IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 20, 2020, accepted October 4, 2020, date of publication October 6, 2020, date of current version October 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3029224

Short-Term Load Forecasting Using Recurrent Neural Networks With Input Attention Mechanism and Hidden Connection Mechanism

MINGFEI ZHANG, (Graduate Student Member, IEEE), ZHOUTAO YU, (Graduate Student Member, IEEE), AND ZHENGHUA XU, (Member, IEEE)

State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300130, China
Key Laboratory of Electromagnetic Field and Electrical Apparatus Reliability of Hebei Province, Hebei University of Technology, Tianjin 300130, China

Corresponding author: Zhenghua Xu (zhenghua.xu@hebut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61906063, in part by the Natural Science Foundation of Tianjin City, China, under Grant 19JCQNJC00400, in part by the 100 Talents Plan of Hebei Province, China, under Grant E2019050017, and in part by the Yuanguang Scholar Fund of the Hebei University of Technology, China.

ABSTRACT Short-term load forecasting is a critical task in the smart grid, which can be used to optimize power deployment and reduce power losses. Recurrent neural networks (RNNs) are the most popular deep learning models for short-term load forecasting. However, despite of achieving better forecasting accuracy than the traditional models, the performance of the existing RNN-based load forecasting approaches is still unsatisfactory for practical usage. Therefore, in this work, we have proposed input attention mechanism (IAM) and hidden connection mechanism (HCM) to greatly enhance the accuracy and efficiency of RNN-based load forecasting models. Specifically, we use IAM to assign the importance weights on input layers, which have better performances in both efficiency and accuracy than traditional attention mechanisms. To further enhance the models' efficiency, HCM is then applied to utilize residual connections to enhance the model's converging speed. We have applied both IAM and HCM on four state-of-the-art RNN implementations, and then conducted extensive experimental studies on two public datasets. Experimental results show that the proposed RNNs with IAM and HCM models achieve much better performances than the state-of-the-art baselines in both accuracy and efficiency. Ablation studies show that both IAM and HCM are essential to achieve such superior performances.

INDEX TERMS Short-term load forecasting, recurrent neural networks, input attention mechanism, hidden connection mechanism.

I. INTRODUCTION

The goal of Short-term load forecasting (STLF) is to forecast the load values in the next few hours or days based on historical load values that sometimes consider the weather, temperature and other factors. Accurate and efficient STLF can strike a balance between supply and demand, avoiding the waste of resources and improving the stability of the power system. With the application of distributed generations and smart meters in smart grid, a new challenge on the accuracy and efficiency of STLF is imposed [1], [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Giambattista Gruosso.

Recurrent Neural Networks are an effective method for short-term load forecasting in recent years [3]. Theoretically, the recurrent neural networks can take the temporal correlation of the time series and use historical information of any length. However, the inherent shortcoming of gradient explosion and gradient vanishing during training in traditional recurrent neural networks is a restraint. In response to the deficiency of traditional recurrent neural network, long short-term memory network (LSTM) is proposed to overcome the disadvantage by the gate mechanisms [4]. The calculation method of the hidden layer can be changed through these mechanisms, thereby effectively combining short-term memory with long-term memory. In [5], LSTM is applied

to STLF, which proves its great superiority compared to the traditional models. Jiao *et al.* further expands the LSTM and successfully applies it to multi-sequence STLF [6]. Nevertheless, complex structure and numerous learn-able parameters of LSTM make it difficult for the forecasting model to converge [7]. Therefore, gated recurrent unit network (GRU) is proposed to optimize the LSTM by combining the input gate and the forget gate into a single update gate [8]. Wang *et al.* proposed a GRU-based load forecasting model for electricity generation's planning and operation, and results show the performance of GRU is superior to LSTM [9]. At present, LSTM and GRU have been widely used as basic RNN-based models in deep learning methods for STLF [10]. Although RNN-based load forecasting models have achieved better performance than historical models, the accuracy is still not satisfactory due to the poor performance on forecasting peak and valley load values.

Therefore, many efforts have been made on enhancing the accuracy of the basic RNN-based models. Bidirectional RNN-based models (Bi-RNN-based models) are proposed to achieve higher accuracy by taking the historical and future information into consideration [11]. Liu *et al.* successfully applies the bidirectional LSTM (Bi-LSTM) to STLF, and concludes that the prediction results of Bi-LSTM are more accurate [12]. Tang *et al.* proposed bidirectional GRU (Bi-GRU) to further verify that the Bi-RNN-based models are better than RNN-based models in terms of accuracy, but the efficiency is reduced, that is, the convergent time is greatly increased [13]. Currently, Recurrent neural networks (RNNs), including RNN-based models and Bi-RNN-based models, have become the most widely used models in STLF because they can effectively deal with the time series problems. However, the training time of Bi-RNN-based models is almost doubled due to the doubling of the learn-able parameters.

Two mechanisms, Convolution Neural Network (CNN) and attention mechanism (AM), have been proposed to modify RNNs to further improve the performance of the forecasting models. CNN has achieved excellent performance in load forecasting due to its ability to effectively extract features. LSTM with CNN and GRU with CNN have been proposed to enhance the forecasting accuracy by using CNN to extract the features of the load values, respectively [14], [15]. Nevertheless, the training of RNNs with CNN is too slow owing to a large number of learn-able parameters in convolution operation. In recent years, the attention mechanism (AM) is widely used to modify the RNNs because AM allows the forecasting models to pay more attention to effective features. In [16], a hybrid short-term framework with an attention-based encoder-decoder network is proposed by encoding the input sequence as a fixed-length vector and using the decoder to generate a target sequence. Ju *et al.* proposed RNN-based models with attention mechanism to address the issue of unsatisfying accuracy in STLF and the results show that the attention mechanism can achieve better performance than RNN-based models [17]. Attention mechanism is further applied to Bi-RNN-based models in [18], indicating that attention mechanism has wonderful generalization ability.

Although the load forecasting accuracies of RNN-based models have been improved by using the existing CNN and AM solutions, both CNN and AM introduce a large number of additional learn-able parameters into the RNN-based deep models, making them unsatisfactory for practical usage. Specifically, their shortcomings are as follows: i) the additional parameters dramatically decrease the RNN-based models' learning efficiency, and ii) due to additional parameters, the RNN-based models become more complex, making them more difficult to learn, which thus also limits their improvements in the forecasting accuracy. Since efficiency and accuracy are both very important in short-term load forecasting, which directly affect the safety and economy of power systems [19], it is compelling to improve the accuracy and efficiency of the existing RNN-based load forecasting models to make the short-term load forecasting using deep models become more practical.

To address these issues, we proposed recurrent neural networks with the proposed mechanism called input attention mechanism (IAM) and the hidden connection mechanism (HCM). First, IAM is proposed to fine-tune the structure of AM, which assigns the importance weights on input layers instead of using encoder-decoder structure. Through this mechanism, learn-able parameters are reduced while improving the accuracy and efficiency compared to traditional AM. However, using IAM alone cannot achieve the desired efficiency. Furthermore, HCM applies residual connection mechanism on hidden layers to speed up convergence in order to decrease training time. We have observed that the efficiency is greatly improved, and the accuracy is also slightly improved by HCM. Consequently, the accuracy of load prediction has been improved, and the convergent time of forecasting models has been greatly decreased through the combination of IAM and HCM.

- Input Attention Mechanism is proposed to assign the importance weights on input layers instead of hidden layer, of which the parameters are greatly reduced. In the meanwhile, the accuracy does not decrease.
- Hidden Connection Mechanism is proposed to apply residual connection mechanism on hidden layers. The convergence has been accelerated during training, thereby improving the efficiency of forecasting models.
- We have applied both IAM and HCM on four state-of-the-art RNN implementations, and then conducted extensive experimental studies on two public datasets. Experimental results show that the proposed RNNs with IAM and HCM models achieve much better performances than the state-of-the-art baselines in both accuracy and efficiency.
- Ablation studies are also conducted to show that both IAM and HCM are essential to achieve such superior performances. Moreover, some additional investigations are further conducted, which discover two possible

reasons for the superior performances of the proposed RNNs with IAM and HCM models: i) the proposed models are not only more accurate but also more stable than the state-of-the-art baselines; ii) The proposed models can achieve generally much better performances than the state-of-the-art baselines in forecasting the peak and valley load values.

The rest of paper is organized as follows: Section II contains the literature review. Section III contains the methodology. Section IV provides the detailed process and configuration of the experiments, analyzes the results of the experiments, and gives an explanation to the superiority of proposed mechanism. Section V contains conclusions and future work.

II. LITERATURE REVIEW

Deep neural networks, especially recurrent neural networks have recently shown their excellent capability for load forecasting. Zheng *et al.* proposed an LSTM model for short-term forecasting to process complex uni-variate power load data. It indicates that forecasting models based on LSTM outperform other methods, including auto-regressive integrated moving average (ARIMA), support vector regression (SVR), and traditional feed-forward neural network (FNN) [20]. Kumar *et al.* used the LSTM model for short-term electric load forecasting using twelve years of historical data for ISO New England electricity market and reported very reliable and robust results [21]. Zheng *et al.* proposed a short-term load forecasting method for residential community based on gated recurrent unit neural network. The simulation results show that the GRU is faster within the similar forecasting accuracy, compared with the LSTM network [22]. However, the RNN-based models can not achieve the satisfactory result due to the poor performance in peak values and valley values. Therefore, in this work, we propose IAM and HCM to utilize attention mechanism and residual connections to overcome these problems and further enhance the forecasting performances.

