the actual load.

Step4 Integrate forecast. The CGANs are integrated by HC. The forecasting errors of each sub-CGAN serve as external nodes of HC, and the weight w at which the forecasting error of the $l^{\text{-th}}$ sub-CGAN occurs is optimized by IDA (details see Section 2, Chapter 8). Calculate the ensemble error.

Step5 Judge whether to meet the expected accuracy. If the upper limit of iterations or expected accuracy is not reached, turn to ***Step3***, or else, output the forecast results.

2.5. Principles of CGAN

As shown in Fig. 3, CGAN consists of two sub-models (CNNs): the generator model (G) and the other is the discriminator model (D). G was applied to deal with the variability in people's electrical behavior and weather forecast errors (sampled from a noise vector z), and to generate RSD (defined as $X_{\text{generated}} = G(z)$) using the condition y, z and training set; y serves as a label, which is used to distinguish different load types (regular day, Saturday, Sunday and holiday). D is applied to distinguish RSD and real data set. Therefore, the loss function L, can be defined:

$$\begin{aligned} L &= E_{x\sim P_{data}}[\log p(s = real|x_{real})] + \\ &E_{z\sim P_{(z)}}\left[\log p\left(s = generated\middle|x_{generated}\right)\right] = \\ &E_{x\sim P_{data}}[\log(D(x|y))] + E_{z\sim P_{(z)}}[\log p(1 - D(G(z|y)))] \end{aligned} \tag{32}$$

Here P_{data} represents the real distribution of training set, $E_{x\sim Pdata}$ is the expectation that x comes from P_{data}, $P(z)$ represents the prior distribution on z, $E_{z\sim P(z)}$ represents the expectation that z is sampled from noise, $D(x)$ represents the probability that x comes from real data set X_{real}. The task of G is to minimize $E_{z\sim P(z)}$ to prevent D from correctly distinguishing from RSD and real data set; while the task of D is to maximize L to ensure real data set X_{real}. This likes an adversarial process with an objective function defined by Eq. (33), when the objective function value reaches 0.5, the model performs optimally [37].

$$fun = \arg\min_{G}\max_{D} L(G, D) \tag{33}$$

2.6. Principles of CNN

G and D of CGAN proposed in this paper are both CNN. As shown in Fig. 4, according to Ref. [41] a basic CNN is composed of convolutional layer, subsampling layer, full-connected layer and output layer. A convolutional layer extracts the feature maps of input images using convolutional kernel operations.

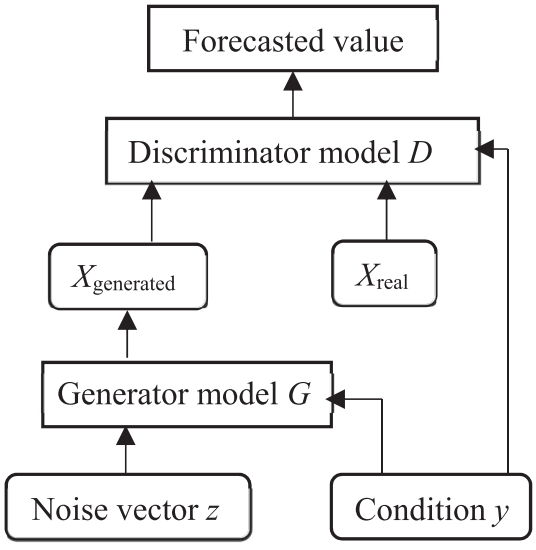

Fig. 3. The structure of CGAN.

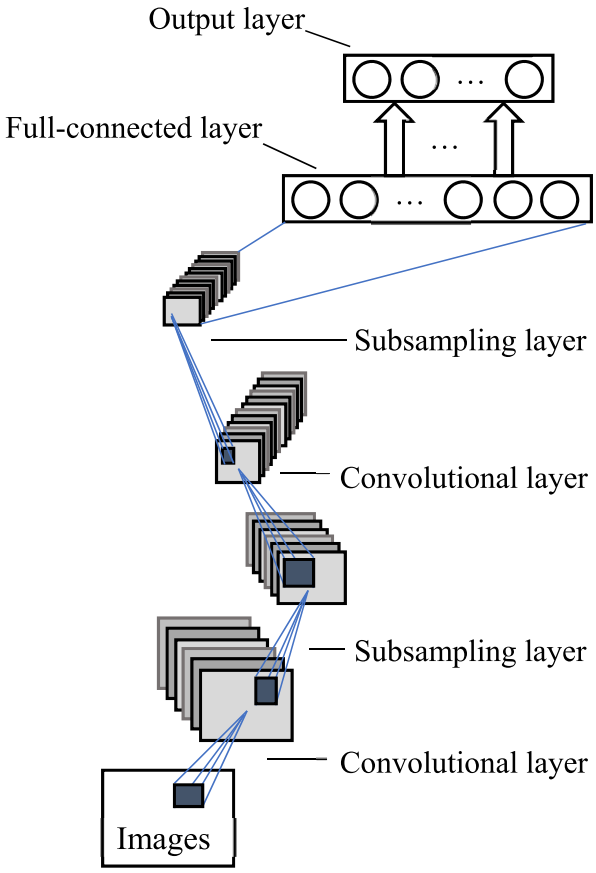

Fig. 4. The structure of CNN.

$$x_j^l = f\left(\sum_{i\in M_j} x_i^{l-1} * w_{ij}^l + b_j^l\right) \tag{34}$$

here x_i^{l-1} means the $i^{\text{-th}}$ feature map of the $(l\text{-}1)^{-\text{th}}$ layer; w_{ij}^l refers to the convolutional kernel (weight matrix) connecting the $i^{\text{-th}}$ feature map of the $(l\text{-}1)^{-\text{th}}$ layer and the $j^{\text{-th}}$ feature map of the $l^{\text{-th}}$ layer; b_j^l represents the $j^{\text{-th}}$ bias term of the $(l\text{-}1)^{-\text{th}}$ layer; $*$ means the convolutional operation.

At the subsampling layer, the feature map is divided into a series of subareas, then a down-sampling operation is applied to reduce redundant features:

$$y_j^l = f\left(\sum_{i\in M_j} down\left(x_j^{l-1}\right) * \alpha_j^l + c_j^l\right) \tag{35}$$

Here down$(\cdot)$ means down-sampling operation; the value of down-sampling function is an average pooling. And then the full-connected layer serves as a neural network (NN) to train the features extracted by down-sampling operation. At last, the forecasting results are output by the output layer.

2.7. Principles of DA

Dragonfly algorithm (DA) [61] is a novel intelligent optimization algorithm that simulates the three behaviors (separation, alignment, and cohesion, defined by Eqs. 36–38, respectively) of dragonflies in nature.

$$S_i = -\sum_{j=1}^{N} X - X_j \tag{36}$$

$$A_i = \frac{1}{N}\sum_{j=1}^{N} V_j \tag{37}$$

$$C_i = \frac{1}{N}\sum_{j=1}^{N} X_j - X \tag{38}$$

Here N means the number of neighbouring dragonfly individuals, X means the location of the current dragonfly individual; while V_j and X mean the velocity and location of the j-th neighbouring dragonfly individual, respectively. Then all dragonfly individuals fly towards a food source according to Eq. (39), in flying process, a strategy to avoid enemy's attach is simulated by Eq. (40).

$$F_i = X^+ - X \tag{39}$$

$$E_i = X^- + X \tag{40}$$

Here X^+ and X^- mean the locations of the food source and the enemy, respectively; then the dragonfly individual locations can be updated as follows:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \tag{41}$$

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{42}$$

Here s, a, and c mean the separation weight, alignment weight, and cohesion weight, respectively; while f, e, and w mean the food factor, enemy factor, and inertia weight, respectively. However, when there is no neighbouring dragonfly individuals, Levy fights is introduced into Eq. (43) to update the locations of dragonfly individuals:

$$X_{t+1} = X_t + \text{Lévy}(d) \times X_t \tag{43}$$

2.8. Principles of IDA

In this paper, multiple population is introduced into DA to improve its convergence performance, that is, all dragonflies are divided into several subgroups. The individual with the best fitness among all subgroups serves as a leader, and all the other individuals move towards the leader by using the attractiveness (Eq. (44)) and attracted movement (Eq. (45)) in firefly algorithm (FA) [62]. Therefore, Eq. (41) is modified to Eq. (46).

$$\beta = \beta_0 e^{-\gamma r^2} \tag{44}$$

$$x_i(t+1) = x_i(t) + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + b\varepsilon_i \tag{45}$$

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t + \beta_0 e^{-\gamma r_{ij}^2}(x_{best} - x_i) + b\varepsilon_i \tag{46}$$

Here β and γ mean the attractiveness and light absorption coefficient, respectively; $r_{ij} = \|x_{best} - x_i\|$ means the Cartesian distance between the optimal individual and current individual; b means step factor.

3. Experiment and analysis

Two datasets, gathered from the China's State Grid (which manages national power generation and sales.), are composed of weather conditions, electricity price and residential building data. One comes from a Chinese northern city (Daqing, used in Case I-Case III) and the other comes from a Chinese southern city (Sansha, used in Case IV-Case V). They are used to perform the hourly ahead load forecasting in this paper. The two datasets are both sampled every 15 min. Many researches are convinced that treating regular workdays, weekends and holidays with different methods can obtain higher forecasting performance [9,64]. Therefore, the two datasets are both divided into four sub-datasets by virtual tags, based on regular workdays, Saturday, Sunday and holidays, respectively. In other words, Tag 1 represents regular workdays; Tag 2 represents Saturday; Tag 3 represents Sunday; Tag 4 represents holidays. Moreover, if weekends are holidays, the weekends would be treated as holidays. The same parameters (shown in Table 1), are used to the tested models and the following evaluation functions were selected to evaluate the performance of the tested models, in this paper.

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i^t - y_i)^2} \\ MAE &= \frac{1}{n}\sum_{i=1}^{n}|y_i^t - y_i| \\ \text{RE} &= \frac{y_i^t - y_i}{y_i} \times 100\% \end{aligned} \tag{47}$$

here y_i^t and $y_i$denote the forecasted and actual value of the load at the i-th tested point, respectively.

Case I. To verify the convergence efficiency of IDA, MPGA [59], dragonfly algorithm (DA) [61], firefly algorithm (FA) [62], and grey wolf optimizer (GWO) [63] are chosen to test the following functions, using the same parameters. The test results (shown in Figs. 5–6) reveal that the convergence effect of IDA is obviously supper to those of the other test methods, verifying the effectiveness of the multiple population strategy introduced into DA. Therefore, IDA is chosen to calculate the weight matrix of the proposed model.

$$\begin{cases} f_5(x) = \sum_{i=1}^{n-1}\left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right), [-30, 30] \\ f_7 = \sum_{i=1}^{n} i x_i^4 + random\left[0, 1\right), x \in \left[-1.28, 1.28\right] \end{cases} \tag{48}$$

Case II. The impact of redundant and excessive information from original variables on the forecasting performance was tested in this

Table 1
Parameters of the tested methods.

Parameters	Values
Maximum training times	6000
Learning rate	0.01
End error	0.000001
Scope of neurons	10–30
Scope of layers	4–12
Max iteration of IDA	500

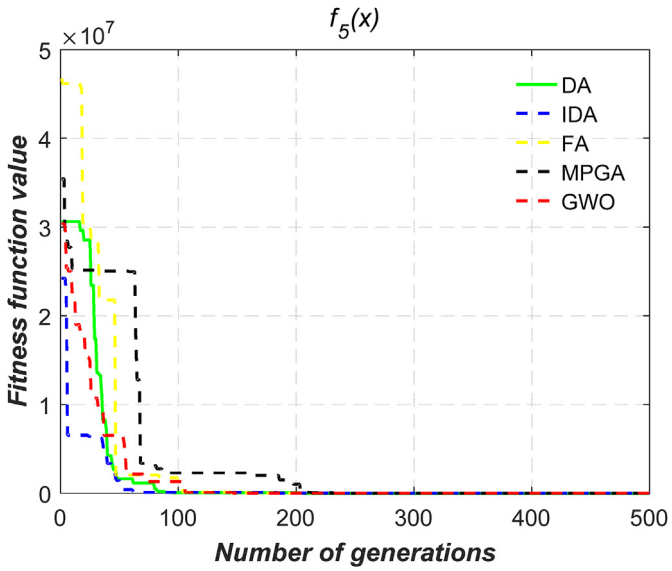

Fig. 5. Convergence performance comparison of $f_5(x)$.

