IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE TECHNOLOGIES FOR ELECTRIC POWER SYSTEMS

Received March 19, 2020, accepted March 28, 2020, date of publication March 31, 2020, date of current version April 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984722

Ensemble Residual Networks for Short-Term Load Forecasting

QINGSHAN XU, XIAOHUI YANG, AND XIN HUANG
College of Information Engineering, Nanchang University, Nanchang 330031, China

Corresponding author: Xiaohui Yang (yangxiaohui@ncu.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 51765042, Grant 61463031, Grant 61662044, and Grant 61862044, and in part by the Educational Reform Project of Jiangxi Provincial Department of Education under Grant JXYJG-2017-02 and Grant JXJG-18-1-43.

ABSTRACT In this paper, we propose a new ensemble residual network model for short-term load forecasting (STLF). This model improves the accuracy of short-term load forecasting (24 hours in advance). The model has a two-stage network structure. First, the different fully-connected layers are combined, and the combined structure is similar to a recurrent neural network (RNN). Features obtained from historical load data are input to the first stage of the model to get preliminary prediction results. The second stage of the model is a modified residual network, and the final predictions are output from here. We use the ensemble snapshot model with learning rate decay to improve the generalization capability of the model. The model proposed in this paper was trained and tested on two public datasets. Numerical testing shows that the proposed model can get better forecasting results in comparison with other methods, and the ensemble method adopted effectively improves the generalization ability of the model.

INDEX TERMS Load forecasting, deep learning, residual network, ensemble, learning rate decay.

I. INTRODUCTION

Load forecasting is a critical task in the energy field. Accurate forecasting results enables useful support for the optimal pricing strategies, seamless integration of renewables, and reduce the maintenance costs of power systems. The short-term load forecast has a forecast range of one hour ahead up to one week ahead [1]. With the development of the electricity market and the smart grids, load forecasting has become more critical. However, the power load is affected by multiple external factors, such as temperature, weather, seasonal characteristics, and so on. Many forecasting methods for load forecasting have been proposed to express the non-linear characteristics of load forecasting in recent years. For example, In the early stages, autoregressive moving average models (ARMA) [2] were used in load forecasting. And support vector machines (SVM) [3], [4], multi-objective algorithm [5], fuzzy-logic approach [6], have been reported in the literature. Artificial neural networks (ANN) are also often used to build STLF systems. Some methods have achieved good results by considering sophisticated techniques. Such as wavelet neural networks [7], [8], echo state network (ESN) [9], extreme learning machines (ELM) [10], and radial basis function (RBF) neural networks [11], etc. However, as the scale of neural networks increases, the problems of gradient explosion and gradient disappearance are challenging to solve, and the overfitting of neural networks becomes more serious. Therefore, for neural network architecture, the number of hidden layers will not exceed ten in general, which limits the performance of the models. In recent years, deep learning has gradually emerged. Various neural network structures, including Convolutional Neural Networks (CNN) [12], [13] and Gated Recurrent Unit (GRU) [14], have made a massive impact on image recognition, speech recognition, machine translation, and other fields. At the same time, there are various novel methods to help researchers effectively train the model to avoid the disappearance of gradients or severe overfitting. The application of deep learning in power load forecasting problems has attracted researchers' attention [15]–[20]. Reference [15] proposed a non-residential load prediction framework based on a multi-sequence LSTM recurrent neural network. The method successfully captures the dependencies between these sequences. A Seq2seq short-term load forecast model based on LSTM is developed in [16]. Two deep learning methods were proposed for electric load forecasting in [17]. Two methods, time-dependent convolutional neural