Recently, bidirectional recurrent neural networks have gained popularity due to the ability to take both historical and future information into consideration. Liu *et al.* utilized the combination of wavelet decomposition, radial basis function, and bidirectional LSTM to predict electric energy consumption. The experimental results indicate that bidirectional LSTM framework outperforms the unidirectional approaches in terms of several performance metrics for electric prediction [12]. Deng *et al.* proposed a deep learning framework based on bidirectional gated recurrent unit for wind power prediction to improve the accuracy by making full use of the information provided by multiple data sources of numerical weather forecast. Results show bidirectional model helps to further improve prediction accuracy [23]. Atef *et al.* conduct a systematic experimental methodology to investigate the impact of using deep-stacked unidirectional and bidirectional networks on predicting electricity load consumption and draw a conclusion that bidirectional models have significant improvement in the prediction accuracy while they consume almost twice the time of the unidirectional models [24]. However, due to the doubling of the learn-able parameters, the training time of Bi-RNN-based models is also doubled, that is, the efficiency is greatly reduced. Therefore, in this work, residual connection is utilized by HCM to achieve more efficient model learning capability.

Several published works have successfully used convolutional neural network as the feature extractor for load forecasting. Le *et al.* proposed an electric energy consumption prediction model utilizing the combination of CNN and Bi-LSTM to predict electric energy consumption. Results show that the prediction performance of Bi-LSTM has been significantly improved after extracting features through CNN [25]. Kim *et al.* proposed a CNN-LSTM neural network that can extract spatial and temporal features to effectively predict the housing energy consumption and found that LSTM with CNN achieves better prediction performance than LSTM [14]. Sajjad *et al.* achieved the mentioned tasks by developing a hybrid sequential learning-based energy forecasting model that employs CNN and GRU into a unified framework for accurate energy consumption prediction. Results show that features are extracted by CNN from input dataset and fed into GRU, which is selected as optimal and observed to have enhanced sequence learning abilities after extensive experiments [15]. However, the efficiency is also greatly reduced with complex convolution operation in CNN module. Comparing to these works, in this work, HCM is introduced to ensure the deep models can be learned in an efficient way.

Several works utilizing attention mechanism for improving the accuracy of RNNs in the load forecasting domain are reported in the literature. Wilms *et al.* used an origin attention mechanism called sequence-to-sequence recurrent neural networks to combine elements of auto-regressive forecasting techniques with multivariate regression for each forecasting time step as well as previous values when inferring forecasts. Results show that AM can effectively improve the accuracy of load forecasting [26]. Li *et al.* established a wind power forecasting model based on LSTM network two-stage attention mechanism and found that the attention mechanism is extensively employed to weight the input feature and strengthen the trend characteristic [27]. Wang *et al.* proposed a novel short-term load forecasting method based on attention mechanism, rolling update, and bidirectional long short-term memory neural network. The experimental results prove that the attention mechanism has a good generalization in the bidirectional models [28]. However, the typical encoder-decoder structure greatly increases an amount of parameters, which affects the performance of forecasting results in both accuracy and efficiency. Differently, in this work, IAM achieves attention mechanism using much less additional parameters, and the HCM is additionally applied to greatly enhance the learning efficiency; so RNNs with IAM and HCM can achieve much better accuracy and efficiency than those with the existing attention mechanisms.

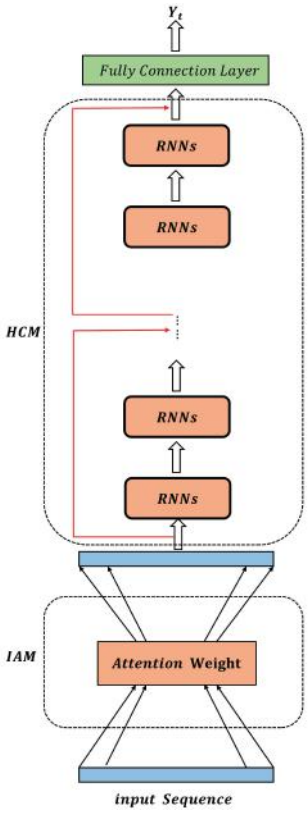


FIGURE 1. The overall architecture of RNNs with IAM and HCM.

III. METHODOLOGY

Figure 1 introduces the overall architecture of the proposed model at time step t, using a combination of input attention mechanism (IAM) and hidden connection mechanism (HCM) with recurrent neural networks (RNNs) to predict the load value of the next time step. At time t, the input sequence X of the dataset is assigned attention weights by the first module IAM, and the weighted input sequence $\tilde{X}$ is passed to the RNNs layers in the second module. RNNs layer is used for information analysis and time series prediction, of which HCM is used to speed up convergence in order to decrease training time. Finally, a fully connected layer is used to generate the predicted load value $\hat{y}$ of the next time step.

Through the feed-forward process, the effective features are extracted from the input sequence, and finally the predicted load value is obtained. Then the error between the predicted load value and the true load value y can be calculated by loss function:

$$J(W) = \frac{1}{2}||\hat{y}-y||^2 + \frac{\lambda}{2}||W||^2, \tag{1}$$

where the first term is $l2$ loss function, the second term is weight decay, W represents the learn-able parameters of RNNs with IAM and HCM.

In the back-forward process, the learn-able parameters of each RNNs layer are iteration updated by the optimization algorithms with the error back propagation. Adaptive moment estimation (Adam) [29] is used to update W and the algorithms can be expressed as follows:

$$\begin{aligned} m_t &= \beta_1 \cdot m_{t-1} + (1-\beta_1)\cdot g_t, \\ v_t &= \beta_2 \cdot v_{t-1} + (1-\beta_2)\cdot g_t^2, \\ \hat{m}_t &= \frac{m_t}{1-\beta_1^t}, \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t}, \\ W_t &= W_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{v_t}+\epsilon}, \end{aligned} \tag{2}$$

where m_t and v_t are the biased first moment estimate and the biased second raw moment estimate at timestep t, respectively, g_t is the gradients of W, g_t^2 indicates the element-wise square $g_t \odot g_t$, β_1 and β_2 are exponential decay rates for moment estimates, η is the learning rate.

The feed-forward and the back-forward process alternate until the error reaches a certain standard, and the training of the forecasting model is completed.

The rest of this section is organized as follows, we briefly introduce original RNNs, and its common variants, LSTM and bidirectional LSTM, GRU and bidirectional LSTM. In subsection III-B, we explain the principle and structure of input attention mechanism. Subsection III-C gives the details of hidden connection mechanism.

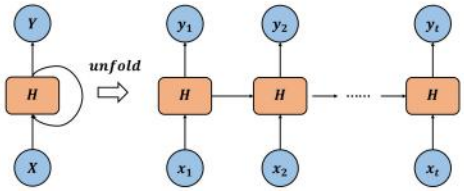


FIGURE 2. The general structure of recurrent neural networks.

A. RECURRENT NEURAL NETWORKS

Recurrent Neural Networks (RNNs) is theoretically capable of processing time series with arbitrary length by using self-feedback neurons. The general structure of RNNs is shown in Figure 2, given an input sequence $X = (x_1, x_2, \ldots, x_t) \in \mathbb{R}^{N\times T}$, the RNN computes the hidden state sequence $H = (h_1, h_2, \ldots, h_t) \in \mathbb{R}^{H\times T}$ as well as the output sequence $Y = (y_1, y_2, \ldots, y_t) \in \mathbb{R}^{M\times T}$ iteratively by followings equations:

$$\begin{aligned} h_t &= f(x_t W_{xh} + h_{t-1} W_{hh} + b_h), \\ y_t &= f(h_t W_{hy} + b_y), \end{aligned} \tag{3}$$

where $x_t \in \mathbb{R}^N$ denotes input sequence at current time step t, $h_t \in \mathbb{R}^H$ denotes hidden states at time step t, $y_t \in \mathbb{R}^M$ denotes the output sequence at time step t, f is the activation function, $W_{xh} \in \mathbb{R}^{N\times H}$, $W_{hh} \in \mathbb{R}^{H\times H}$, $W_{hy} \in \mathbb{R}^{H\times M}$ denote the weight matrices of the RNNs, $b_h \in \mathbb{R}^M$, $b_y \in \mathbb{R}^M$ denote the bias matrices of the RNNs.

Internal issues of gradient explosion and gradient vanishing limit the performance of traditional RNNs. Therefore, efforts have been devoted to improving the structure of traditional RNNs to overcome their natural shortcomings. Among them, long short-term memory network (LSTM) and gated recurrent unit network (GRU) are the most successful RNN-based models. Accordingly, bidirectional LSTM and bidirectional GRU are the most successful Bi-RNN-based models.

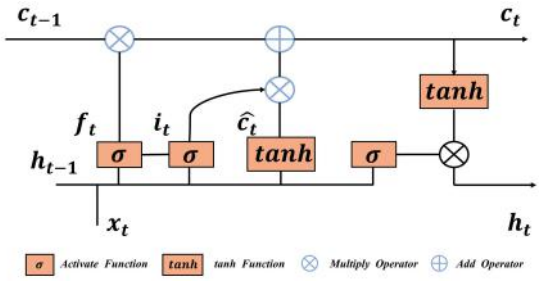


FIGURE 3. The structure of long short-term memory network.

1) LSTM AND BIDIRECTIONAL LSTM

Compared with the traditional RNNs with only one state in the hidden layer, the concept of cell state is introduced by LSTM to consider the time correlation hidden in the long-term state. Figure 3 shows the structure of the LSTM unit. At time step t, the LSTM takes three inputs: $x_t \in \mathbb{R}^N$ is the input sequence at current time step, $h_{t-1} \in \mathbb{R}^H$ and $c_{t-1} \in \mathbb{R}^H$ are the output and cell state from the previous LSTM unit. As for output, $h_t \in \mathbb{R}^H$ and $c_t \in \mathbb{R}^H$ are the output and cell state of the current LSTM unit. Three gates are introduced to control the cell state, namely input gate, forget gate and output gate. The formulation of LSTM are given by followings:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i), \tag{4}$$
$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f), \tag{5}$$
$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o), \tag{6}$$
$$\widehat{c_t} = tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c), \tag{7}$$
$$c_t = f_t \odot c_{t-1} + i_t * \widehat{c_t}, \tag{8}$$
$$h_t = o_t \cdot tanh(c_t), \tag{9}$$

where the input gate i_t decides what information to add from the current input to the cell state, the forget gate f_t decides what must be removed from the h_{t-1} state, thus keeping only relevant information, and the output gate o_t decides what information to output from the current cell state. In Equations (5)-(8), the notation σ is a sigmoid function and $[W_{ix}, W_{ih}, b_i]$, $[W_{fx}, W_{fh}, b_f]$, $[W_{ox}, W_{oh}, b_o]$, and $[W_{cx}, W_{ch}, b_c]$ are learn-able parameters for the input, forget, output, and cell modulation gates respectively. In Equations (9)-(9), the notation $\odot$ is an element-wise product operator.