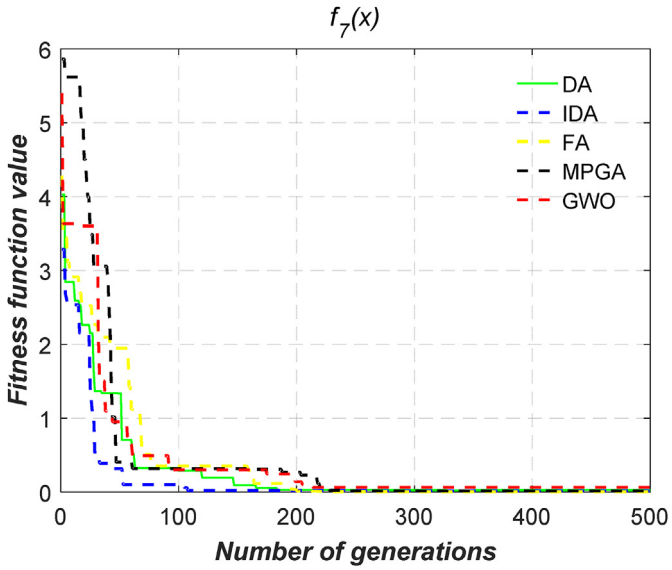

Fig. 6. Convergence performance comparison of $f_7(x)$.

case. A summer month (July, 2012) was chosen as the test set. The data, from July 1, 2009, to June 30, 2012, was used as the training set. The test results are shown in Figs. 7–11. Fig. 7 shows that when the number of BLIMFs $K = 6$, the difference of the center frequency of BLIMF6 minus that of BLIMF5 reaches the minimum. Therefore, it can be considered that when $K = 6$, the best value is just obtained according to Ref. [52], that is, $6 \times 8 = 48$ input variables can be obtained after VMD decomposition. However, excessive variables would decrease the forecasting accuracy [50]. Moreover, 48 input variables may contain redundant information, and removing the redundant information can increase the forecasting accuracy [51].

Here T means temperature; WS means wind speed; H means humidity; BD means building data; P means precipitation; AP means air pressure; EP means electricity price; SR means solar radiation.

To test the impact of excessive and redundant information more fairly and effectively, the authors decomposed the original variables by continuously increasing the number of BLIMFs in the case of fixing the original variables. In other words, the forecasting performance involving the use of original variables directly does not change with the number of BLIMFs. Therefore, the ranges of RE are all straight lines (Figs. 8 and 9). Similarly, the scopes of RMSE and MAE are also constants that do not vary with the number of BLIMFs (Figs. 10 and 11). As shown in Figs. 8 and 9, when K is less than or equal to 6, the scopes (minimum, average and maximum) of relative error (RE) after VMD decomposition gradually decreases as K increases compared with directly using the original variables for forecasting; when K is greater than 6, the test results are opposite. The same phenomenon (the scopes of RMSE and MAE) can also be found in Figs. 10 and 11. These test results indicate that decomposing the original variables can effectively remove the redundant information from original variables when K is not greater than 6, while the decomposed results is excessive when K is greater than 6.

Moreover, compared with VMD decomposition for forecasting, whether K is greater than 6 or not, as K rises, the scopes of RE after IKPCA extracting gradually declines (see Figs. 8 and 9). The same phenomenon can also be observed in Figs. 10 and 11. These test results suggest that IKPCA can effectively extract the VMD decomposition results and remove redundant information.

Case III. The impact of different input features on the forecasting performances of the tested models were tested in this case. The GADF, GASF, VMD, EMD, two wavelets (Daubechies (Db4) and Coiflets (Coif4)) and using variables directly (VD) are tested by using the proposed method (ensemble CGANs-HC), LSTM, and DBN, respectively. A winter month (January, 2015) and a summer month (July, 2015) were tested, respectively. For the winter month, the data, from January 1, 2012, to December 31, 2014, is used as the training set; while for the summer month, the data, from July 1, 2012, to June 30, 2015, is used as the training set. The program was repeatedly operated for 80 times, the 1-h ahead forecasting error ranges are shown in Figs. 12–21.

As shown in Figs. 12–17, the scopes of the relative error (RE) of GADF and GASF are optimal in the three tested methods, both in summer and winter days, compared with other tested input features in this case. As shown in Figs. 18 and 20, the scopes of RMSE and MAE of GADF are all optimal in the three tested models, both in summer and winter months, compared with other tested input features in this case. The reasons behind the good forecasting results achieved with the proposed input feature (GADF and GASF) mainly include:

For one thing, the redundant information in the original electricity consumption time series are reduced by decomposed via VMD. In other words, a series of sub-components (with more stable variances and fewer outliers) has been obtained, which contributes to the enhancement of forecasting performance [48,49].

For another, the feature relationships between the decomposed results by VMD have been extracted via IKPCA. That is, excessive input information is reduced with no loss of original information.

As shown in Figs. 19 and 21, GADF and GASF both take more training time and predicting time than those of other compared input features in the three tested methods in this case, both in summer and winter months. For the summer month, the maxima of training time and predicting time of the proposed are 88.83 and 1.90 s, respectively. For the winter month, the maxima of training time and predicting time are 75.52 and 0.88 s, respectively. The maximum time can meet the demand of 1-h ahead load forecasting. Moreover, the scopes (minimum, average and maximum) of the training time and predicting time of GADF are all superior to those of GASF. The scopes of RMSE and MAE of GADF are all optimal in the three tested models, both in summer and winter months, compared with other tested input features in this case. Therefore,

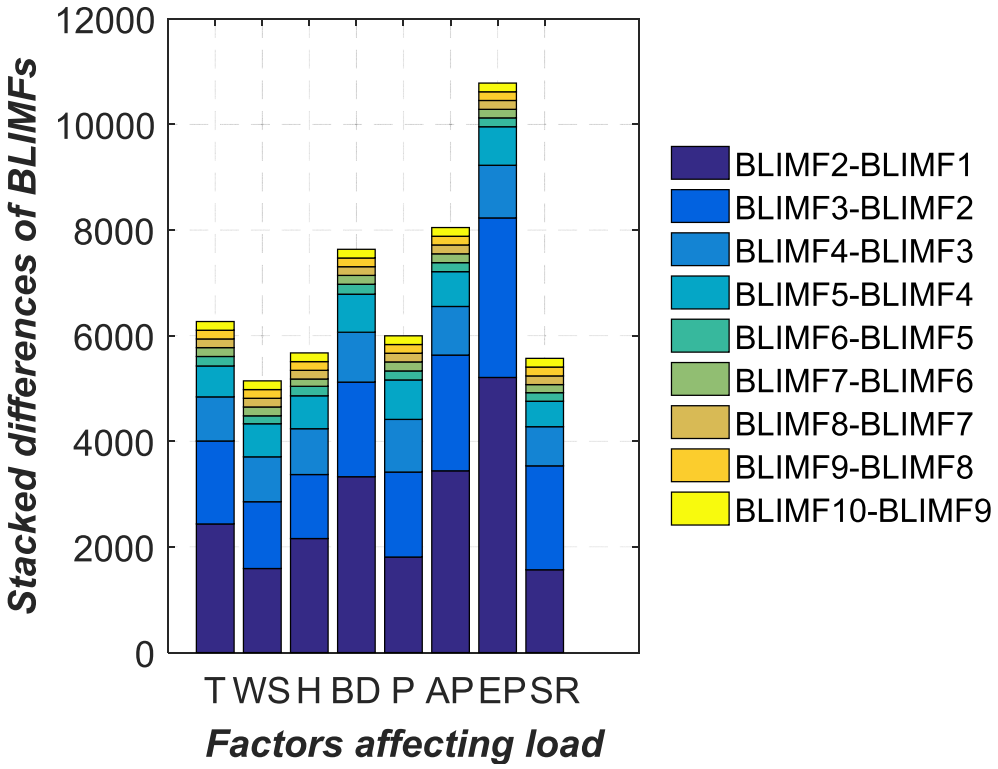

Fig. 7. The decomposed results by VMD.

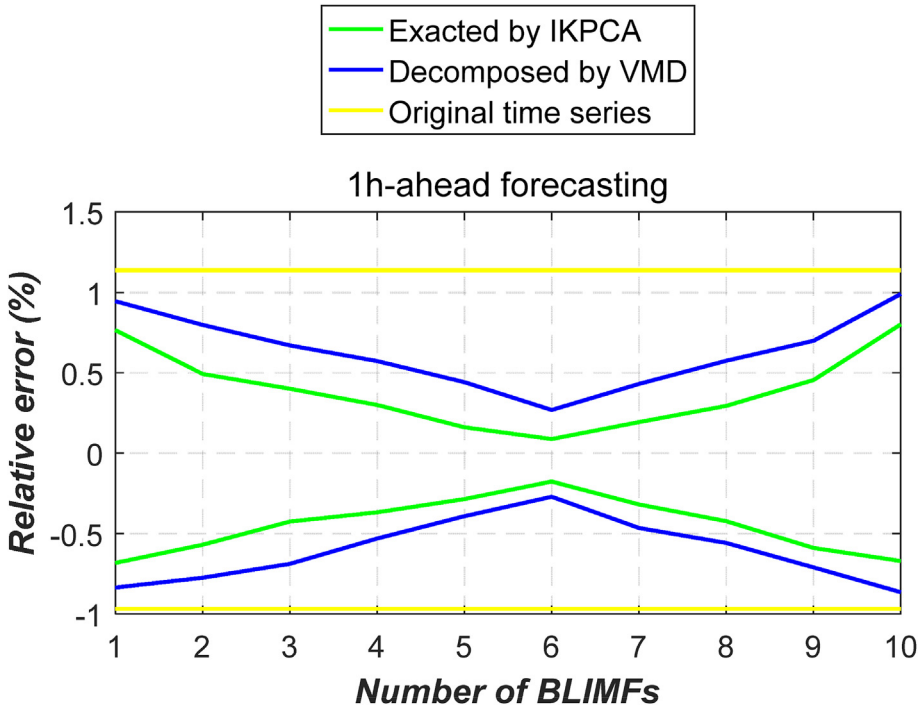

Fig. 8. The redundant information test for 1h-ahead forecasting.

GADF is selected as the new input feature in this paper.

Case IV. The forecasting performances of different ensemble methods were tested in this case. The proposed method, partial least squares regression (PLSR), gradient boosting decision tree (GBDT), random forest (RF) and adaptive boosting (AdaBoost) are compared. It is noted that the weights of all tested methods are calculated by IDA, using the same parameters. The data, from January 1, 2013, to December 31, 2015, is used as the training set, and a winter month (January, 2016) is tested. The program was

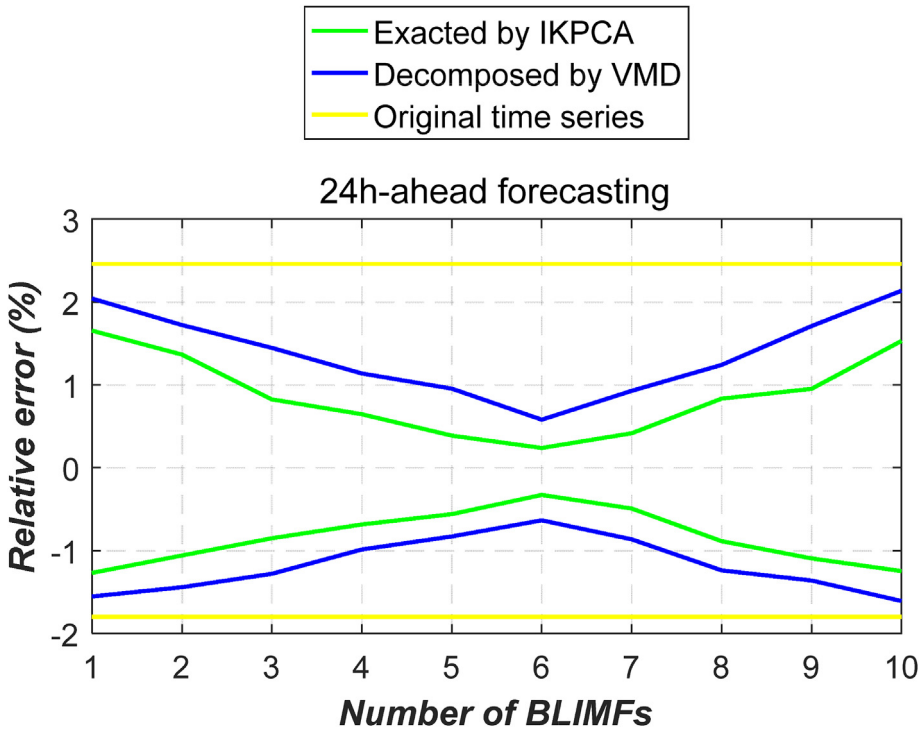

Fig. 9. The redundant information test for 24h-ahead forecasting.