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

network (TD-CNN) and cycle-based LSTM (C-LSTM) network, significantly reduce the computational complexity. Reference [18] proposed a deep learning method based on empirical mode decomposition (EMD), which combines the EMD method with the LSTM network model. Reference [19] researched the application of deep neural network (DNN) in real load data sets, and different combinations of activation functions are used to make accurate load prediction. Reference [20] proposed a framework based on LSTM-RNN to predict the short-term residential load, which can accurately predict the load of a single household. In our model, we are not merely stacking a large number of hidden layers, which will cause severe overfitting of the model. Reference [21] proposed a residual network. This method makes the application of deep neural networks a reality. Reference [22] proposed a modified residual network, and the input of the network was replaced from the output of the previous layer by the average value of the multilayer output. We use the modified residual network to construct a model for STLF. At the same time, to improve the robustness of the prediction model and overfitting, we adopted an ensemble model of neural networks. References [23]–[26] prove that the ensemble STLF model has advantages with the single model in terms of accuracy and robustness. In most cases, integrated neural network models include multiple separate models whose average output is used as the final output. The disadvantage is that it requires a lot of computing power. The ensemble method proposed by [27], [28] only needs one training and does not need to train multiple models independently.

We proposed an STLF model based on ensemble residual network. The work of this paper is as follows: Different parameters that can influence the electricity load are considered. These parameters include past values of the electricity load as well as weather parameters and time-related information. The proposed model can be divided into a two-stage model. The first stage model is called the basic structure. The input of the basic structure is not only derived from the inputs of the current time step but also depends on the output of the previous time step, which helps the basic structure to adjust the hourly output automatically. The second stage model is the residual network. The residual network can effectively improve the performance in the deep neural network (DNN). The preliminary prediction results obtained from the basic model, and we get the final forecasting results through the residual network. Finally, to improve the robustness of the proposed model, the snapshot ensemble with learning rate decay is used. This ensemble method effectively enhances the prediction accuracy and generalization ability of the proposed model without spending extra computing power. The experimental simulation shows that the proposed approach generates minimal forecasting error compared to other approaches.

The rest of the paper is organized as follows: In the second part, we construct the input features of the model and the two stages of the model, and also introduces the ensemble strategy. In the third section, the proposed model is tested on two datasets: ISO New England data and North American electric utility data, and we providing test results and comparisons with other accepted methods. Section IV summarizes the conclusions.

TABLE 1. Inputs for the proposed model of the *h*th.

Input	Size	Description of the Inputs
D_h^{hour}	24	Loads of the recent 24 hours before the hth hour
T_h	4	The temperature value of the hth hour
S_h	4	One-hot encoding for seasons of the hth hour
W_h	2	One-hot encoding for weekday/weekend of the hth hour
F_h	2	One-hot encoding for holidays of the hth hour
D_h^{day}	7	Loads of the hth hour of every day of the week
T_h^{day}	7	The temperature of the hth hour of every day of the week
M_h^{day}	1	Load average value of the hth hour of every day of the week
D_h^{week}	8	Loads of the hth hour of the days that are 1, 2, 3, 4, 5, 6,7, 8 weeks prior to next day
T_h^{week}	8	The temperatures of the hth hour of the days that are 1, 2, 3, 4, 5, 6,7, 8 weeks prior to next day
M_h^{week}	1	The average value of D_h^{week}
D_h^{month}	3	Loads of the hth hour of the days that are 4, 8,12 weeks prior to next day
T_h^{month}	3	The temperature of the hth hour of the days that are 4, 8,12 weeks prior to next day

II. ENSEMBLE RESIDUAL NETWORK

In this paper, we propose a short-term load forecast based on an ensemble residual network. The ensemble residual networks model consists of two stages. The first stage, composed of a fully-connected layer, and the second stage is the modified residual network. In recent years, feature engineering has drawn a lot of attention. The results obtained by a model depend not only on the model itself but also on the characteristics of the data itself. Load data is a typical time series, and it also has non-linear characteristics. And it is difficult to strike a balance with traditional methods. We detach the time window from the historical load data and obtain the past data as input to the network. The hourly load values predicted in the first stage of the model were combined to obtain a preliminary 24-hour load forecast. The output of the basic structure is the input to the next model. Finally, the ensemble method enables to enhance the generalization ability of the model.

A. FEATURE ENGINEERING

Considering the time series characteristics of load forecasting, we not only consider the short-term laws but also construct the characteristics that can reflect the long-term trend [29]. P_h is the prediction result of the hth hour of the next day. The inputs used to get P_h are listed in Table 1. We normalize the load and temperature of the datasets, both of which are divided by their respective maximums.