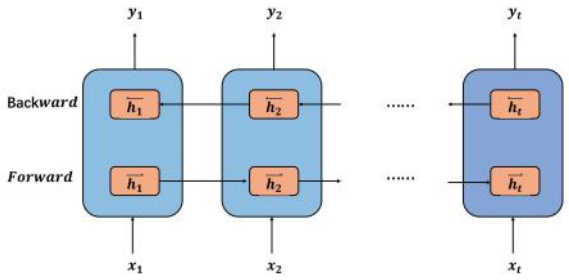


FIGURE 4. The structure of bidirectional LSTM and GRU.

However, the traditional LSTM can just use the historical information but cannot use future information. So the bidirectional LSTM (Bi-LSTM) is proposed to take both historical and future information into consideration. As shown in Figure 4, the historical information is obtained by forward LSTM and the future information is received by backward LSTM, the forward $\overrightarrow{h_t}$ and the backward $\overleftarrow{h_t}$ are calculated as follows:

$$\begin{aligned} \overrightarrow{h_t} &= \overrightarrow{LSTM}\ (h_{t-1}, x_t), \quad t \in [1, T], \\ \overleftarrow{h_t} &= \overleftarrow{LSTM}\ (h_{t-1}, x_t), \quad t \in [T, 1]. \end{aligned} \tag{10}$$

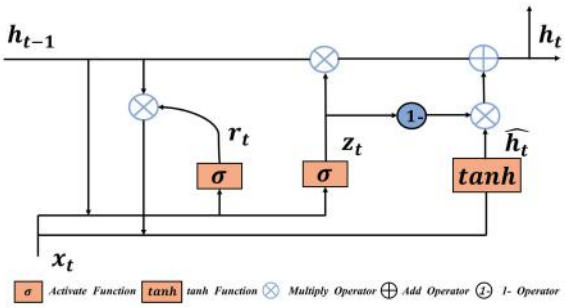


FIGURE 5. The structure of gate recurrent unit network.

2) GRU AND BIDIRECTIONAL GRU

Gated Recurrent Unit Network (GRU) is a special type of recurrent neural network based on optimized LSTM. Figure 5 shows the structure of the GRU. It can be noticed that the internal unit of GRU is similar to the internal unit of the LSTM, except that the GRU combines the input gate and the forget gate in the LSTM into a single update gate. Therefore, two gates are left in GRU, one is the update gate, which preserves previous information to the current state, the other is the reset gate, which is used to determine whether the current status and previous information are to be combined. The formulation of GRU can be expressed by the following equations:

$$z_t = \sigma(W_{zx}x_t + W_{zh}h_{t-1} + b_z), \tag{11}$$
$$r_t = \sigma(W_{rx}x_t + W_{rh}h_{t-1} + b_r), \tag{12}$$
$$\tilde{h_t} = tanh(W_{hx}x_t + W_h(r_t \odot h_{t-1}) + b_h), \tag{13}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h_t \tag{14}$$

where z_t is the the update gate, and r_t is the reset gate. $x_t \in \mathbb{R}^N$ denotes the input sequence at the current time step, $h_t, h_{t-1} \in \mathbb{R}^H$ denotes the hidden state at the current time step and previous time step, respectively. In Equations (12)-(14), the notation σ is a sigmoid function and $[W_{zx}, W_{zh}, b_z]$, $[W_{rx}, W_{rh}, b_h]$, and $[W_{hx}, W_h, b_h]$ are learn-able parameters for the update, reset, and hidden modulation gates respectively. In Equation (14), the notation $\odot$ is an element-wise product operator.

Bidirectional GRU (Bi-GRU) is also proposed to consider both historical and future information. Similar to LSTM, the historical information is obtained by forward GRU and the future information is received by backward GRU, the forward $\overrightarrow{h_t}$ and the backward $\overleftarrow{h_t}$ are calculated as follows:

$$\begin{aligned} \overrightarrow{h_t} &= \overrightarrow{GRU}\ (h_{t-1}, x_t), \quad t \in [1, T], \\ \overleftarrow{h_t} &= \overleftarrow{GRU}\ (h_{t-1}, x_t), \quad t \in [T, 1]. \end{aligned} \tag{15}$$

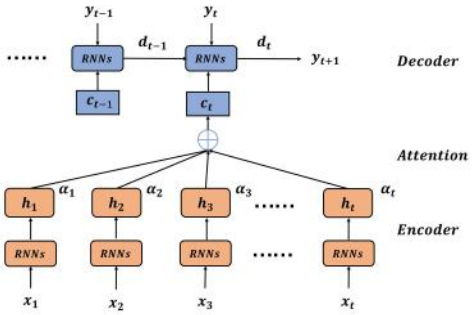


FIGURE 6. The architecture of RNNs with typical attention mechanism.

B. INPUT ATTENTION MECHANISM

The attention mechanism is originated from the signal processing mechanism of the brain. Important information is quickly selected by the attention mechanism, which improves the efficiency and accuracy of visual information processing. The core idea of the attention mechanism in deep learning is also to ignore the irrelevant information and only to select the information that is more critical to the current mission [30]. AM allows the forecasting models to pay more attention to effective features, so attention mechanism is widely used in RNNs for load forecasting. Figure 6 shows the architecture of RNNs with AM for STLF. Given a sequence of input $x_t \in \mathbb{R}^N$, the RNNs with AM learns to calculate $y_t \in \mathbb{R}^M$ by three modules: the encoder, the attention, the decoder. The encoder maps each input x_t to the hidden state $h_t \in \mathbb{R}^H$ by neural networks. The attention weights α_t can be obtained by the softmax of h_t, then the context vector c_t provide summaries of the input sequence by linearly combining the hidden states through a set of attention weights by following equations:

$$\begin{aligned} h_t &= RNNs(x_t), \\ \alpha_t &= \frac{exp(h_t)}{\sum_{t=1}^{T} exp(h_t)}, \end{aligned}$$

$$c_t = \sum_{t=1}^{T} \alpha_t h_t. \tag{16}$$

In the decoder, the weighted summed context vector $c_t \in \mathbb{R}^H$ is combined with the output y_{t-1} of the decoder at the last time step. Then, the hidden state of the decoder is updated with $d_t = f_d(d_{t-1}, y_{t-1}, c_t)$, where f_d represents the neural network used in the decoder. Eventually, the prediction can be made as below:

$$y_t = W_{hy}(W_{hc}[d_t, c_t] + b_{hc}) + b_{hy}, \tag{17}$$

where $W_{hc} \in \mathbb{R}^{(C+H)\times H}$, and $b_{hc} \in \mathbb{R}^H$ map the concatenation to the size of the decoder hidden states. $W_{hy} \in \mathbb{R}^{H\times M}$ and $b_{hy} \in \mathbb{R}^M$ are the parameters of the linear transformation.

The conventional RNNs with AM ignore the fact that each input feature contributes differently to load forecasting. Besides, after applying the attention mechanism, the training of the resulting model become much more time-consuming because the traditional attention mechanism introduces much more learn-able parameters in the hidden layers of the model.

In order to resolve this problem, in this work, we propose a novel attention mechanism called input attention mechanism (IAM). Generally, IAM is effective and efficient, it not only maintains the similar capability to the existing attention mechanisms, i.e., assigning importance weights on the information that is more critical to the current mission, but also needs much less additional parameters, which makes it consume much less computational costs and more efficient than the existing attention mechanisms.

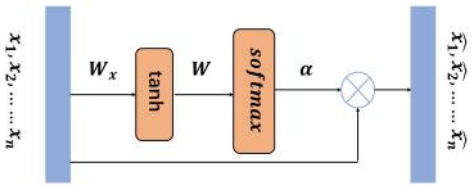


FIGURE 7. The structure of input attention mechanism.

Specifically, as shown in Figure 7, IAM applies an attention mechanism on the input layer to assign different weights for the model's inputs, the learn-able parameters are greatly reduced because it does not need to use the typical encoder-decoder network as the existing attention mechanisms. Formally, IAM can be defined as follows:

$$\begin{aligned} h_j &= tanh(W_j x_t + b_j), \\ \alpha_j &= \frac{exp(h_j)}{\sum_{j=1}^{n} exp(h_j)}, \\ \hat{x_t} &= \alpha_j \odot x_t, \end{aligned} \tag{18}$$

where $x_t \in \mathbb{R}^N$ denotes the input sequence, $W_j \in \mathbb{R}^{N\times H}$ and $b_j \in \mathbb{R}^H$ donates the weight matrix and the bias matrix, α_j is the attention weight measuring the importance of the input sequence. A softmax function is applied to h_j to ensure all the attention weights sum to 1.

C. HIDDEN CONNECTION MECHANISM

Depth has an essential influence on the expression of neural networks. The deep neural networks naturally integrate the characteristics of different levels of features. The deeper the network, the richer the feature levels that can be extracted. Therefore, deeper network structures are generally used in order to obtain higher-level features. However, the deeper structure of RNNs leads to problematic and slow training progress. A hidden connection mechanism (HCM) is proposed to tackle the issue by implementing the residual connection mechanism on the hidden layers of RNNs. The residual connection mechanism can be realized in the form of skip connection of different layers, that is, the input of the layer is directly added to the output of the layer.

The residual connection mechanism was first proposed to address gradient degradation issues of the deep network [31]. In conventional neural networks, the direct mapping function is generally used to establish the connection between the input and the output. The principal formula of traditional neural networks is as follows:

$$Y = F(X), \tag{19}$$

where X denotes the input, $F(\cdot)$ denote the mapping function, and Y denotes the output.