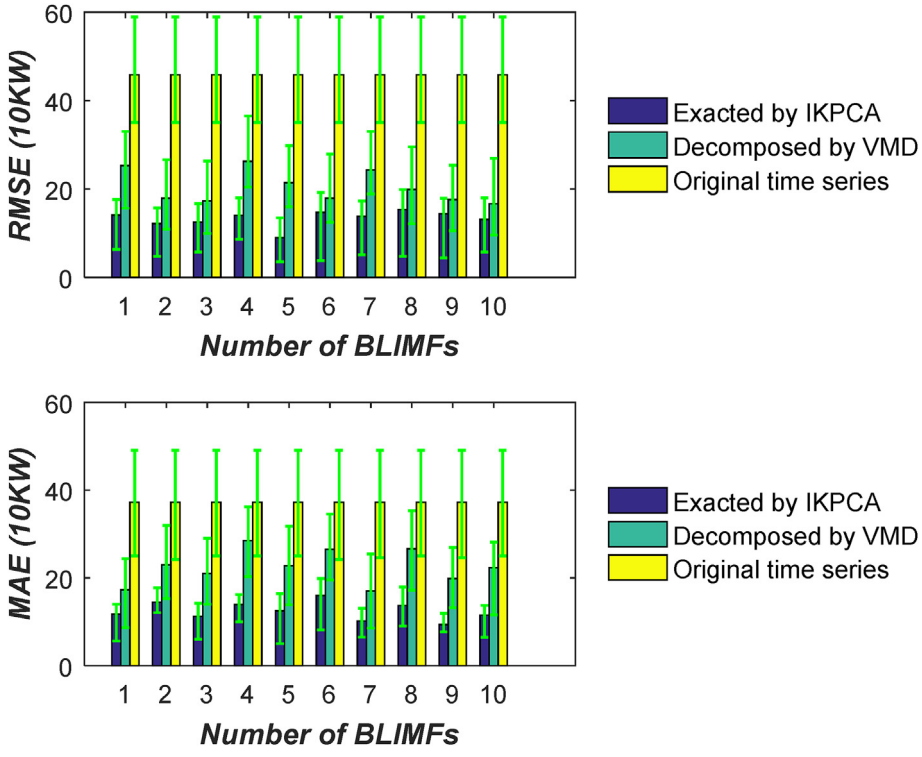

Fig. 10. Test of redundant information for 1h-ahead forecasting.

repeatedly operated for 80 times, the 1-h and 24-h ahead forecasting results are shown in Figs. 22 and 23.

As shown in Fig. 22, the scopes (minimum, average and maximum) of RMSE and MAE of HC are all optimal in all tested models, before and after optimized by IDA. As shown in Fig. 23, the scopes of the training time and predicting time of HC are all optimal in all tested models, before and after optimized by IDA. HC has the advantage of starting with nodes (sub-models) with minimal prediction error and less need for too many nodes (sub-models)

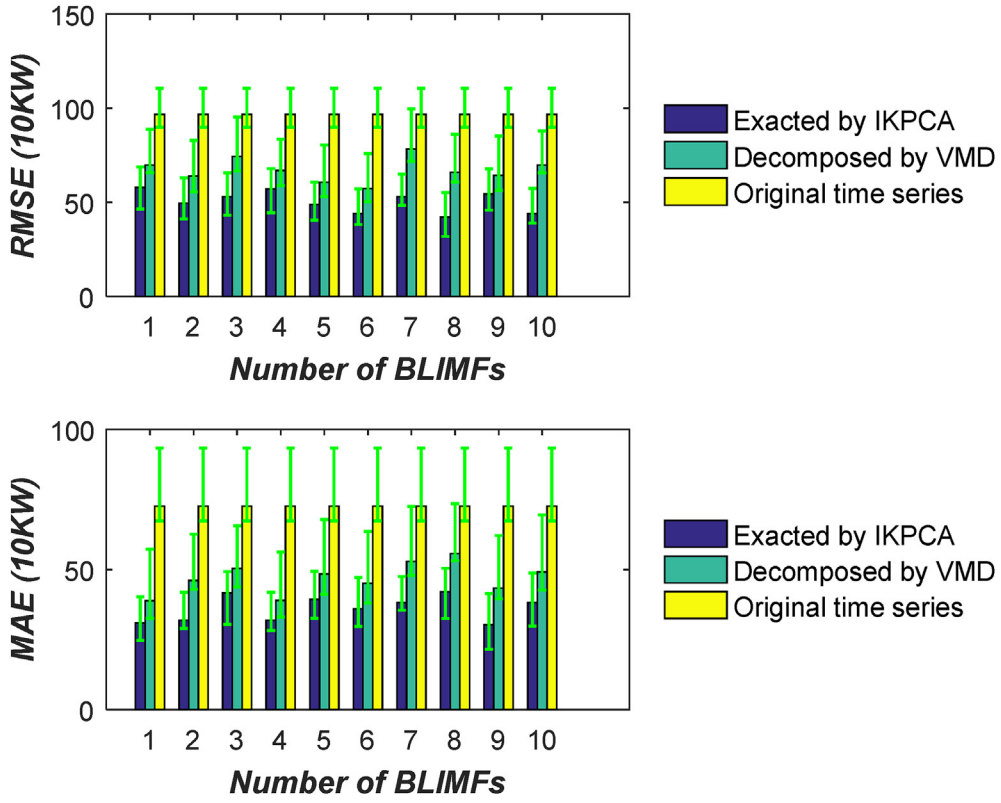

Fig. 11. Test of redundant information for 24h-ahead forecasting.

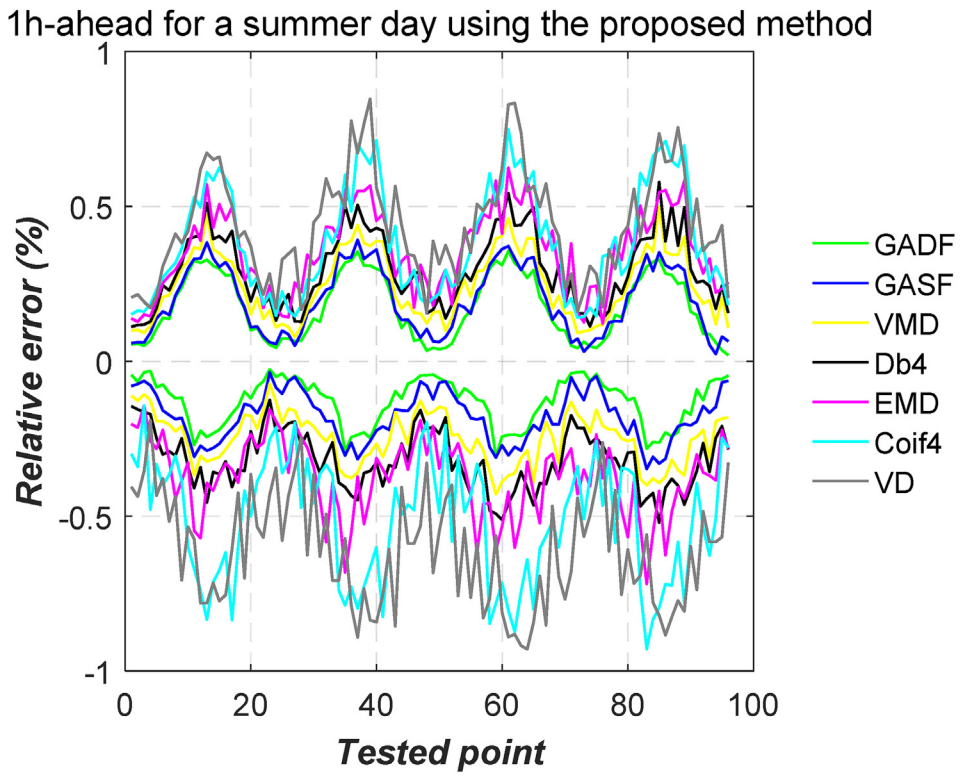

Fig. 12. 1h-ahead test for a summer day using the proposed method.

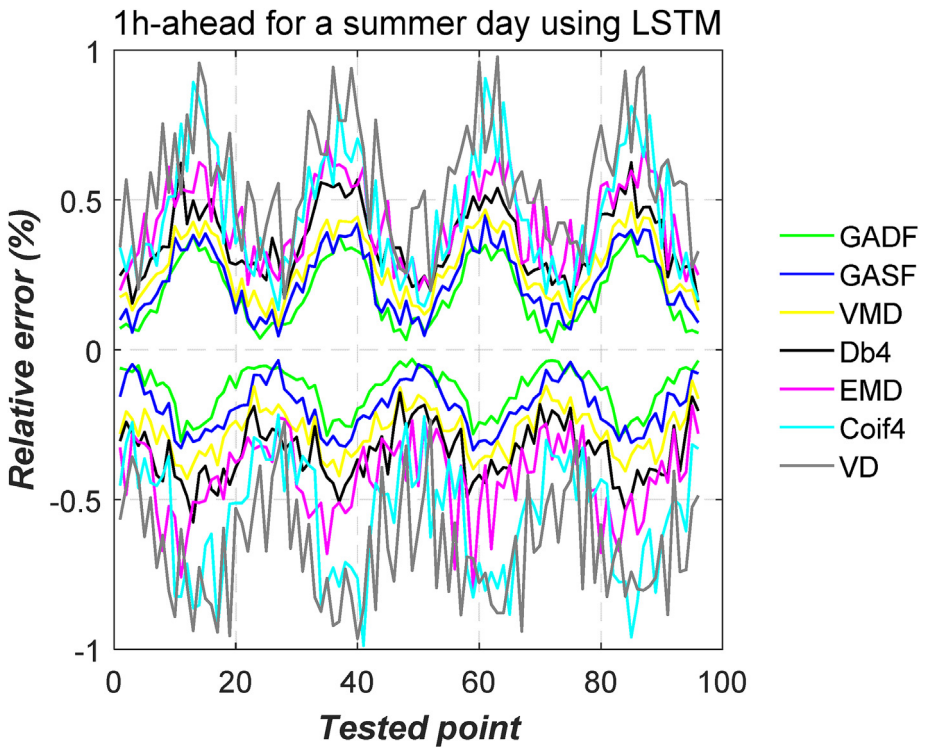

Fig. 13. 1h-ahead test for a summer day using LSTM.

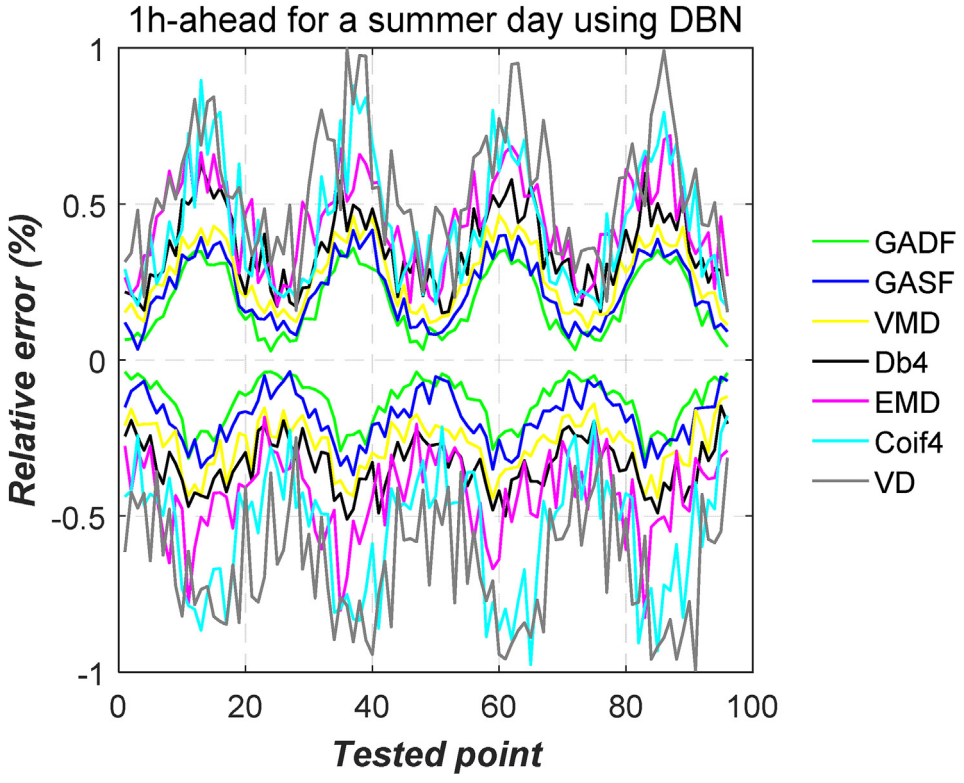

Fig. 14. 1h-ahead test for a summer day using DBN.

compared with the existing ensemble methods tested in ***CASE IV***. The advantage of HC determines that it can enhance forecasting accuracy while cutting the training and forecasting time. After being optimized by the proposed optimization algorithm (IDA), all of the forecasting performance (RMSE and MAE), training time and predicting time have been significantly improved in all tested