T_h is the temperature of the hth hour, S_h, W_h and F_h respectively represent the hth hour season, weekday/weekend, and holiday. We use one-hot encoding when processing these features. D_h^{hour} is the load data of the most recent 24 hours before the hth hour. Note that the model cannot obtain the real data for the prediction day. At $h \neq 1$, D_h^{hour} is a combination of historical data and the data for the day predicted by the model. D_h^{day}, T_h^{day}, M_h^{day} are the load, temperature, and average value corresponding to the hour within seven days before the hth hour. To provide the long-term load characteristics, D_h^{week} represents the historical load of the hth hour of the days (the day of the same day-of-week index as the next day) in the first eight weeks. T_h^{week} are the temperature values of the same hours as D_h^{week}. M_h^{week} is the average of D_h^{week}. And we also get D_h^{month}, T_h^{month} which represents the long-term load and temperature trends for three months.

We not only use the current hour's information but also build some features from historical load data. We hope that these characteristics can help the model capture the nonlinear characteristics of the time series.

B. BASIC STRUCTURE

The first stage of the model is called the basic structure. The prediction model for one hour in basic structure is shown in Figure 1. We are not merely copying the prediction result for one hour as the output, but using the prediction result as the basic structure input for the next hour, which is similar to the RNN (Recurrent Neural Network. The input of the model depends on the input at this time and the output of the last time step. The model can adjust the predicted value every hour automatically. The neurons inside basic structure are different from GRU/LSTM. Weights and bias are not shared in basic structure, which is different from GRU/LSTM. Except for the first hour of prediction, the input of the model for other hours has the prediction results of the previous hour. We expect the model to learn different features from each hour. For the load forecast one day ahead, there are 24 basic structures. $OutPre_1$ is the combination of the output of D_h^{hour} through a fully-connected layer and the output of $[S_h; W_h; F_h; M_h^{day}; M_h^{week}]$ through a fully-connected layer. For D_h^{day}, T_h^{day}, D_h^{week}, T_h^{week}, D_h^{month}, T_h^{onth}, we concatenate the pairs $[D_h^{day}; T_h^{day}]$, $[D_h^{week}; T_h^{week}]$ and $[D_h^{month}; T_h^{onth}]$, and connect them with three separate fully-connected layers. Then we combine the output of $[S_h; W_h; F_h; M_h^{day}; M_h^{week}]$ after a fully-connected layer with the three outputs, and get $OutPre_2$ through a layer of full connection. Finally, we connect $OutPre_1$, $OutPre_2$ and T_h with a fully-connected layer, the output is the preliminary one-hour prediction result. The activation function of the fully-connected layer after $[S_h; W_h; F_h; M_h^{day}; M_h^{week}]$ is Leaky-ReLU. The activation function of other fully-connected layers is SELU [30].

Using ReLU [31] as an activation function can effectively improve the effect of deep neural networks. RELU is given by

$$\mathrm{ReLU}\,(y_i) = \max\,(0, y_i) \tag{1}$$

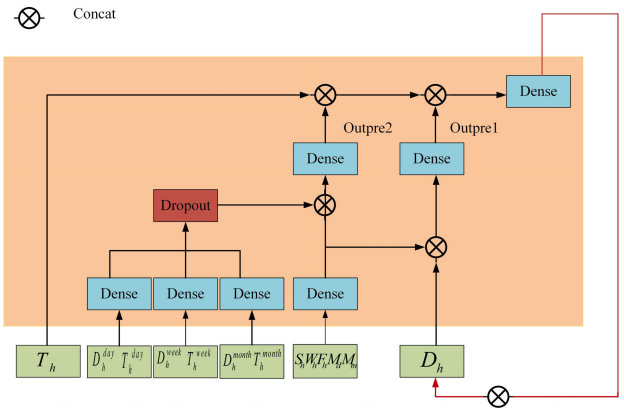

FIGURE 1. The basic structure for one-hour load forecast.