Instead of learning a direct mapping function $F(\cdot)$ from the input X to the output Y, the residual connection mechanism defines the output as a linear superposition and a nonlinear transformation of the input:

$$Y = F(X) + X. \tag{20}$$

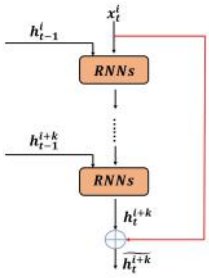


FIGURE 8. The structure of hidden connection mechanism.

Essentially, the residual connection mechanism is to add an identity mapping via a shortcut connection from the input layer to the output layer. Thus, the shortcut connections of the hidden layer are established in HCM. Figure 8 shows the structure of HCM, and the shortcut for identity mapping is highlighted by the red line. In theory, any two hidden layers can be connected by HCM, and the output of HCM of layer i and layer $i+k$ at time step t can be expressed as:

$$\widetilde{h_t^{i+k}} = h_t^{i+k} + W_{trans} \odot x_t^i, \tag{21}$$

where $\widetilde{h_t^{i+k}} \in \mathbb{R}^H$ is the output of HCM, $h_t^{i+k} \in \mathbb{R}^H$ is the hidden state of RNNs layer $i+k$, $x_t^i \in \mathbb{R}^N$ is the input of RNNs layer i, $W_{trans} \in \mathbb{R}^{N \times H}$ is used to match the dimensions of h_t^{i+k} and x_t^i.

IV. EXPERIMENTS

A. DESCRIPTION OF DATASET

Forecasting models are validated on two datasets for different prediction intervals. The PJM dataset[1] is used to verify the effectiveness of the proposed mechanism in daily short-term load forecasting, which includes load values in three months. The time span of the PJM dataset is from 2019.8.1 00 : 00 to 2019.11.1 00 : 00, and the sampling interval is one hour. The EG dataset[2] used to evaluate the performance of the proposed mechanism in weekly short-term load forecasting that includes historical load values in a year. The time span of the PJM dataset is from 2018.1.1 00 : 00 to 2019.1.1 00 : 00, and the sampling interval is also one hour. The raw data of two datasets are shown in Figure 9, which present characteristics without obvious periodicity. The mentioned datasets are divided into train dataset, validation dataset and test dataset, which is used for training, validating and testing the forecasting models, respectively. The details of datasets are shown as Table 1.

TABLE 1. Divisions of PJM and EG dataset.

Dataset	Train Set	Validation Set	Test Set
PJM	2019.8.1 - 2019.10.23	2019.10.24 - 2019.10.30	2019.10.31
EG	2018.1.1 - 2018.11.30	2018.12.1 - 2018.12.24	2018.12.25 - 2018.12.31

B. DATA NORMALIZATION

To ensure the input values are within the same scale and make the training of the RNNs easier and more stable, the dataset is normalized. The mean and variance are calculated to normalize the whole dataset, then ensure the capacity of the entire network through scale changes and offsets. The normalization algorithm can be expressed as the following equations:

$$\begin{aligned} \mu &= \frac{1}{N}\sum_{i=1}^{N} x_i, \\ \sigma^2 &= \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2, \\ \hat{x_i} &= \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}, \end{aligned} \tag{22}$$

where N is the size of the dataset, $x_i \in \mathbb{R}^N$ is the i^{th} term of the dataset, ϵ is a minimum value used to prevent the denominator from being zero.

C. SLIDING WINDOW FOR DATA DIVISION

Sliding window method is used in this work to divide the normalized sequential load forecasting data in both datasets into

[1] https://www.pjm.com
[2] http://www.eirgridgroup.com

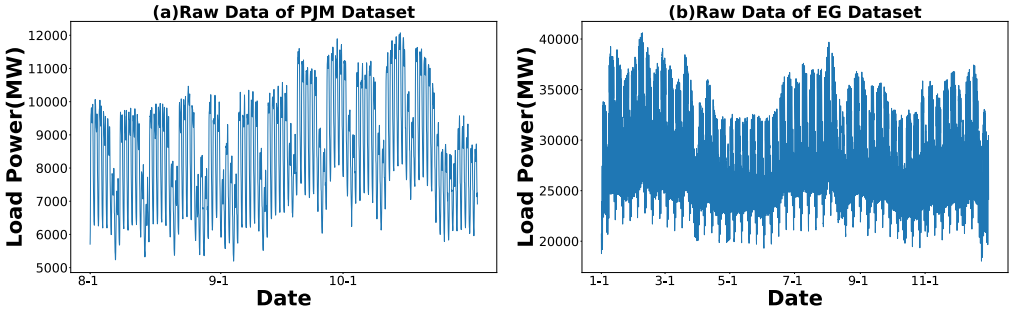


FIGURE 9. Raw data of PJM and EG dataset.

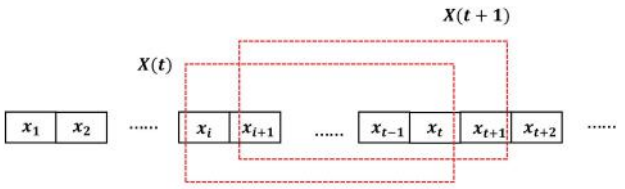


FIGURE 10. Sliding window method for processing dataset.

some subsequences with the same length, which will be used as the input sequence of the RNN-based models. Specifically, as shown in Figure 10, suppose $X(t) = [x_i, x_{i+1}, \ldots, x_t]$ is an input sequence at time step t with the length K of the input sequence set as $t - i + 1$, then the input sequence at the next time step $t+1$ will be $X(t+1) = [x_{i+1}, x_{i+2} \ldots, x_{t+1}]$, where $i \geq 1$ and $t > i$. Finally, a dataset with the size of N will be divided into $N - K + 1$ input sequences.

D. NETWORK CONFIGURATION

- Initialization. In the load forecasting, random uniform initialization is used in the traditional RNNs, which can not guarantee the eigenvalues of the weight matrices. It may result in gradient explosion or gradient vanishing. Therefore, the orthogonal initialization is introduced to ensure that eigenvalue of weight matrix is 1, avoiding gradient explosion or gradient vanishing so that the gradient matrix can be better back-propagated [32].
- Structural Hyperparameter. To verify the effectiveness of the proposed mechanism, the structural hyperparameters are set to uniform values. The dimension of input sequence at time t is set to 9, the dimension of hidden state of RNNs is set to 10, the dimension of the output sequence at time t is set to 1. The number of RNN layers is set to 32, and the number of layers that the HCM skips is set to 2.
- Learning Rate. The initial learning rate is set to 0.001, and the learning rate decay is adopted to decrease the learning rate automatically as the number of iterations increases. Specifically, the learning rate is multiplied by 0.9998 every 5 epochs.
- Optimization Algorithm. Adam [29] is utilized as the optimization algorithm for updating the learn-able parameters in neural network. The advantage of Adam is that the learning rate has a certain range for each iteration, which updates the learn-able parameters more stable. The hyperparameters of Adam are set as: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.

E. EVALUATION METRICS

Accuracy and Efficiency are the two most important metrics for judging the quality of short-term load forecasting models. In this paper, the root mean square error (RMSE) and mean absolute percentage error (MAPE) are calculated to evaluate forecasting accuracy. The smaller the values of RMSE and MAPE, the better the forecasting accuracy. The evaluation indexes of accuracy are defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{23}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\hat{y}_i - y_i}{y_i}, \tag{24}$$

where N is the number of the load values, y_i is the true load value, $\hat{y}_i$ is the predicted value of forecasting models.

The convergent time is used as the criterion of forecasting efficiency. The traditional method to judge the convergence of forecasting models is to observe the loss curve, which is subjective. Thus a new algorithm of judging convergence is proposed that precisely achieves the convergence time of the model. The algorithm for judging convergence of forecasting models can be expressed as Algorithm 1, where the threshold of training loss to validation loss ratio t_r is set as 1.05, and the threshold of epoch continue number t_c is set as 5.

Furthermore, t-test is used to infer whether there is a difference between the mean of forecasting results of the proposed model and other forecasting models. The t-test can be expressed as following equations:

$$t = \frac{\overline{X_1} - \overline{X_2}}{S_p \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$
$$S_p = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}}, \tag{25}$$

Algorithm 1 Algorithm for Judging Convergence

Input:

The training loss set $L = [L_1, L_2, \ldots, L_n]$, the validation loss set $V = [V_1, V_2, \ldots, V_n]$, the threshold of training loss to validation loss ratio t_r, and the threshold of epoch continue number t_c.

Output:

The convergent epoch e.

```
for e = 0 to n do
    R_i = L_i / V_i;
    if R_{i+1} − R_i < t_r then
        count+ = 1;
        if count > t_c then return e
        end if
    else
        count = 0;
    end if
end for
```

where $\overline{X_1}$ and $\overline{X_2}$ are the sample mean, S_1^2 and S_2^2 are standard error, and n_1 and n_2 are the size of two samples.

Null hypothesis H_0 and alternative hypothesis H_1 are defined as follows:

$$H_0 : \mu_1 = \mu_2,$$
$$H_1 : \mu_1 \neq \mu_2, \tag{26}$$

where μ_1 and μ_2 are the mean of two examples. The p value determines whether or not to reject null hypothesis. The smaller the p value, the more confident to reject null hypothesis.

F. BENCHMARK

In order to show the superior performance of the proposed RNN-based load forecasting with IAM and HCM models, we have used 12 benchmarking models as the state-of-the-art baselines in our experimental studies. In general, all the existing RNN-based short-term load forecasting models can be categorized into three groups:

- The first group is the models based on the effective variants of the vanilla RNNs; in this group, previous studies show that LSTM and GRU achieve the best performances, so we choose LSTM and GRU as the benchmarking models in this work.
- The second group is the models based on the bidirectional RNNs, where bidirectional LSTM and bidirectional GRU achieve the better performances than others, so we also select bidirectional LSTM and bidirectional GRU as the benchmarking models in this work.
- The third group is the models that additionally apply some performance improvement mechanisms to enhance the performances of RNN-based models in short-term load forecasting; to the best of our knowledge, attention mechanism (AM) and CNN, are the state-of-the-art improvement solutions within this group; so we integrate AM and CNN respectively with the selected four state-of-the-art RNNs in the above two groups (i.e., LSTM, GRU, Bi-LSTM, and Bi-GRU) to form another eight state-of-the-art solutions for the short-term load forecasting, i.e., LSTM with CNN, LSTM with AM, GRU with CNN, GRU with AM, Bi-LSTM with CNN, Bi-LSTM with AM, Bi-GRU with CNN, and Bi-GRU with AM.