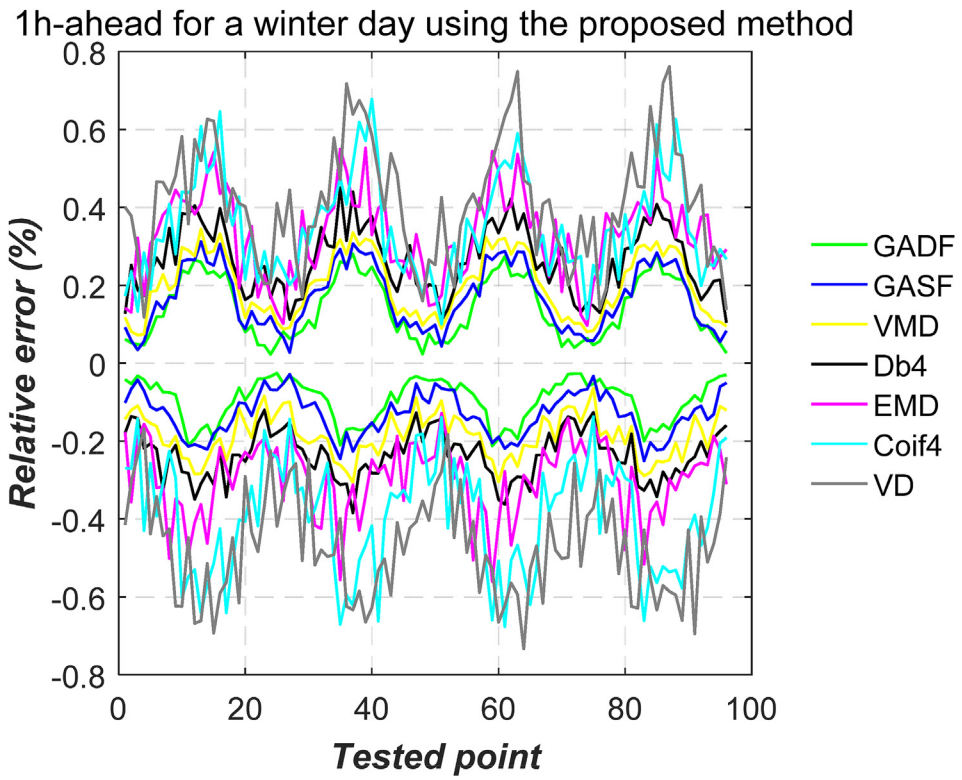

Fig. 15. 1h-ahead test for a winter day using the proposed method.

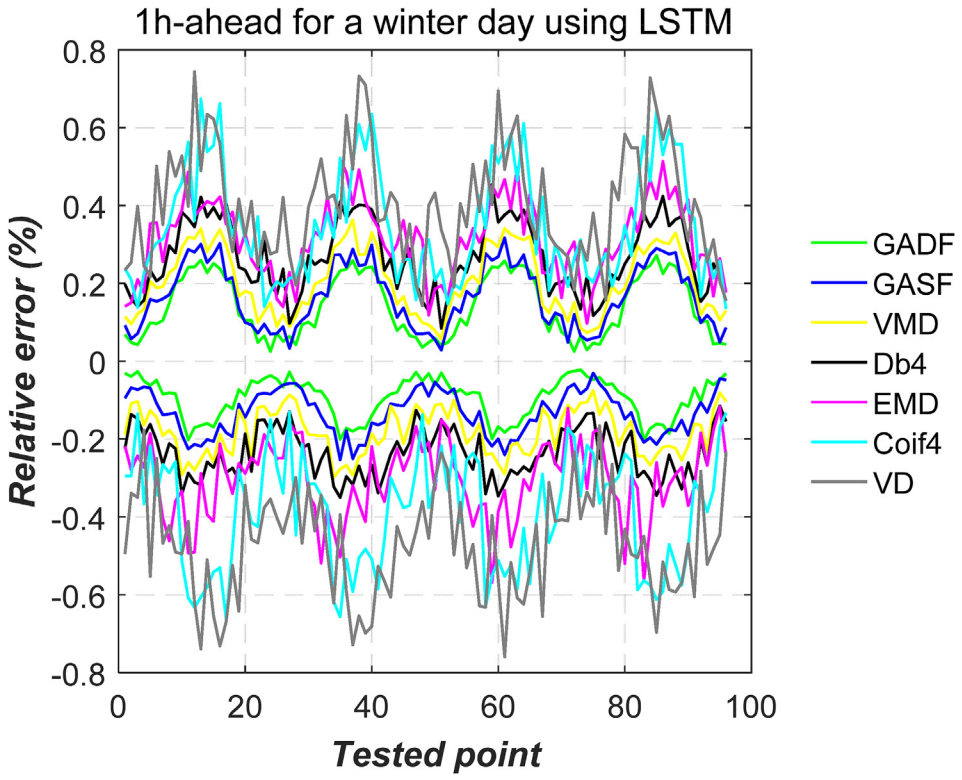

Fig. 16. 1h-ahead test for a winter day using LSTM.

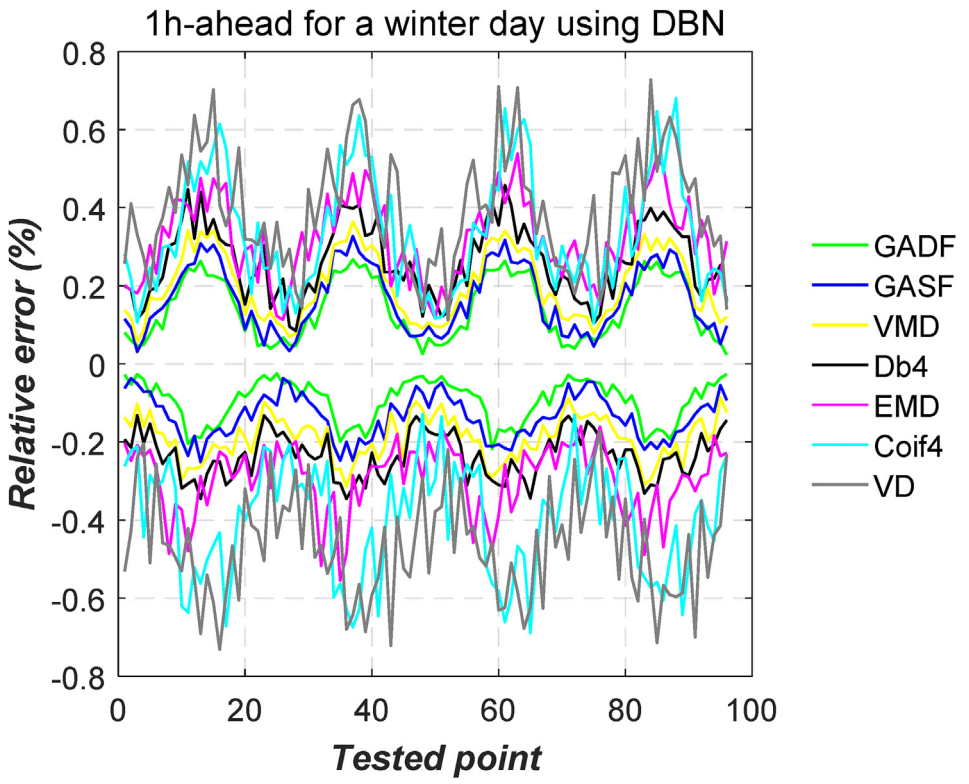

Fig. 17. 1h-ahead test for a winter day using DBN.

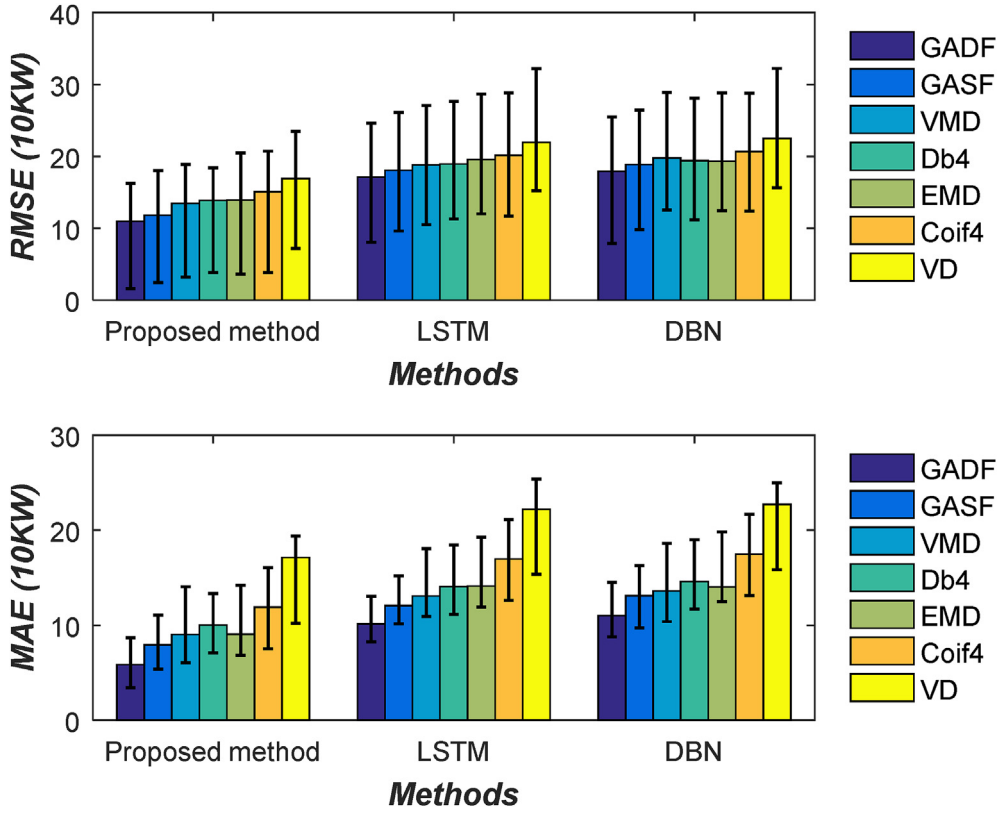

Fig. 18. Comparison of RMSE and MAE for summer month.

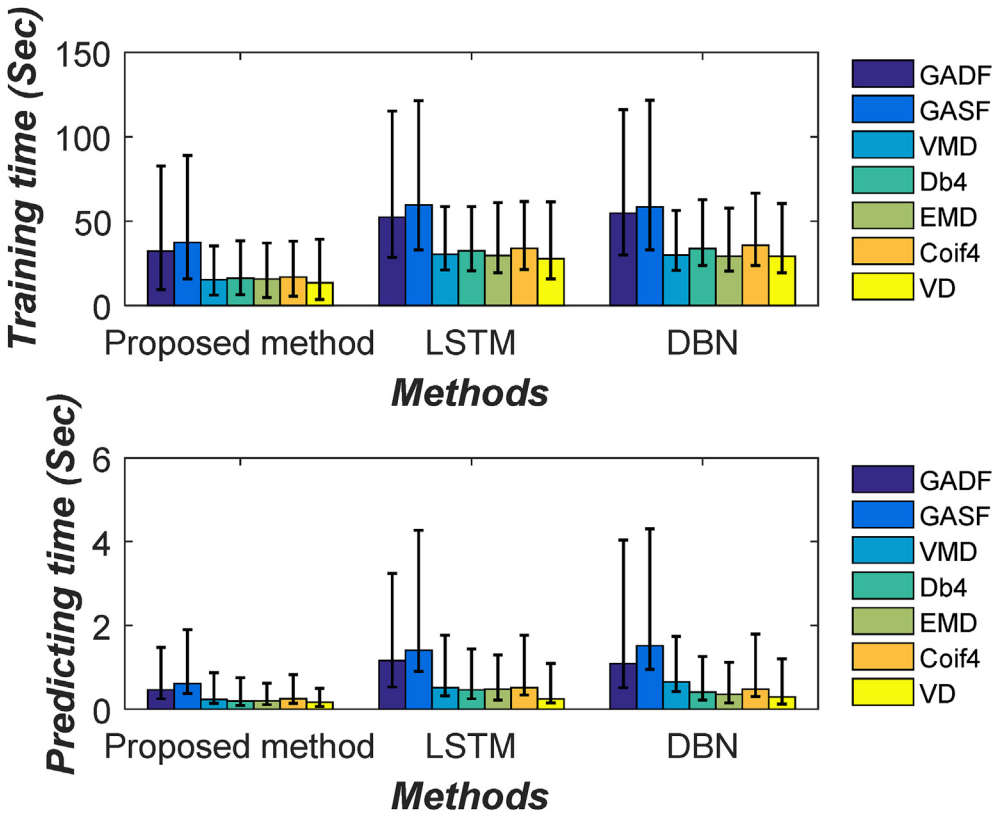

Fig. 19. Time comparison for summer month.