Consequently, by selecting these 12 RNN-based load forecasting solutions as the benchmarks, we have already considered all the-state-of-the-art solutions in the RNN-based short-term load forecasting task as the baselines in our experimental studies. Therefore, it is reasonable to conclude that the proposed RNNs with IAM and HCM models are superior to other deep models in short-term load forecasting if they outperform all the selected benchmarking models. In addition, to make a fair comparison, hyperparameters such as the learning rate, the dimension of the input sequences, the number of hidden layers, etc. are kept equal.

TABLE 2. The accuracy and efficiency on PJM dataset.

Models	RMSE(MW)	MAPE(%)	Convergent time(s)
LSTM	582.86	6.88	25.87
LSTM with CNN	462.22	4.57	37.94
LSTM with AM	389.48	4.31	43.21
LSTM with IAM and HCM	**287.51**	**3.17**	**23.18**
GRU	551.10	5.35	20.25
GRU with CNN	354.79	4.09	31.57
GRU with AM	328.06	3.57	36.98
GRU with IAM and HCM	**190.28**	**1.79**	**19.61**
Bi-LSTM	314.53	3.63	48.42
Bi-LSTM with CNN	217.54	2.35	65.80
Bi-LSTM with AM	162.17	1.81	72.63
Bi-LSTM with IAM and HCM	**92.66**	**1.06**	**42.15**
Bi-GRU	245.06	2.70	39.20
Bi-GRU with CNN	191.13	2.17	58.44
Bi-GRU with AM	161.02	1.46	63.79
Bi-GRU with IAM and HCM	**69.58**	**0.78**	**35.29**

G. RESULTS AND ANALYSIS

1) RESULTS AND ANALYSIS OF DAILY PREDICTION ON PJM DATASET

Figure 11 shows the daily forecasting curves of different models on the PJM dataset. It can be noticed that the predicting trends of models are all consistent with the actual trends, the RNNs with IAM and HCM especially fits better than other models. The predicting accuracy and efficiency of RNN-based models are demonstrated in Table 2. We can observe that the RNN-based models for short-term load forecasting have high accuracy because deep learning methods can better extract the valid features of time series automatically. However, using the RNN-based model directly for short-term load forecasting does not meet the requirement of the smart grid due to the severe deviation on the peak value and valley value. Since the CNN layer can extract complex

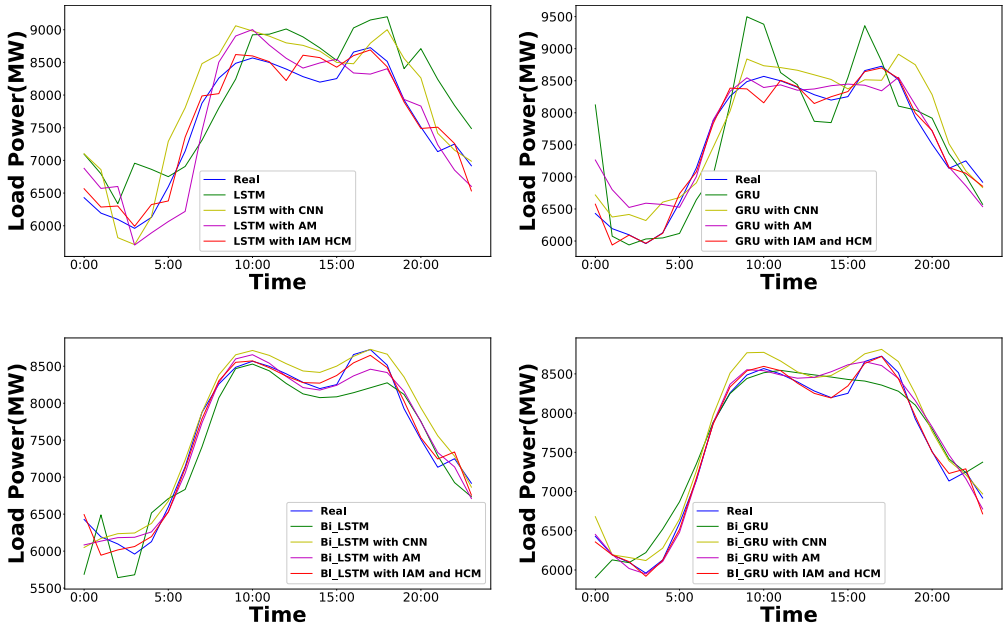


FIGURE 11. Daily prediction results of all models on PJM dataset.

TABLE 3. Comparison of t statistic and p value on PJM dataset.

Models	LSTM with IAM and HCM			GRU with IAM and HCM		
	LSTM	LSTM with CNN	LSTM with AM	GRU	GRU with CNN	GRU with AM
t	1.374	1.001	-0.156	0.786	0.729	0.433
p	0.176	0.322	0.876	0.273	0.469	0.667
Models	Bi-LSTM with IAM and HCM			Bi-GRU with IAM and HCM		
	Bi-LSTM	Bi-LSTM with CNN	Bi-LSTM with AM	Bi-GRU	Bi-GRU with CNN	Bi-GRU with AM
t	-0.520	0.481	-0.054	0.254	0.586	0.204
p	0.606	0.633	0.957	0.799	0.561	0.838

variable features, the accuracy is significantly improved by adding the CNN layer. Nonetheless, the forecasting efficiency is greatly reduced due to the convolution operation. Similarly, After the AM applied, the predicting accuracy has a great leap, because it stressed on the effective values that have more influence on the forecasting models. However, it is at the expense of the convergent time due to the increase in parameters caused by the typical encoder-decoder framework. By contrast, the RNN-based models with IAM and HCM not only improves the accuracy but also significantly decreases the convergent time.

To verify the generalization of the proposed mechanism, the test is expanded to the Bi-RNN-based models. Table 2 and Figure 11 show the evaluation index and forecasting curves, respectively. It can be seen that the Bi-RNN-based model performs better than the RNN-based model in terms of accuracy. That is because Bi-RNN-based models take both historical load values and future load values into consideration. Due to the increase in learn-able parameters, the convergent time of Bi-RNN-based models also increases accordingly. After applying the CNN layers, the Bi-RNN-based model performs better in terms of accuracy at the expense of efficiency. Bi-RNN-based models with AM improves the accuracy of prediction, but they also consume enormous training time. While maintaining high efficiency, Bi-RNN-based models with the proposed mechanism can achieve satisfactory performance in terms of accuracy. Based on the preliminary comparison, recurrent neural networks with the proposed mechanism are superior to the other models in terms of accuracy and efficiency, indicating that the application of IAM and HCM into RNNs can fully extract the complex correlations between load values, being beneficial to improve the accuracy and decrease the convergent time.

The t statistics and p values of different forecasting models on PJM are shown in Table 3. The p value determines whether or not to reject the null hypothesis. The results of RNNs with IAM and HCM are most different from the original RNNs, and the closet to RNNs with AM. This is because RNNs with AM have higher accuracy than the original RNNs, and the accuracy of RNNs with IAM and HCM is relatively similar to that of RNN with AM. Through the statistic test, it is convinced that the proposed forecasting model is different from other forecasting models.

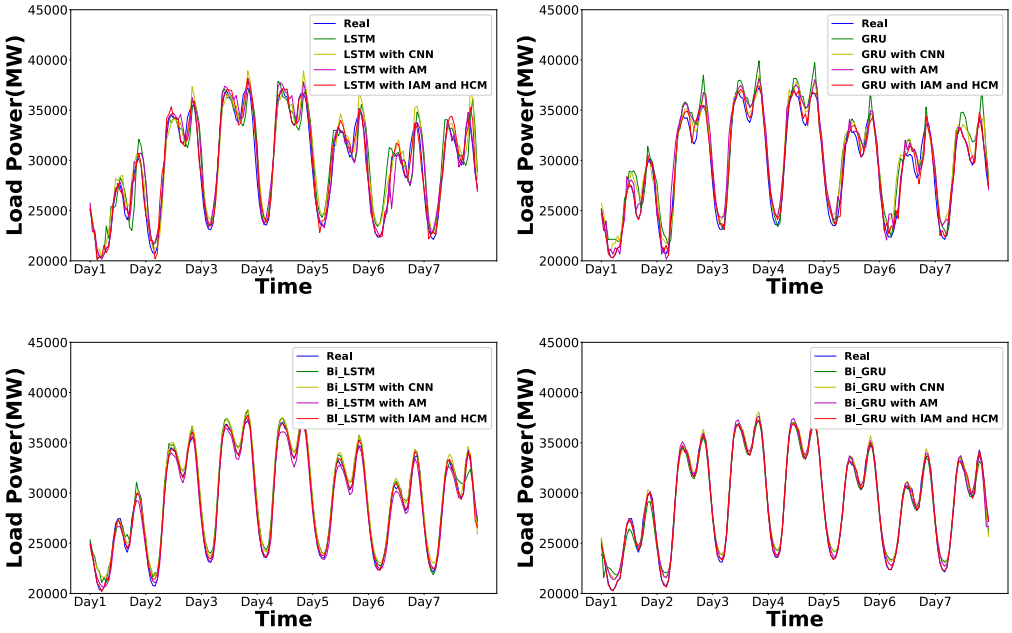


FIGURE 12. Weekly prediction results of all models on EG dataset.

2) RESULTS AND ANALYSIS OF WEEKLY PREDICTION ON EG DATASET

To further verify the forecasting ability on different prediction intervals of the proposed mechanism, the weekly prediction has been made on EG. As can been seen from Figure 12, for RNN-based models and Bi-RNN-based models, the result of prediction has revealed a similar trend. Due to poor forecasting performance on peak and valley values, RNN-based models can not be practical to the smart grid. After extracting effective features through CNN, the accuracy of RNN-based models is significantly enhanced, but the efficiency is reduced due to the complexity of convolution operations. After applying the AM, the forecasting accuracy has a great leap by assigning the importance weights on different features. However, the increase in terms of accuracy comes at the cost of decreasing efficiency, and poor efficiency still does not meet the requirement of the smart grid. Thus, the RNN-based models with IAM and HCM have been proposed to keep high accuracy and reduce the convergent time. As shown in Table 4, the RNN-based models with the proposed mechanism have a decline of 440MW and 0.99 on average, and its convergent time decreases by 39.98%. Similarly, the Bi-RNN-based models have better performance than RNN-based models by taking historical information and future information into consideration. Due to the increase in the number of learn-able parameters, the convergent time of model training has also increased dramatically. Whether it is CNN or AM, their combination with Bi-RNN-based models greatly increased the convergent time. With the application of IAM and HCM, accuracy and efficiency have been improved. The RMSE, MAPE, and convergent time of Bi-RNN-based models with IAM and HCM decreased by 49.58%, 55.87% and 41.88%. Based on the above comparison results, the conclusion can be drawn that the RNNs with the proposed mechanism have the advantage in terms of accuracy and efficiency on different prediction intervals.

TABLE 4. The accuracy and efficiency on EG dataset.

Models	RMSE(MW)	MAPE(%)	Convergent time(s)
LSTM	1.77e3	4.64	90.25
LSTM with CNN	1.36e3	3.87	130.54
LSTM with AM	1.27e3	3.09	146.83
LSTM with IAM and HCM	**0.83e3**	**2.15**	**84.42**
GRU	1.45e3	4.13	79.16
GRU with CNN	1.23e3	3.61	108.46
GRU with AM	1.14e3	2.94	121.59
GRU with IAM and HCM	**0.71e3**	**1.91**	**72.74**
Bi-LSTM	0.71e3	2.10	169.27
Bi-LSTM with CNN	0.66e3	2.04	230.67
Bi-LSTM with AM	0.57e3	1.64	258.71
Bi-LSTM with IAM and HCM	**0.34e3**	**0.89**	**162.52**
Bi-GRU	0.69e3	1.97	133.54
Bi-GRU with CNN	0.55e3	1.65	196.81
Bi-GRU with AM	0.51e3	1.53	226.79
Bi-GRU with IAM and HCM	**0.21e3**	**0.52**	**121.13**

The t statistics and p values of different forecasting models on EG dataset are shown in Table 5. As the prediction interval increases, the proposed model has been significantly different from other forecasting models. Specifically, the p value between the original RNNs and the proposed model is 0.055 on average, indicating the results of RNNs with IAM and HCM is distinct from the results of the original RNNs. In general, the forecasting error of RNNs with IAM and

TABLE 5. Comparison of t statistic and p value on EG dataset.

Models	LSTM with IAM and HCM			GRU with IAM and HCM		
	LSTM	LSTM with CNN	LSTM with AM	GRU	GRU with CNN	GRU with AM
t	-1.576	-0.471	0.153	1.218	1.398	0.850
p	0.073	0.237	0.377	0.062	0.123	0.295
Models	Bi-LSTM with IAM and HCM			Bi-GRU with IAM and HCM		
	Bi-LSTM	Bi-LSTM with CNN	Bi-LSTM with AM	Bi-GRU	Bi-GRU with CNN	Bi-GRU with AM
t	-0.742	-0.938	-0.673	-0.142	0.564	0.310
p	0.048	0.149	0.301	0.037	0.272	0.156

HCM is lower than the other models, and there are differences between them.

H. ABLATION STUDY

In order to investigate the functionality and necessity of IAM and HCM in the proposed model, ablation studies are further conducted by implementing two types of additional models: i) RNN-based models with solely IAM, and ii) RNN-based models with solely HCM. The results of ablation studies on two datasets are shown in Table 6 and Table 7, where we can make the following observations:

TABLE 6. Ablation studies of the proposed RNNs with IAM and HCM on PJM dataset.

Models	RMSE(MW)	MAPE(%)	Convergent time(s)
LSTM	582.86	6.88	25.87
LSTM with IAM	328.35	3.76	38.55
LSTM with HCM	500.99	5.89	**12.10**
LSTM with IAM and HCM	**287.51**	**3.17**	23.18
GRU	551.10	5.35	20.25
GRU with IAM	283.93	2.79	31.24
GRU with HCM	443.47	4.54	**10.90**
GRU with IAM and HCM	**190.28**	**1.79**	19.61
Bi-LSTM	314.53	3.63	48.42
Bi-LSTM with IAM	158.67	1.59	67.92
Bi-LSTM with HCM	251.02	2.71	**29.27**
Bi-LSTM with IAM and HCM	**92.66**	**1.06**	42.15
Bi-GRU	245.06	2.70	39.20
Bi-GRU with IAM	126.52	1.41	55.11
Bi-GRU with HCM	180.08	2.19	**19.82**
Bi-GRU with IAM and HCM	**69.58**	**0.78**	35.29

First, by applying IAM to all RNNs (including LSTM, GRU, Bi-LSTM, and Bi-GRU), the load forecasting accuracy is always greatly increased on both datasets while the learning efficiency is always decreased because IAM still introduces additional parameters into the models. Specifically, compared with the original LSTM, the RMSE and MAPE drop by 39.04% and 40.75 on average after employing IAM. Similarly, the application of IAM makes the RMSE and MAPE drop by 40.10% and 41.36% on average compared to original GRU. However, the efficiency is reduced because IAM still has many learn-able parameters. Compared with original LSTM and original GRU, the efficiency decreases by 48.08% and 47.04% on average by combining the IAM. The results of bidirectional models also show the same pattern

TABLE 7. Ablation studies of the proposed RNNs with IAM and HCM on EG dataset.

Models	RMSE(MW)	MAPE(%)	Convergent time(s)
LSTM	1.77e3	4.64	90.25
LSTM with IAM	1.16e3	2.87	132.79
LSTM with HCM	1.55e3	4.25	**63.78**
LSTM with IAM and HCM	**0.83e3**	**2.15**	84.42
GRU	1.45e3	4.13	79.16
GRU with IAM	0.99e3	2.69	110.67
GRU with HCM	1.38e3	3.84	**52.06**
GRU with IAM and HCM	**0.71e3**	**1.91**	72.74
Bi-LSTM	0.71e3	2.10	169.27
Bi-LSTM with IAM	0.44e3	1.19	231.88
Bi-LSTM with HCM	0.62e3	1.86	**109.45**
Bi-LSTM with IAM and HCM	**0.34e3**	**0.89**	162.52
Bi-GRU	0.69e3	1.97	133.54
Bi-GRU with IAM	0.42e3	1.21	197.36
Bi-GRU with HCM	0.60e3	1.75	**98.50**
Bi-GRU with IAM and HCM	**0.21e3**	**0.52**	121.13

after adding the IAM module. Compared with the original Bi-LSTM, using IAM can reduce RMSE and MAPE by 43.79% and 47.77%, while reducing efficiency by 38.63%. The conclusion is similar to Bi-GRU that using IAM can reduce RMSE and MAPE by 43.75% and 55.30%, while reducing efficiency by 37.81%.

Second, in contrast, applying HCM to all RNNs can greatly enhance the original RNNs' learning efficiency, but only slightly increase their accuracy. Specifically, the RMSE and MAPE decline by 13.24% and 11.40 on average, while the efficiency is improved by 41.28% after applying HCM to the original LSTM. For the original GRU, the convergent time reduces by 40.20%, and the RMSE and MAPE drop by 12.18% and 11.08% with the application of HCM. The same conclusion can be obtained in the bidirectional models. Compared with the original Bi-LSTM, the RMSE and MAPE decline by 16.44% and 18.39%, while improving efficiency by 37.45% by adding HCM module. Similarly, the convergent time reduces by 37.84%, and the RMSE and MAPE drop by 19.78% and 15.03% in the case of using HCM compared with Bi-GRU.

Third, by integrating both IAM and HCM into RNNs, the resulting models can achieve great improvements in load-forecasting accuracy, and also slightly enhance the learning efficiency. Consequently, these observations in

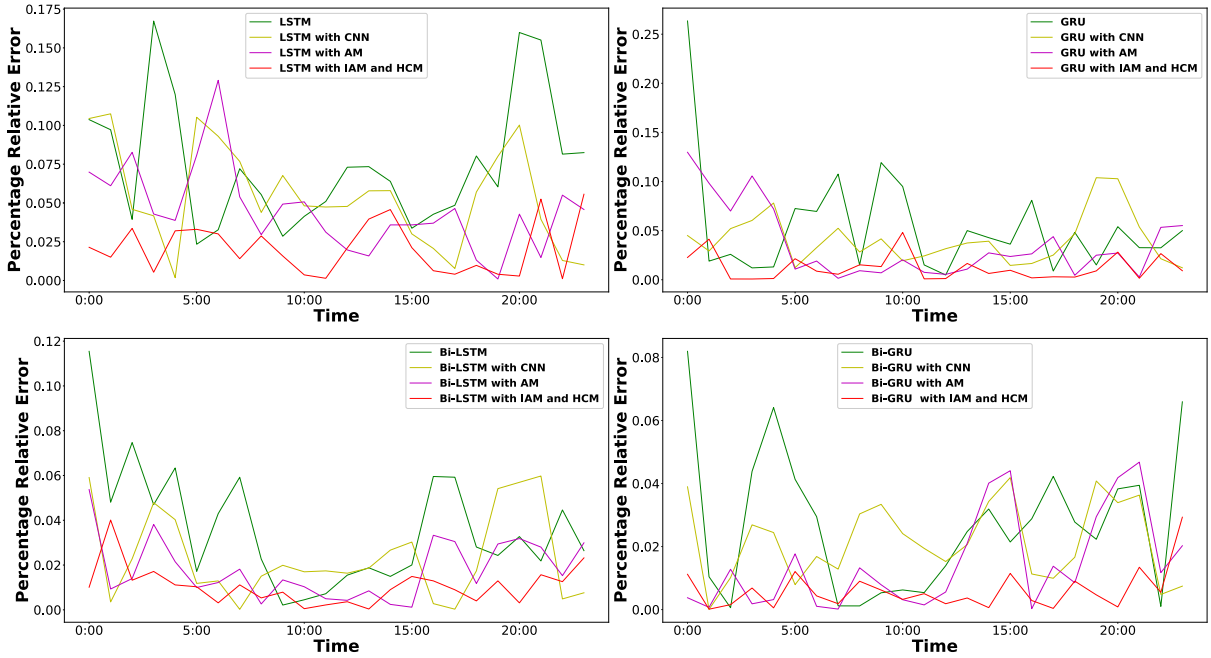


FIGURE 13. Daily prediction results in percentage relative error (PRE) on PJM dataset.