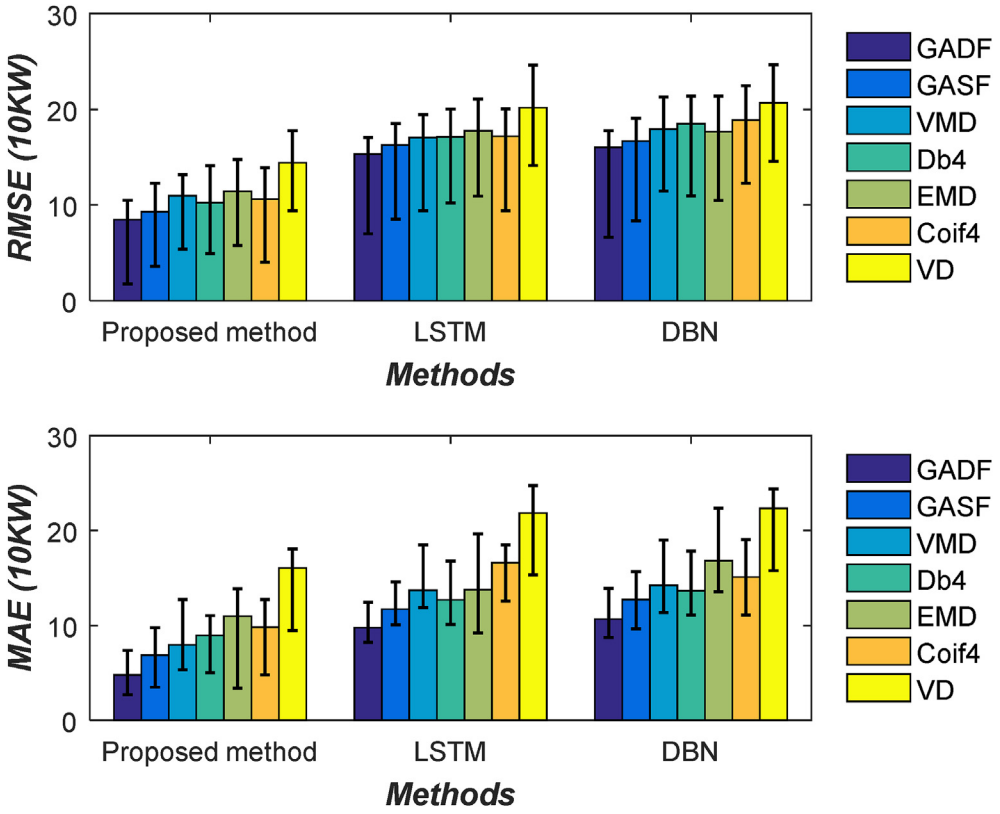

Fig. 20. Comparison of RMSE and MAE for winter month.

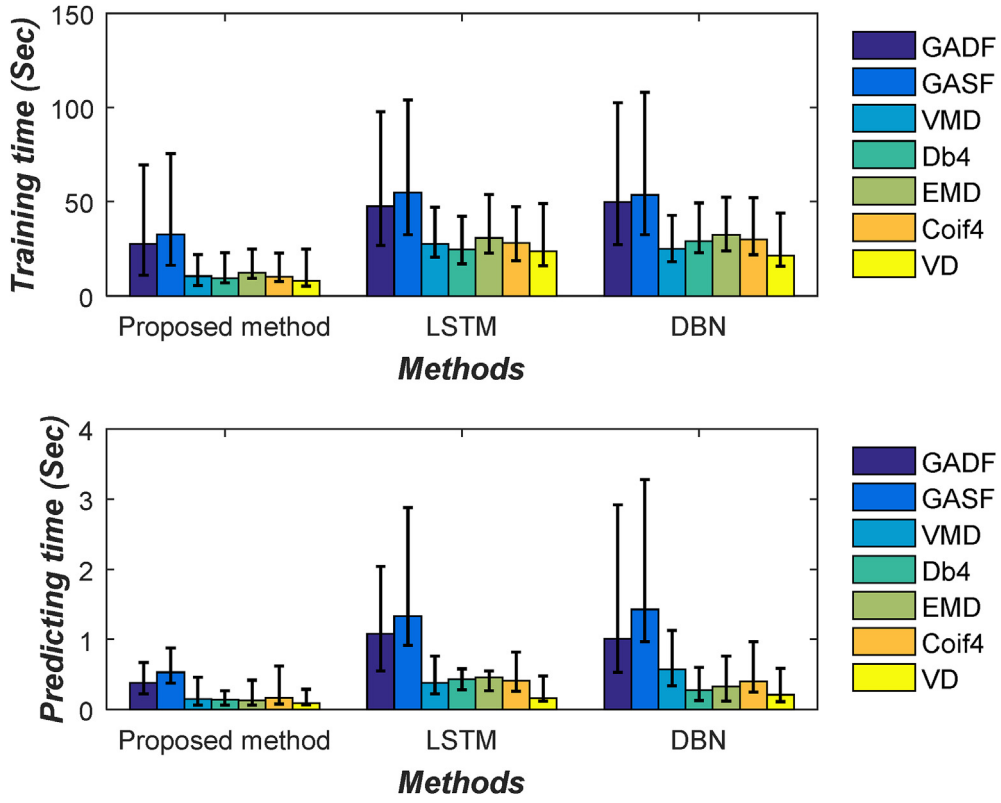

Fig. 21. Time comparison for winter month.

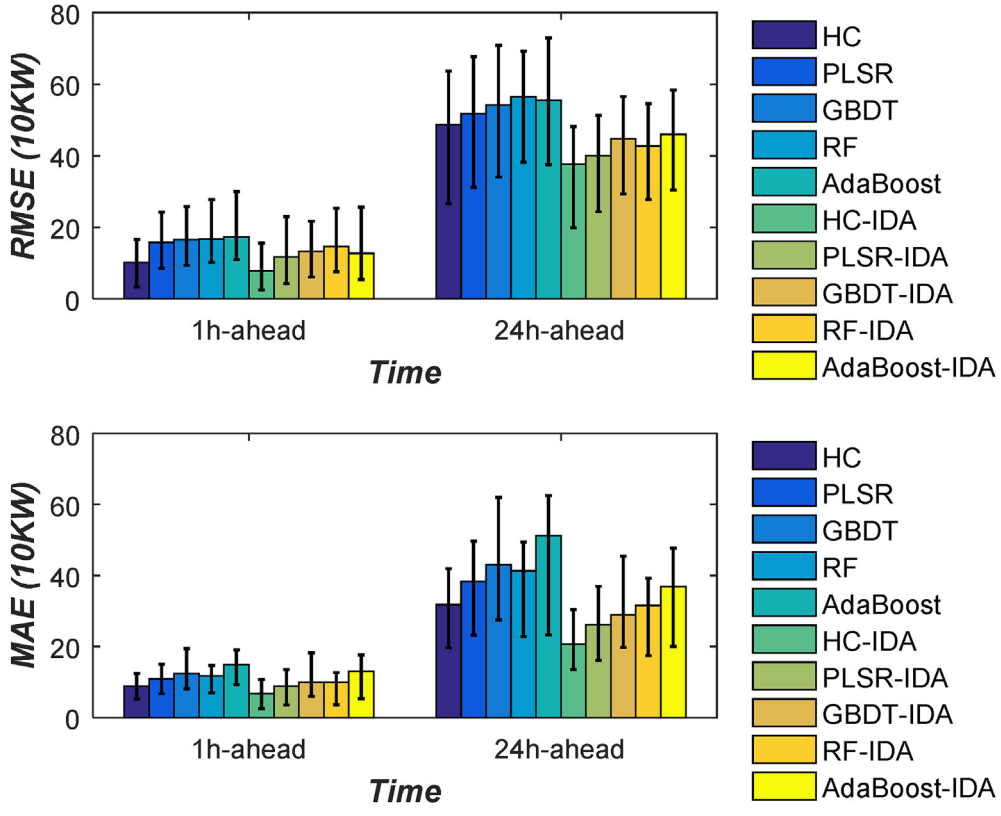

Fig. 22. Comparison of RMSE and MAE of the tested ensemble methods.

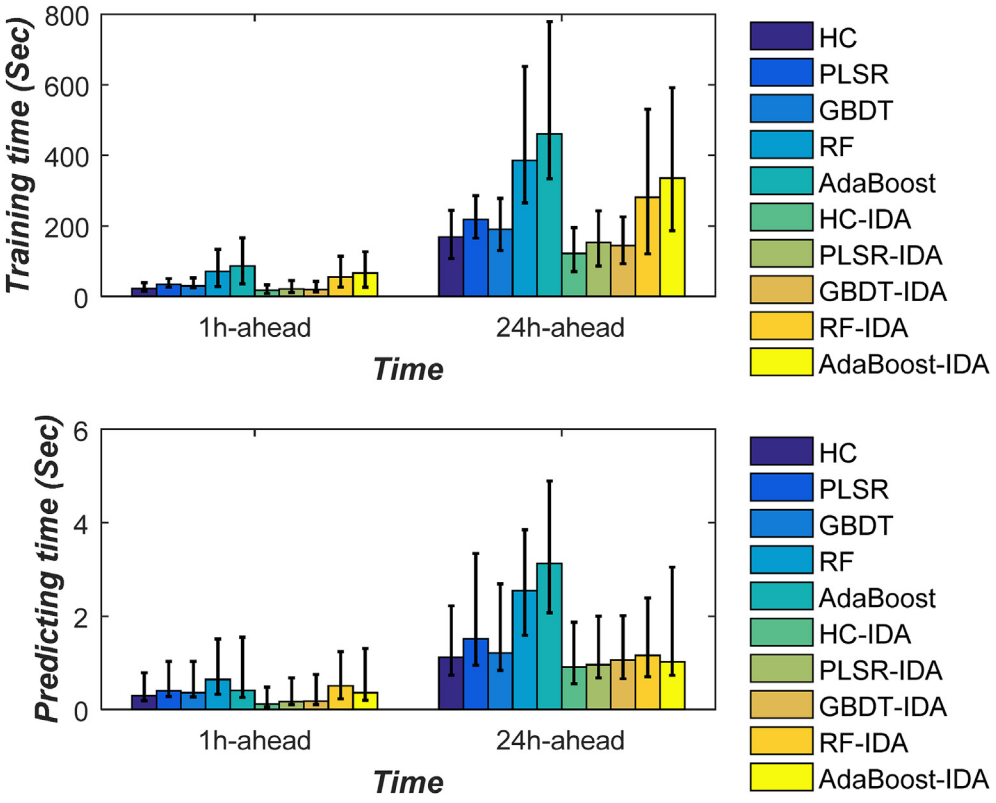

Fig. 23. Time comparison of the tested ensemble methods.

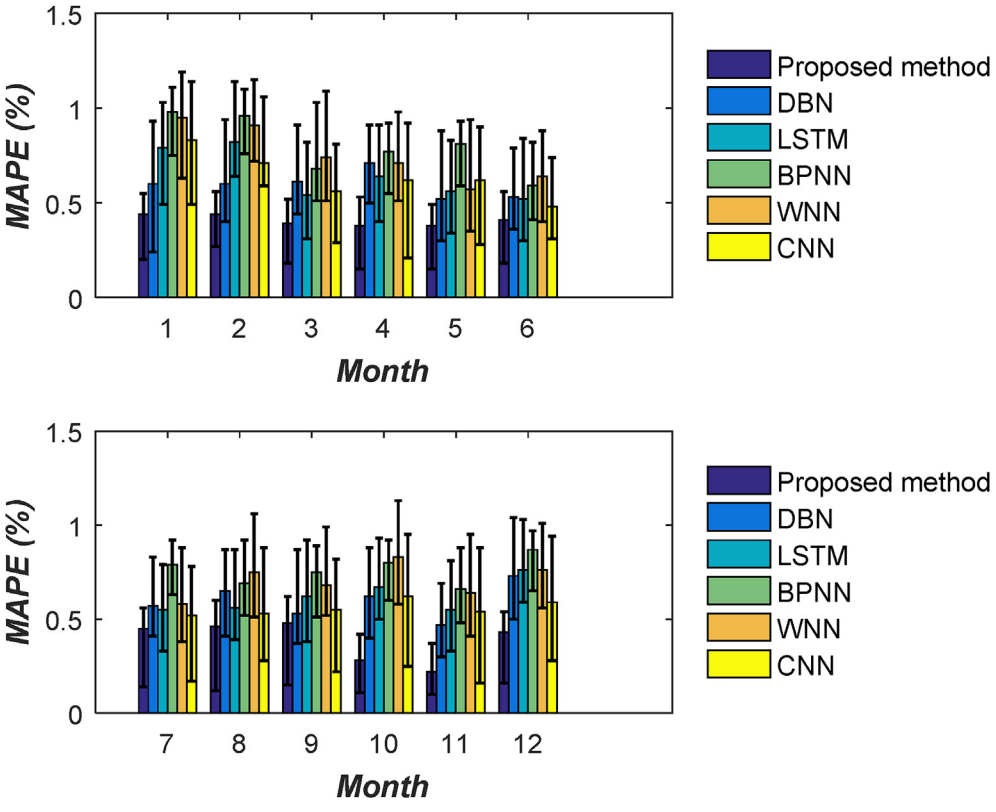

Fig. 24. 1h-ahead test results for month.