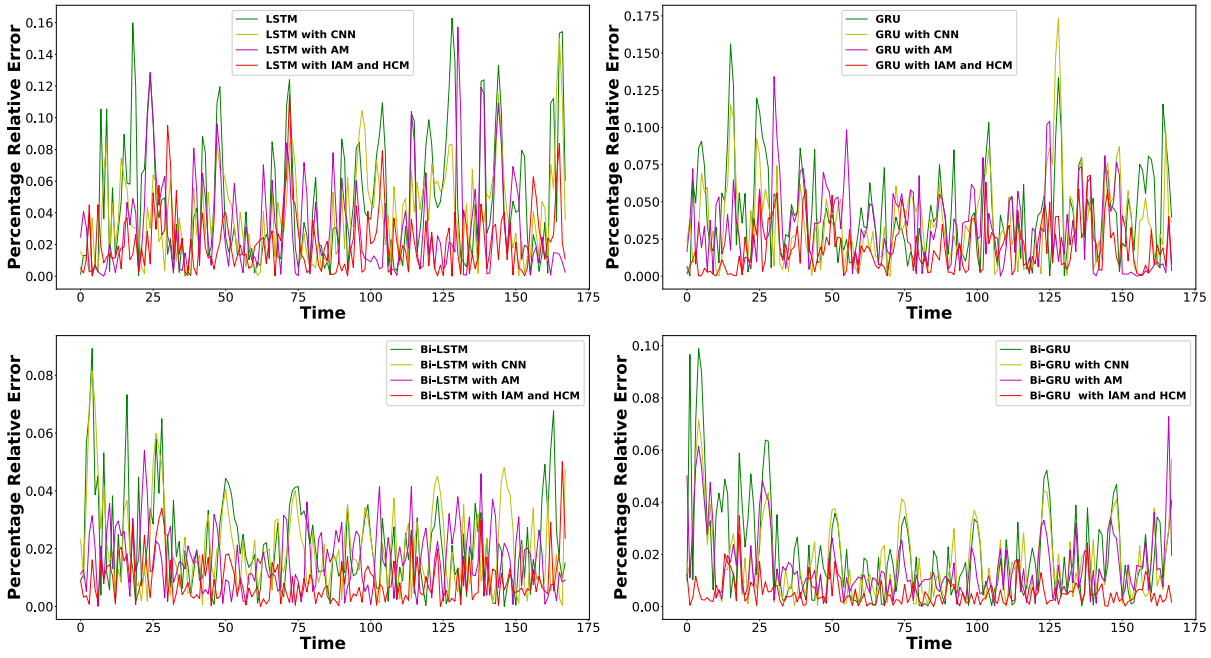


FIGURE 14. Weekly prediction results in percentage relative error (PRE) on EG dataset.

ablation studies exhibit that both IAM and HCM are essential and complement to each other to make the proposed models achieve the superior performances.

I. OTHER INSIGHTS

To further explore the underlying reasons for the superior performance of the proposed mechanisms, we further introduce an additional evaluating metric, i.e., the percentage relative error (PRE), which is calculated by the following equation:

$$\delta_t = \frac{|y_t - \hat{y_t}|}{y_t} \times 100\% \tag{27}$$

The distribution curves of PRE for various RNN-based and Bi-RNN-based models on two datasets are shown in Figure 13 and Figure 14, respectively. It can be noticed

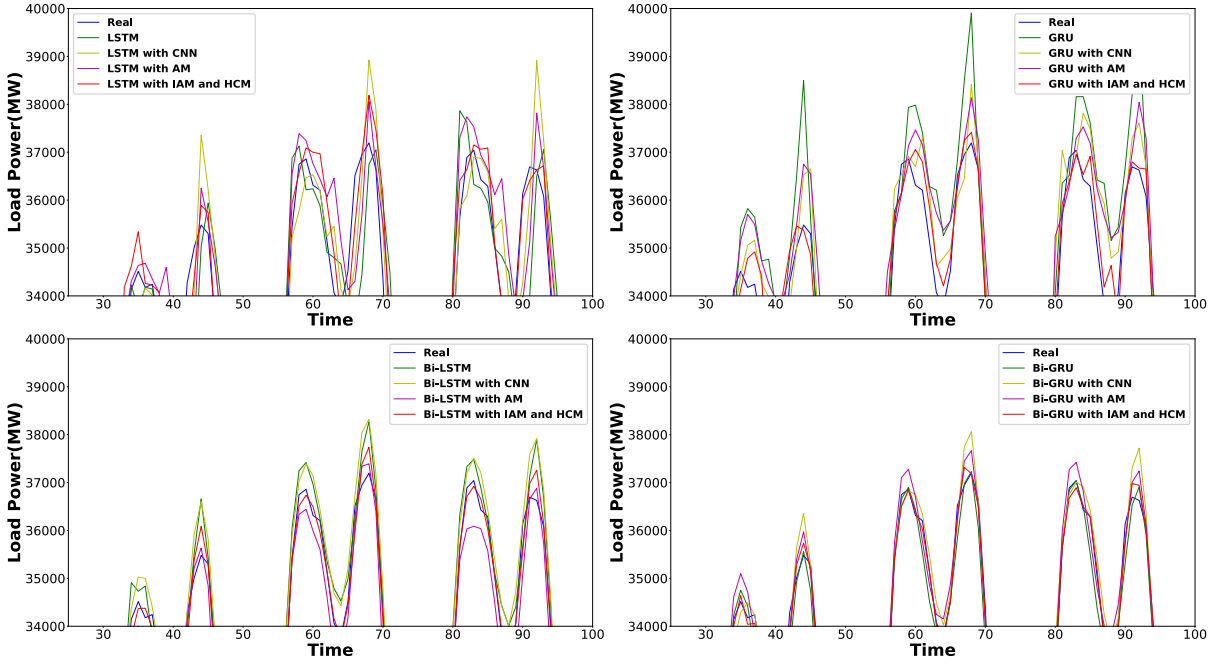


FIGURE 15. The peak load values of different models on EG dataset.

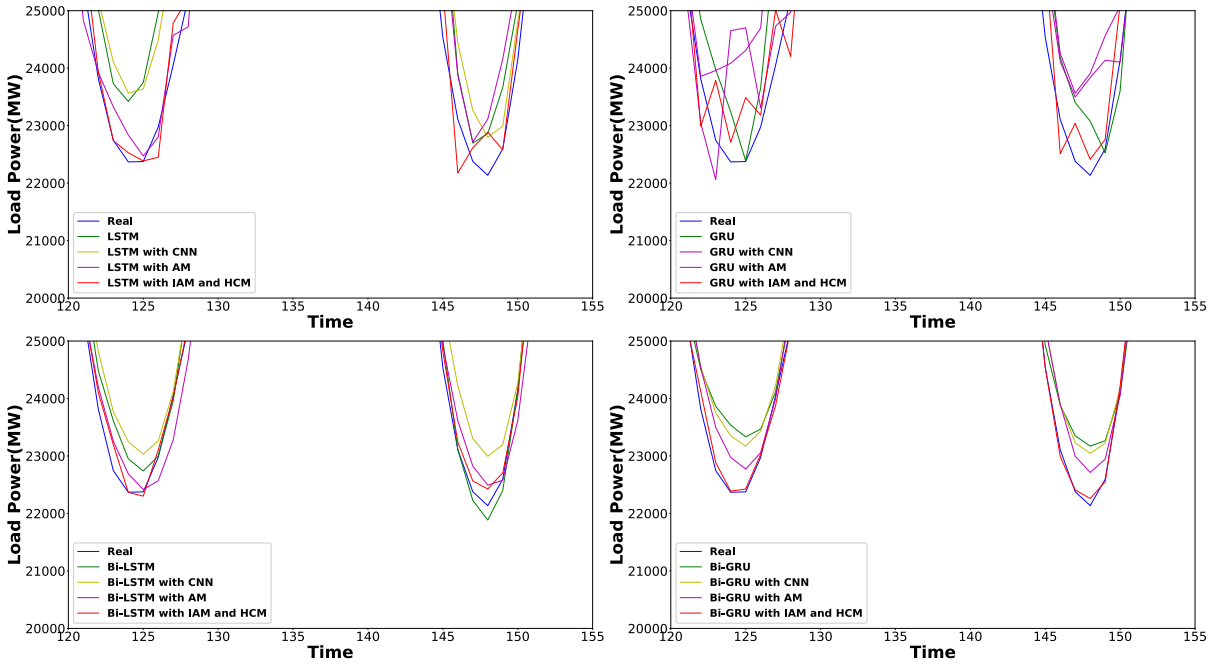


FIGURE 16. The valley load values of different models on EG dataset.

that using IAM and HCM can generally reduce the percentage relative errors of RNN-based models and Bi-RNN-based models. Specifically, in the daily prediction of PJM, the PRE of IAM is significantly better than PRE of CNN and AM in terms of fluctuation range and overall error level in RNN-based models and Bi-RNN-based models. This advantage is particularly prominent in GRU-based models. Similarly, the conclusion can still be drawn in weekly prediction with longer prediction intervals. Consequently, this observation states a possible reason for the proposed models to

achieve superior performances: RNNs with IAM and HCM is not only more accurate but also more stable than the state-of-the-art baselines.