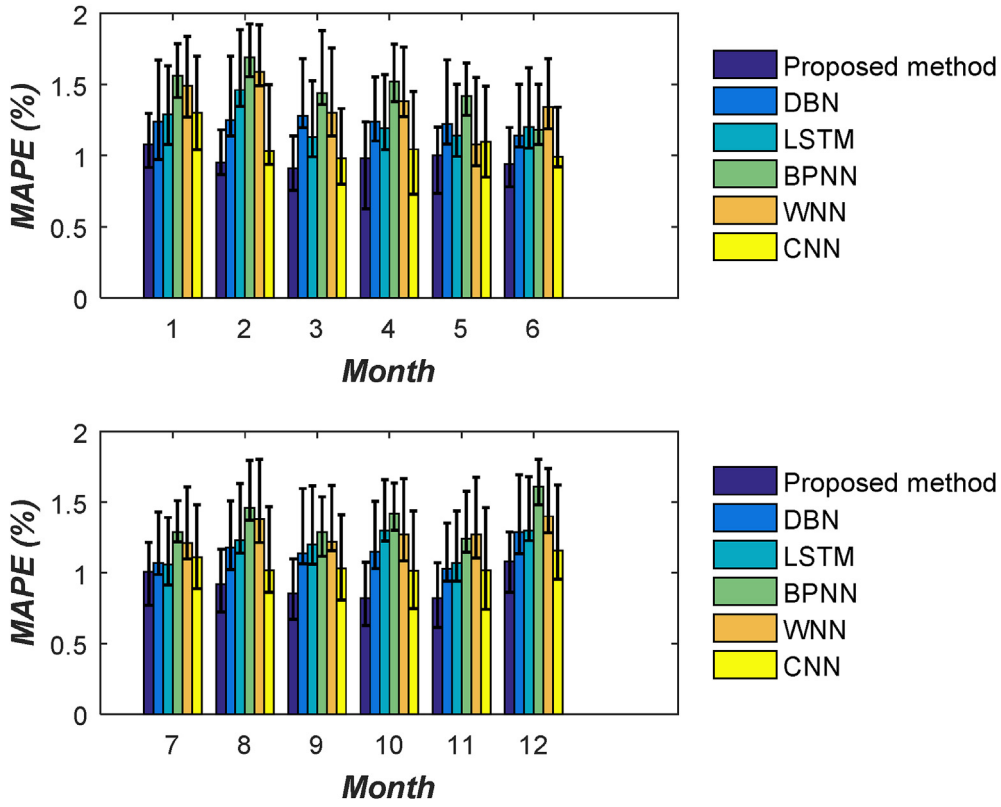

Fig. 25. 24h-ahead test results for month.

ensemble methods in this Case. It could be possibly attributed to better weights provided by IDA to HC, which plays a significant role in the convergence effect of HC [56].

Case V. The forecasting performances of different methods were tested in this case. The proposed method, DBN, LSTM, wavelet neural networks (WNN), back propagation neural network (BPNN) and CNN are compared. The data, from 2012 to 2016, is used as the training set, and the data in 2017 is tested. The program was repeatedly operated for 100 times, the 1-h and 24-h ahead forecasting results are shown in Figs. 24 and 25. The MAPE scopes (minimum, average and maximum) of the proposed method are all optimal in all months, compared with other tested models in this case.

The reasons why the proposed method can achieve good forecasting results can be summarized as follows:

Firstly, the redundant and excessive input information, which can undermine the forecasting performance [50,51], are reduced from the proposed input feature.

Secondly, HC can be first calculated start from the node (sub-model) with the smallest forecasting error without too many nodes (sub-models). This calculation strategy of HC promotes forecasting accuracy and reduces training and forecasting time.

Case VI. The forecasting performances of the proposed method, CNN, DBN and LSTM methods in weekends (Saturdays and Sundays) were tested in this case. The weekends, from 2010 to 2015, are used as the training set, and the weekends in 2016 are tested.

The program was repeatedly operated for 100 times, the 1-h and 24-h ahead forecasting results are shown in Figs. 26–29. As shown in Figs. 26 and 27, the scopes (minimum, average and maximum) of RMSE and MAE of the proposed method are all optimal in all tested methods, both for Saturdays and Sundays. As shown in Figs. 28 and 29, the scopes (minimum, average and maximum) of training time and predicting time of the proposed method are all optimal in all tested methods, both for Saturdays and Sundays. Apart from the calculation strategy of HC and removing the redundant and excessive input information from input features, another important reason is that the proposed method can generate more realistic simulation samples according to the input sample data (Section 2, Chapter 5). Sample simulation strategy makes up for the shortage of data samples on weekends in load forecasting. In other words, adequate training samples play a significant role in improving the forecasting performance.

Case VII. The forecasting performances of the proposed method, CNN, DBN and LSTM methods in holidays were tested in this case. The holidays, from 2005 to 2015, are used as the training set, and the holidays in 2016 are tested. The program was repeatedly operated for 100 times, and the test results are shown in Figs. 30 and 31.

As shown in Fig. 30, the scopes (minimum, average and maximum) of RMSE and MAE of the proposed method are all optimal in all tested methods. As shown in Fig. 31, the scopes (minimum, average and maximum) of training time and predicting time of the proposed method are all optimal in all tested methods.

As is known to all, lots of AI-based methods are difficult to be applied to holiday electricity consumption forecasting owing to the shortage of training samples [65].

Under the circumstance, this paper proposed a novel ensemble method, which can generate more realistic simulation samples according to the input sample data (Section 2, Chapter 5). It is such sample simulation strategy that enables adequate training samples to be used in holiday electricity consumption forecasting, which

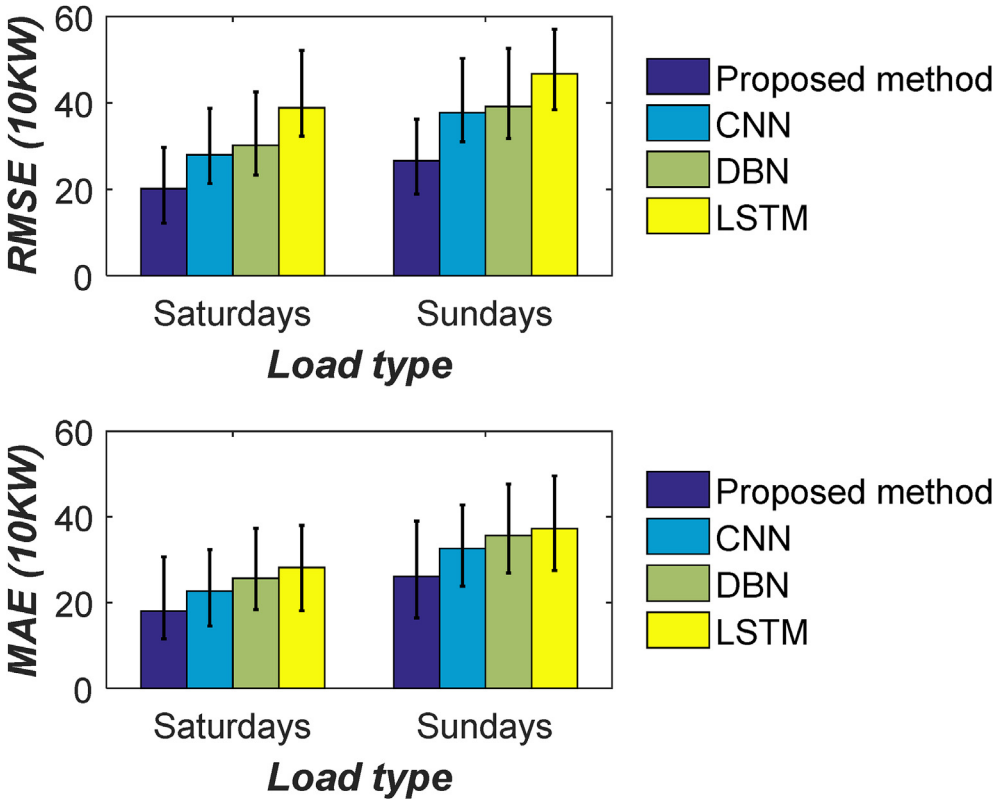

Fig. 26. 1h-ahead test results for weekends.

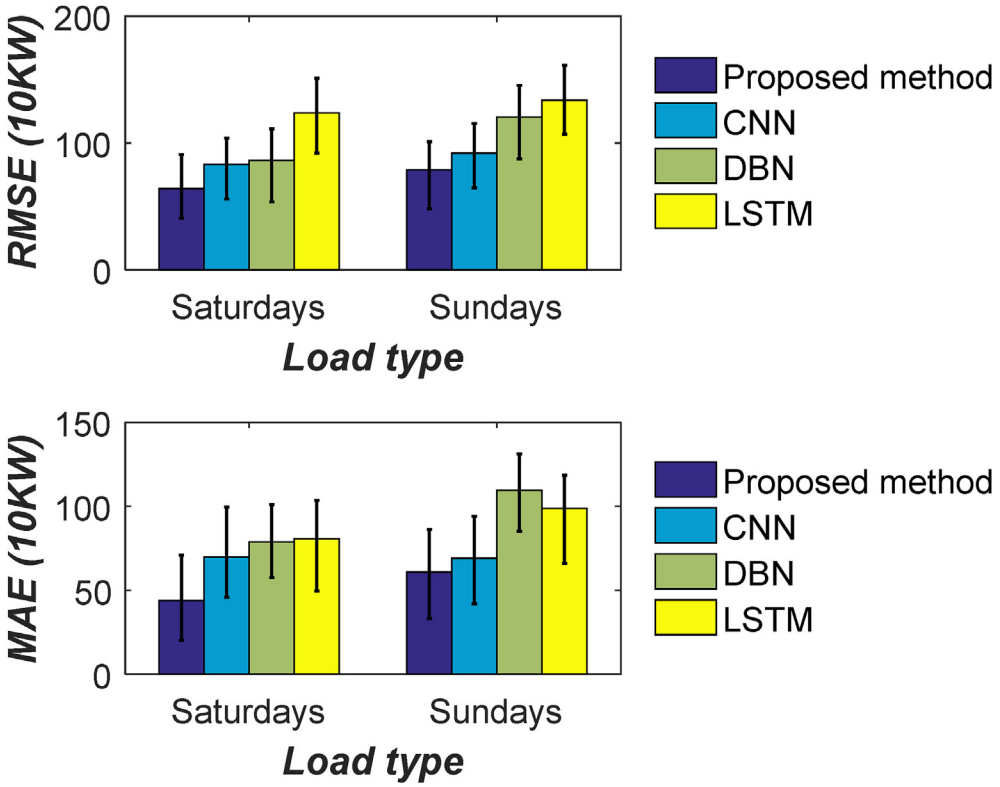

Fig. 27. 24h-ahead test results for weekends.

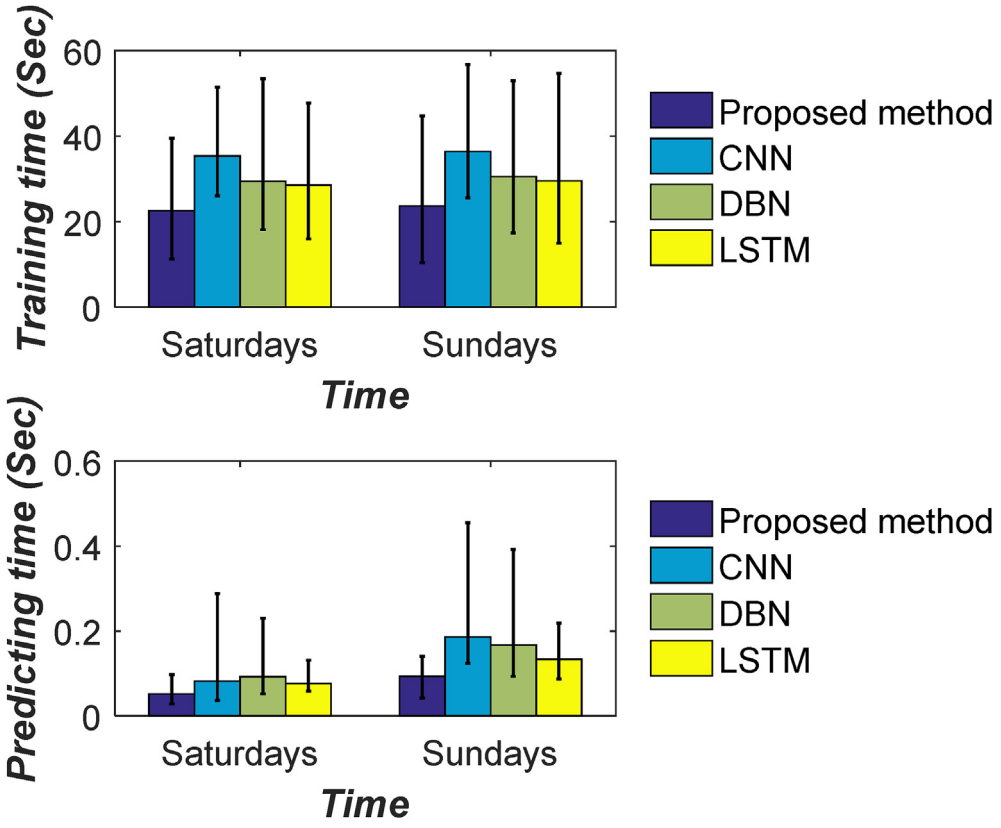

Fig. 28. 1-h ahead time comparison for weekends.

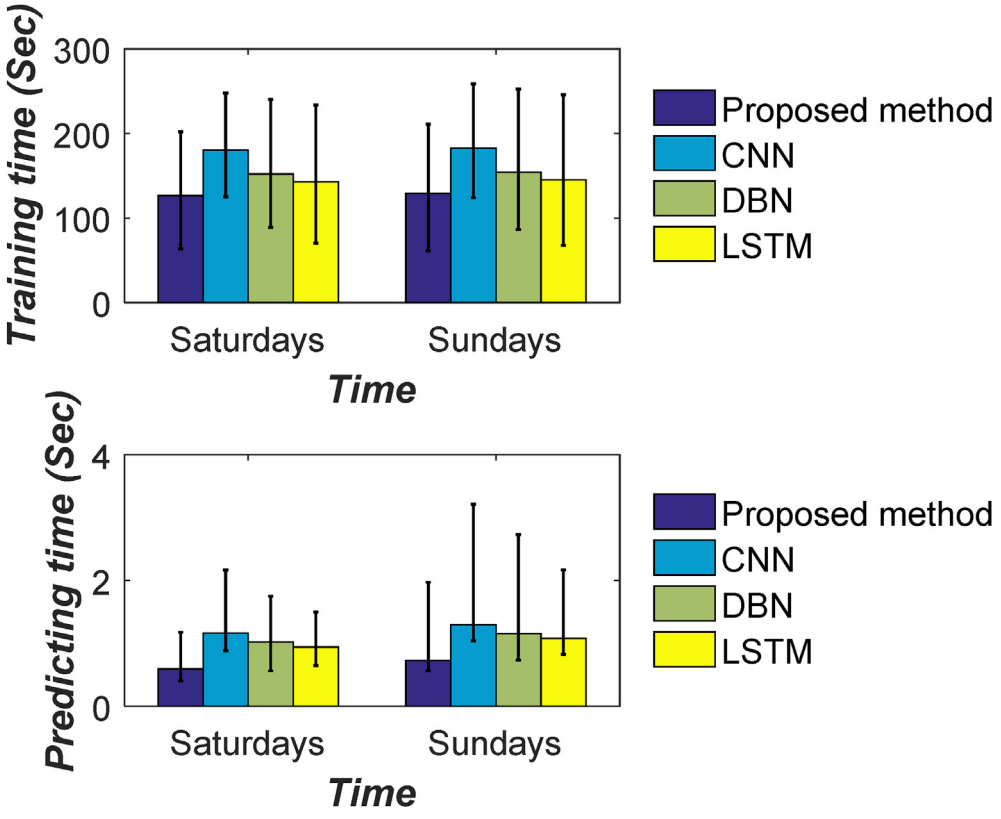

Fig. 29. 24-h ahead time comparison for weekends.

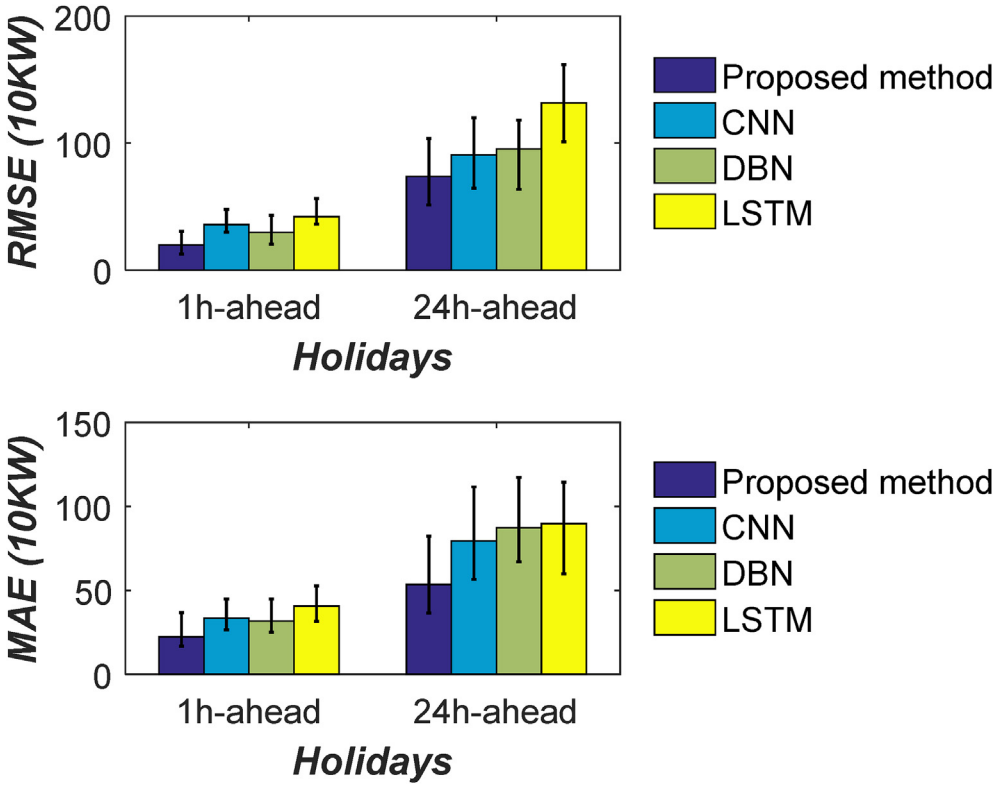

Fig. 30. Test results for holidays.

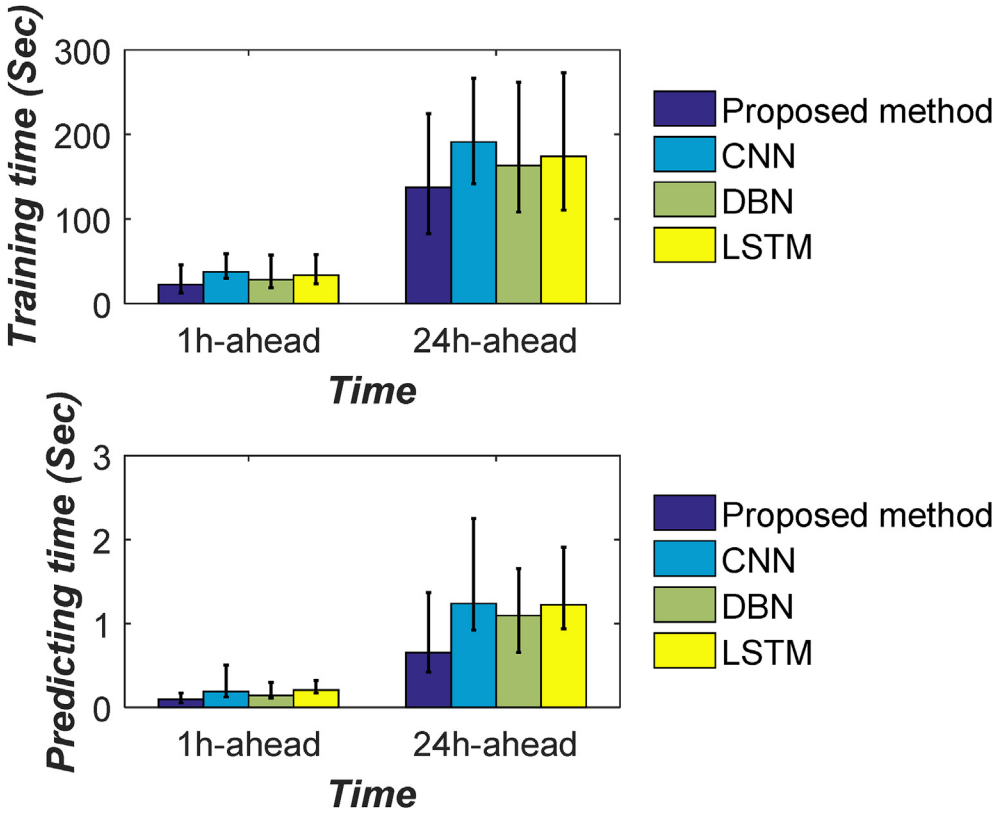

Fig. 31. Time comparison for holidays.

helps improve the forecasting performance. Besides, the calculation strategy of HC and removing the redundant and excessive input information are also helpful to improve the forecasting performance.

4. Conclusion

This paper proposes a novel ensemble method for the forecasting of hourly residential electricity consumption by encoding time series into images. The forecasting performance can be improved through removing the redundant and excessive information from input features by VMD and IKPCA. Then, CGAN is applied to generate more realistic simulation samples according to input sample data. The sample simulation strategy ensures sufficient training samples on weekends and holidays electricity consumption, which enhances the forecasting performance effectively. Next, all sub-CGANs are integrated by HC. The calculation strategy of HC improves forecasting accuracy while cutting the training and forecasting time. Finally, by optimizing the weights of HC via IDA, the performance of the ensemble model is effectively improved.

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.