Furthermore, Figure 11 and Figure 12 show an observations that the predicted load values of all deep models at the peak and valley points are generally less accurate than those at other points. Consequently, we further enlarge Figure 12 to show the detailed curves for peak and valley load values in Figure 15 and Figure 16, respectively. It can be seen from Figure 15 that the predicted values of our proposed mechanism are generally much closer to the true values than those of the state-of-the-art baselines at peak points; similarly, as shown in Figure 16, the proposed RNNs with IAM and HCM models achieve generally more accurate load forecasting than the state-of-the-art baselines at valley points. This thus states that the superior performances achieved by the proposed RNNs with IAM and HCM may also because they can achieve generally much better performances than the state-of-the-art baselines in forecasting the peak and valley load values.

V. CONCLUSION AND FUTURE WORK

In this paper, a novel mechanism, called input attention mechanism and hidden connection mechanism is proposed to enhance the forecasting accuracy and efficiency of recurrent neural networks, including RNN-based models and Bi-RNN-based models for short-term load forecasting. Unlike the traditional attention mechanism, the input attention mechanism is to assign the importance weights into the input sequences, keeping the high-level accuracy of the attention mechanism and reduce the learn-able parameters. Besides, the hidden connection mechanism can speed up the convergence of training, improving the efficiency of recurrent neural networks.

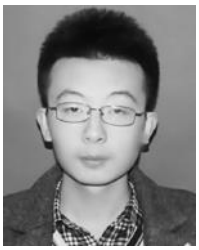
Plenty of experiments on two public datasets are carried out to illustrate the superiority of the proposed mechanism. It turns out that the models with IAM and HCM have higher accuracy and faster training efficiency than any other forecasting models. Furthermore, experiments of RNNs with IAM and RNNs with HCM are carried out to determine the role of the proposed mechanism, results show that RNNs with IAM is substantially improved in terms of accuracy but decreased in terms of efficiency, while RNNs with HCM mainly contributes greatly to efficiency. In addition, insightful discussion of the results is provided to further explore the underlying reasons for the superior performance of the proposed mechanism. The experimental results indicate that the proposed model achieves an excellent prediction performance by reducing the error in peak load values and valley load values.

In the future, more influencing factors will be taken into consideration to improve the forecasting accuracy further. Moreover, our proposed mechanism can be applied to more general RNN-based models and Bi-RNN-based models to keep high forecasting accuracy and decrease the convergent time during training.

REFERENCES

[1] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *Int. J. Forecasting*, vol. 32, no. 3, pp. 914–938, Jul. 2016.
[2] K. Zor, O. Timur, and A. Teke, "A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting," in *Proc. Int. Youth Conf. Energy (IYCE)*, 2017, pp. 1–7.
[3] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—A novel pooling deep RNN," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Mar. 2017.
[4] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.
[5] A. Narayan and K. W. Hipel, "Long short term memory networks for short-term electric load forecasting," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 2573–2578.
[6] R. Jiao, T. Zhang, Y. Jiang, and H. He, "Short-term non-residential load forecasting based on multiple sequences lstm recurrent neural network," *IEEE Access*, vol. 6, pp. 59438–59448, 2018.
[7] G. Xiuyun, W. Ying, G. Yang, S. Chengzhi, X. Wen, and Y. Yimiao, "Short-term load forecasting model of GRU network based on deep learning framework," in *Proc. 2nd IEEE Conf. Energy Internet Energy Syst. Integr. (EI2)*, Oct. 2018, pp. 1–4.
[8] Y. Wang, W. Liao, and Y. Chang, "Gated recurrent unit network-based short-term photovoltaic forecasting," *Energies*, vol. 11, no. 8, p. 2163, Aug. 2018.
[9] W. Li, T. Logenthiran, and W. L. Woo, "Multi-GRU prediction system for electricity generation's planning and operation," *IET Gener., Transmiss. Distrib.*, vol. 13, no. 9, pp. 1630–1637, May 2019.
[10] S. Fallah, M. Ganjkhani, S. Shamshirband, and K.-W. Chau, "Computational intelligence on short-term load forecasting: A methodological overview," *Energies*, vol. 12, no. 3, p. 393, Jan. 2019.
[11] X. Tang, Y. Dai, Q. Liu, X. Dang, and J. Xu, "Application of bidirectional recurrent neural network combined with deep belief network in short-term load forecasting," *IEEE Access*, vol. 7, p. 160 660–160 670, 2019.
[12] Y. Liu, K. Zhang, S. Zhen, Y. Guan, and Y. Shi, "Wrl: A combined model for short-term load forecasting," in *Proc. Asia–Pacific Web (APWeb) Web-Age Inf. Manage. (WAIM) Joint Int. Conf. Web Big Data*, 2019, pp. 35–42.
[13] X. Tang, Y. Dai, T. Wang, and Y. Chen, "Short-term power load forecasting based on multi-layer bidirectional recurrent neural network," *IET Gener., Transmiss. Distrib.*, vol. 13, no. 17, pp. 3847–3854, 2019.
[14] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using CNN-LSTM neural networks," *Energy*, vol. 182, pp. 72–81, Sep. 2019.
[15] M. Sajjad, Z. A. Khan, A. Ullah, T. Hussain, W. Ullah, M. Y. Lee, and S. W. Baik, "A novel CNN-GRU-based hybrid approach for short-term residential load forecasting," *IEEE Access*, vol. 8, pp. 143759–143768, 2020.
[16] Z. Meng and X. Xu, "A hybrid short-term load forecasting framework with an attention-based encoder–decoder network based on seasonal and trend adjustment," *Energies*, vol. 12, no. 24, p. 4612, 2019.
[17] Y. Ju, J. Li, and G. Sun, "Ultra-short-term photovoltaic power prediction based on self-attention mechanism and multi-task learning," *IEEE Access*, vol. 8, pp. 44821–44829, 2020.
[18] J. Du, Y. Cheng, Q. Zhou, J. Zhang, X. Zhang, and G. Li, "Power load forecasting using BiLSTM-attention," *E&ES*, vol. 440, Mar. 2020, Art. no. 032115.
[19] A. Ahmad, N. Javaid, A. Mateen, M. Awais, and Z. A. Khan, "Short-term load forecasting in smart grids: An intelligent modular approach," *Energies*, vol. 12, no. 1, p. 164, Jan. 2019.
[20] J. Zheng, C. Xu, Z. Zhang, and X. Li, "Electric load forecasting in smart grids using long short-term-memory based recurrent neural network," in *Proc. Annu. Conf. Inf. Sci. Syst. (CISS)*, 2017, pp. 1–6.
[21] R. K. Agrawal, F. Muchahary, and M. M. Tripathi, "Long term load forecasting with hourly predictions based on long-short-term-memory networks," in *Proc. IEEE Texas Power Energy Conf. (TPEC)*, Feb. 2018, pp. 1–6.
[22] J. Zheng, X. Chen, K. Yu, L. Gan, Y. Wang, and K. Wang, "Short-term power load forecasting of residential community based on gru neural network," in *Proc. Int. Conf. Power Syst. Technol. (POWERCON)*, 2018, pp. 4862–4868.
[23] Y. Deng, H. Jia, P. Li, X. Tong, X. Qiu, and F. Li, "A deep learning methodology based on bidirectional gated recurrent unit for wind power prediction," in *Proc. 14th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2019, pp. 591–595.

[24] S. Atef and A. B. Eltawil, "Assessment of stacked unidirectional and bidirectional long short-term memory networks for electricity load forecasting," *Electr. Power Syst. Res.*, vol. 187, Oct. 2020, Art. no. 106489.
[25] T. Le, M. T. Vo, B. Vo, E. Hwang, S. Rho, and S. W. Baik, "Improving electric energy consumption prediction using CNN and bi-LSTM," *Appl. Sci.*, vol. 9, no. 20, p. 4237, Oct. 2019.
[26] H. Wilms, M. Cupelli, and A. Monti, "Combining auto-regression with exogenous variables in sequence-to-sequence recurrent neural networks for short-term load forecasting," in *Proc. IEEE 16th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2018, pp. 673–679.
[27] P. Li, X. Wang, and J. Yang, "Short-term wind power forecasting based on two-stage attention mechanism," *IET Renew. Power Gener.*, vol. 14, no. 2, pp. 297–304, Feb. 2020.
[28] S. Wang, X. Wang, S. Wang, and D. Wang, "Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting," *Int. J. Electr. Power Energy Syst.*, vol. 109, pp. 470–479, Jul. 2019.
[29] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5998–6008.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[32] E. Vorontsov, C. Trabelsi, S. Kadoury, and C. Pal, "On orthogonality and learning recurrent networks with long term dependencies," 2017, *arXiv:1702.00071*. [Online]. Available: http://arxiv.org/abs/1702.00071



MINGFEI ZHANG (Graduate Student Member, IEEE) received the B.E. degree in electrical engineering from Northeast Electric Power University, Jilin, China, in 2018. He is currently pursuing the M.E. degree in electrical engineering with the Hebei University of Technology, Tianjin, China. His main research interests include load forecasting and demand response.

ZHOUTAO YU (Graduate Student Member, IEEE) received the B.E. degree from the Anhui University of Technology, in 2019. He is currently pursuing the M.S. degree in electrical engineering with the Hebei University of Technology, Tianjin, China. His research interests are deep learning, electricity market, and smart grids.

ZHENGHUA XU (Member, IEEE) received the M.Phil. degree in computer science from The University of Melbourne, Parkville, VIC, Australia, in 2012, and the D.Phil. degree in computer science from the University of Oxford, U.K., in 2018.

From 2017 to 2018, he worked as a Research Associate with the Department of Computer Science, University of Oxford. He is currently a Professor with the Hebei University of Technology, China, and an Awardee of 100 Talents Plan of Hebei Province, China. He has published dozens of papers in top AI or database conferences, e.g., AAAI, IJCAI, ICDE, EDBT, and CIKM. His research focuses on topics within artificial intelligence and data mining, especially deep learning, medical artificial intelligence, health data mining, computing vision, and smart grids.