References

[1] Bunn DW. Forecasting loads and prices in competitive power markets. Proc IEEE Feb. 2000;88(2):163–9. https://doi.org/10.1109/5.823996.
[2] Zachariadis T, Hadjinicolaou P. The effect of climate change on electricity needs -a case study from Mediterranean Europe. Energy Nov. 2014;76(1):899–910. https://doi.org/10.1016/j.energy.2014.09.001.
[3] Dadkhah M, Rezaee MJ, Chavoshib AZ. Short-term power output forecasting of hourly operation in power plant based on climate factors and effects of wind direction and wind speed. Energy Apr. 2018;148:775–88. https://doi.org/10.1016/j.energy.2018.01.163.
[4] Wang YP, Bielicki JM. Acclimation and the response of hourly electricity loads to meteorological variables. Energy Jan. 2018;142(1):473–85. https://doi.org/10.1016/j.energy.2017.10.037.
[5] Ahmed T, Vu DH, Muttaqi KM, Agalgaonkar AP. Load forecasting under changing climatic conditions for the city of Sydney, Australia. Energy Jan. 2018;142:911–9. https://doi.org/10.1016/j.energy.2017.10.070.
[6] Barman M, Dev Choudhury NB, Sutradhar S. A regional hybrid Goa-SVM model based on similar day approach for short-term load forecasting in Assam, India. Energy Feb. 2018;145:710–20. https://doi.org/10.1016/j.energy.2017.12.156.
[7] Zhang Y, Zhou Q, Sun CX, Lei SL, Liu YM, Song Y. RBF neural network and ANFIS-based short-term load forecasting approach in real-time price environment. IEEE Trans Power Syst Aug. 2008;23(3):853–8. https://doi.org/10.1109/TPWRS.2008.922249.
[8] Sorjamaa A, Hao J, Reyhani N, Ji YN, Lendasse A. Methodology for long-term prediction of time series. Appl Energy Oct. 2007;70(16–18):2861–9. https://doi.org/10.1016/j.neucom.2006.06.015.
[9] Song KB, Baek YS, Hong DH, Jang G. Short-term load forecasting for the holidays using fuzzy linear regression method. IEEE Trans Power Syst Feb. 2005;20(1):96–101. https://doi.org/10.1109/TPWRS.2004.835632.
[10] Amjady N. Short-term hourly load forecasting using time-series modeling with peak load estimation capability. IEEE Trans Power Syst Aug. 2001;16(3):498–505. https://doi.org/10.1109/59.932287.
[11] Pappas SS, Ekonomou L, Karamousantas DC, Chatzarakis GE, Katsikas SK, Liatsis P. Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models. Energy Sept. 2008;33(9):1353–60. https://doi.org/10.1016/j.energy.2008.05.008.
[12] Pappas SSp, Ekonomou L, Karampelas P, Karamousantas DC, Katsikas SK, Chatzarakis GE, Skafidas PD. Electricity demand load forecasting of the Hellenic power system using an ARMA model. Elec Power Syst Res Mar. 2010;80(3):256–64. https://doi.org/10.1016/j.epsr.2009.09.006.
[13] Li HZ, Guo S, Li CJ, Sun JQ. A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. Knowl Base Syst Jan. 2013;37:378–87. https://doi.org/10.1016/j.knosys.2012.08.015.
[14] Khoshrou A, Pauwels EJ. Short-term scenario-based probabilistic load forecasting: a data-driven approach. Appl Energy Mar. 2019;238:1258–68. https://doi.org/10.1016/j.apenergy.2019.01.155.
[15] Lauret P, Fock E, Randrianarivony RN, Manicom-Ramsamya JF. Bayesian neural network approach to short time load forecasting. Energy May. 2008;49(5):1156–66. https://doi.org/10.1016/j.enconman.2007.09.009.
[16] Hippert HS, Pedreira CE, Souza RC. Neural networks for short-term load forecasting: a review and evaluation. IEEE Trans Power Syst Feb. 2001;16(1):44–55. https://doi.org/10.1109/59.910780.
[17] Ferreira VH, Silva PA. Toward estimating autonomous neural network-based electric load forecasters. IEEE Trans Power Syst Nov. 2007;2(4):1554–62. https://doi.org/10.1109/TPWRS.2007.908438.
[18] Çevik HH, Çunkaş M. Short-term load forecasting using fuzzy logic and ANFIS. Neural Comput Appl Aug. 2015;26(6):1355–67. https://doi.org/10.1007/s00521-014-1809-4.
[19] Azadeh A, Saberi M, Seraj O. An integrated fuzzy regression algorithm for energy consumption estimation with non-stationary data: a case study of Iran. Energy June. 2010;35(6):2351–66. https://doi.org/10.1016/j.energy.2009.12.023.
[20] Lee WJ, Hong JY. A hybrid dynamic and fuzzy time series model for mid-term power load forecasting. Int J Electr Power Energy Syst Jan. 2015;64:1057–62. https://doi.org/10.1016/j.ijepes.2014.08.006.
[21] Chen BJ, Chang MW, Lin CJ. Load forecasting using support vector machines: a study on EUNITE competition 2001. IEEE Trans Power Syst Nov. 2004;19(4):1821–30. https://doi.org/10.1109/TPWRS.2004.835679.
[22] Nagi J, Yap KS, Nagi F, Tiong SK, Ahmed SK. A computational intelligence scheme for the prediction of the daily peak load. Appl Soft Comput Dec. 2011;11(8):4773–88. https://doi.org/10.1016/j.asoc.2011.07.005.
[23] Hinojosa VH, Hoese A. Short-term load forecasting using fuzzy inductive reasoning and evolutionary algorithms. IEEE Trans Power Syst Feb. 2010;25(1):565–74. https://doi.org/10.1109/TPWRS.2009.2036821.
[24] Hong WC. Electric load forecasting by support vector model. Appl Math Model May 2009;33:2444–54. https://doi.org/10.1016/j.apm.2008.07.010.
[25] Wang JZ, Zhu SL, Zhang WY, Lu HY. Combined modeling for electric load forecasting with adaptive particle swarm optimization. Energy Apr. 2010;35:1671–8. https://doi.org/10.1016/j.energy.2009.12.015.
[26] Kong WC, Dong ZY, Hill DJ, Luo FJ, Xu Y. Short-term residential load forecasting based on resident behaviour learning. IEEE Trans Power Syst Sept. 2017;33(1):1087–8. https://doi.org/10.1109/TPWRS.2017.2688178.
[27] A. Dedinec, S. Filiposka, A. Dedinec, and L. Kocarev, “Deep belief network based electricity load forecasting: an analysis of Macedonian case” Energy, vol. 115, pp. 1688-1700, Nov. 2016, doi: 10.1016/j.energy.2016.07.090.
[28] Fu GY. Deep belief network based ensemble approach for cooling load forecasting of air-conditioning system. Energy Apr. 2018;148:269–82. https://doi.org/10.1016/j.energy.2018.01.180.
[29] Xiao LY, Wang JZ, Hou R, Wu J. A combined model based on data pre-analysis and weight coefficients optimization for electrical load forecasting. Energy Mar. 2015;82:524–49. https://doi.org/10.1016/j.energy.2015.01.063.
[30] Wang L, Lv SX, Zeng YR. Effective sparse adaboost method with ESN and FOA for industrial electricity consumption forecasting in China. Energy Pol Apr. 2018;155:1013–31. https://doi.org/10.1016/j.energy.2018.04.175.
[31] Wu JR, Cui ZS, Chen YY, Kong DM, Wang YG. A new hybrid model to predict the electrical load in five states of Australia. Energy Oct. 2019;166:598–609. https://doi.org/10.1016/j.energy.2018.10.076.
[32] Jiang P, Liu F, Song YL. A hybrid forecasting model based on date-framework strategy and improved feature selection technology for short-term load forecasting. Energy Jan. 2017;119:694–709. https://doi.org/10.1016/j.energy.2016.11.034.
[33] Chen YB, Xu P, Chu YY, Li WL, Wu YT, Ni LZ, Bao Y, Wang K. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. Appl Energy June. 2017;195:659–70. https://doi.org/10.1016/j.apenergy.2017.03.034.
[34] Li S, Goel L, Wang P. An ensemble approach for short-term load forecasting by extreme learning machine. Appl Energy Feb. 2016;170:22–9. https://doi.org/10.1016/j.apenergy.2016.02.114.
[35] Takeda H, Tamura Y, Sato S. Using the ensemble Kalman filter for electricity load forecasting and analysis. Energy Apr. 2016;104:184–98. https://doi.org/10.1016/j.energy.2016.03.070.
[36] Taylor JW, Buizza R. Neural network load forecasting with weather ensemble predictions. IEEE Trans Power Syst Aug. 2002;17(3):626–32. https://doi.org/10.1109/TPWRS.2002.800906.
[37] Antipov G, Baccouche M, Dugelay JL. Face aging with conditional generative adversarial networks. In: 2017 IEEE international conference on image processing (ICIP); Feb. 2018. https://doi.org/10.1109/ICIP.2017.8296650.
[38] Douzas G, Baca F. Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Syst Appl Jan. 2018;91:464–71. https://doi.org/10.1016/j.eswa.2017.09.030.
[39] Zhao LJ, Bai HH, Liang J, Zeng B, Wang AH, Zhao Y. Simultaneous color-depth super-resolution with conditional generative adversarial networks. Pattern Recogn Apr. 2019;88:356–69. https://doi.org/10.1016/j.patcog.2018.11.028.
[40] Ding SH, Wallin A. Towards recovery of conditional vectors from conditional generative adversarial networks. Pattern Recogn Lett May. 2019;122:66–72. https://doi.org/10.1016/j.patrec.2019.02.020.
[41] Gu JX, Wang ZH, Kuen J, Ma LY, Shahroudy A, Shuai B, Liu T, Wang XX, Wang G, Cai JF, Chen T. Recent advances in convolutional neural networks. Pattern Recogn May. 2018;77:354–77. https://doi.org/10.1016/j.patcog.2017.10.013.
[42] Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. Reading text in the wild

with convolutional neural networks. Int J Comput Vis Mar. 2015;116(1):1–20. https://doi.org/10.1007/s11263-015-0823-z.
[43] Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, Liang JM. Convolutional neural networks for medical image analysis: full training or fine tuning. IEEE Trans Med Imag May. 2016;35(5):1299–312. https://doi.org/10.1109/tmi.2016.2535302.
[44] Chen Y, Luh PB, Guan C, Zhao YG, Michel LD, Coolbeth MA, Friedland PB, Rourke SJ. Short-term load forecasting: similar day-based wavelet neural networks. IEEE Trans Power Syst Feb. 2010;25(1):322–30. https://doi.org/10.1109/TPWRS.2009.2030426.
[45] Li S, Wang P, Goel L. A novel wavelet-based ensemble method for short-term load forecasting with hybrid neural networks and feature selection. IEEE Trans Power Syst May. 2016;31(3):1788–98. https://doi.org/10.1109/TPWRS.2015.2438322.
[46] Liang Y, Niu DX, Hong WC. Short term load forecasting based on feature extraction and improved general regression neural network model. Energy Jan. 2019;166:653–63. https://doi.org/10.1016/j.energy.2018.10.119.
[47] Kim SH, Lee G, Kwon GY, Kim DI, Shin YJ. Deep learning based on multi-decomposition for short-term load forecasting. Energies Dec. 2018;11(12):1–17. https://doi.org/10.3390/en11123433.
[48] Chen Y, Luh PB, Guan C, Zhao YG, Michel LD, Coolbeth MA, Friedland PB, Rourke SJ. Short-term load forecasting: similar day-based wavelet neural networks. IEEE Trans Power Syst Feb. 2010;25(1):322–30. https://doi.org/10.1109/TPWRS.2009.2030426.
[49] Bento PMR, Pombo JAN, Calado MRA, Mariano SJPS. A bat optimized neural network and wavelet transform approach for short term price forecasting. Appl Energy Jan. 2018;210:88–97. https://doi.org/10.1016/j.apenergy.2017.10.058.
[50] Amjady N, Keynia F. Day-ahead price forecasting of electricity markets by mutual information technique and cascaded neuro evolutionary algorithm. IEEE Trans Power Syst Feb. 2009;24(1):306–18. https://doi.org/10.1109/TPWRS.2008.2006997.
[51] Massana J, Pous C, Burgas L, Melendez J, Colomer J. Identifying services for short-term load forecasting using data driven models in a Smart City platform. Sustainable Cities and Society Sept. 2017;28:108–17. https://doi.org/10.1016/j.scs.2016.09.001.
[52] Dragomiretskiy K, Zosso D. Variational mode decomposition. IEEE Trans Signal Process Feb. 2014;62(3):531–44. https://doi.org/10.1109/tsp.2013.2288675.
[53] Chin TJ, Suter D. Incremental kernel principal component analysis. IEEE Trans Image Process June. 2007;16(6):1662–74. https://doi.org/10.1109/TIP.2007.896668.
[54] Yang CL, Yang CY, Chen ZX, Lo NW. Multivariate time series data transformation for convolutional neural network. In: 2019 IEEE/SICE international symposium on system integration (SII); Apr. 2019. https://doi.org/10.1109/SII.2019.8700425.
[55] Qin Z, Zhang YB, Meng SY, Qin ZG, Choo KR. Imaging and fusing time series for wearable sensor-based human activity recognition. Inf Fusion Jan. 2020;53:80–7. https://doi.org/10.1016/j.inffus.2019.06.014.
[56] Knuth DE. Dynamic huffman coding. J Algorithm Jun. 1985;6(2):163–80. https://doi.org/10.1016/0196-6774(85)90036-7.
[57] Moffat A. Huffman coding. ACM Comput Surv Sept. 2019;52(4):1–35. https://doi.org/10.1145/3342555.
[58] Cochran JK, Horng SM, Fowler JW. A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. Comput Oper Res June. 2003;30(7):1087–102. https://doi.org/10.1016/S0305-0548(02)00059-X.
[59] Liang Y, Niu DX, Hong WC. Short term load forecasting based on feature extraction and improved general regression neural network model. Energy Jan. 2019;166:653–63. https://doi.org/10.1016/j.energy.2018.10.119.
[60] Fay D, Ringwood JV. On the influence of weather forecast errors in short-term load forecasting models. IEEE Trans Power Syst Aug. 2010;25(3):1751–8. https://doi.org/10.1109/TPWRS.2009.2038704.
[61] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl May. 2016;27(4):1053–73. https://doi.org/10.1007/s00521-015-1920-1.
[62] Yang XS. Firefly algorithm, lévy flights and global optimization. In: Research and development in intelligent systems XXVI; Oct. 2009. p. 209–18. https://doi.org/10.1007/978-1-84882-983-1_15.
[63] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Software Mar. 2014;69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.
[64] Fidalgo JN, Lopes JAP. Load forecasting performance enhancement when facing anomalous events. IEEE Trans Power Syst Feb. 2005;20(1):408–15. https://doi.org/10.1109/TPWRS.2004.840439.
[65] Kim KH, Youn HS, Kang YC. Short-term load forecasting for special days in anomalous load conditions using neural networks and fuzzy inference method. IEEE Trans Power Syst May. 2000;15(2):559–65. https://doi.org/10.1109/59.867141.