where y_i is the output of the ith node of the layer. The disadvantage of ReLU is that if the output of a neuron is 0, the gradient of the neuron will fail to update the weight of the neuron, so the neuron will never be activated. When there are a large number of inactive neurons in the network, the convergence of the model becomes very difficult. There are two activation functions: Leaky-ReLU (LReLU) and SELU are used in the basic model. They all improve the ReLU function. LReLU adds a slope to the negative semi-axis of ReLU. Neurons can still update weights when the outputs of neurons are less than zero. LReLU is defined as

$$\mathrm{LReLU}\,(y_i) = \begin{cases} y_i & \text{if } y_i > 0 \\ \alpha y_i & \text{if } y_i < 0 \end{cases} \tag{2}$$

where α is fixed. In this paper, the value α is the default value, 0.3. LReLU avoids the disadvantage that the gradient of the neuron cannot be updated when the activation value is less than 0 through simple modification. SELU further modified ReLU to induce self-normalizing properties, where SELU is defined as

$$SELU\,(y_i) = \lambda \begin{cases} y_i & \text{if } y_i > 0 \\ \beta e^{y_i} - \beta & \text{if } y_i < 0 \end{cases} \tag{3}$$

where λ and β are two tunable parameters. The author proposed to choose $\lambda \approx 1.0507$ and $\beta \approx 1.6733$ in [30], the output of the fully connected layer network is also close to the standard normal distribution when the input data conforms to the standard normal distribution, which helps the gradient not explode or disappear.

C. RESIDUAL NETWORKS

In [21], a new method for constructing deep neural networks was proposed. In traditional neural networks, the principal formulas of neural networks are as follows

$$z_x = \rho(x) \tag{4}$$

where x is the input of the neuron, $\rho(x)$ represents the calculation inside the neuron and z_x is the output of the neuron. When training a deep neural network to complete

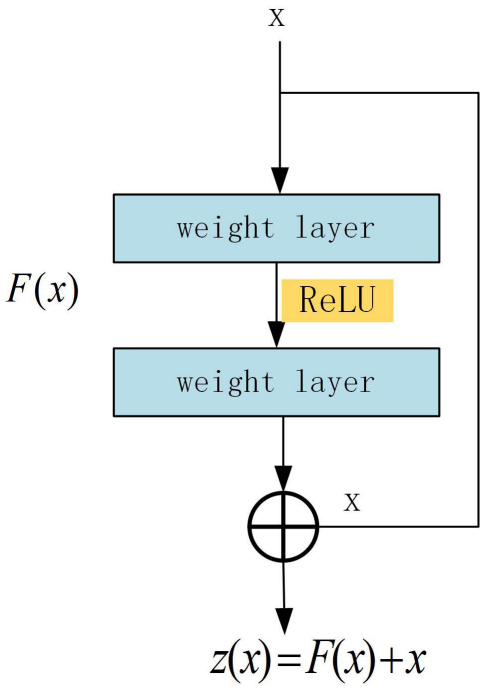

FIGURE 2. Residual block.

a task, two problems occur with deep neural networks: gradient explosion/vanish and network degeneration. The SELU can alleviate the problem of gradient explosion/vanish. Network degeneration means that as the depth of the network increases, the performance of the network gradually increases to saturation and then decreases rapidly. Residual networks can solve the problem of network degeneration. Residual networks construct deep neural network structures through residual blocks. A residual block is illustrated in Figure 2.

In the residual block, the mapping from x to $F(x) + x$ replaces the mapping from x to $\rho(x)$. If n residual blocks are stacked together, the forward propagation formula is defined as

$$z(x) = x_0 + \sum_{i=1}^{n} F\left(x_{i-1}\right) \tag{5}$$

where x_0 is the initial input of the network. In fact, the residual block constructs an artificial identity map that directly adds the input of the neural unit to the output of the neural unit. Experiments show that the residual block solves the degradation problem of deep neural networks well. Reference [32] gives an explanation of the residual block from the perspective of forwarding propagation and backward propagation. Reference [22] proposed an improved deep residual network (ResNetPlus). Each layer includes a main residual block and some side residual blocks. Average with the output of the side residual blocks and the output of each main residual block. The average value is the input of all main residual blocks in the subsequent layers.

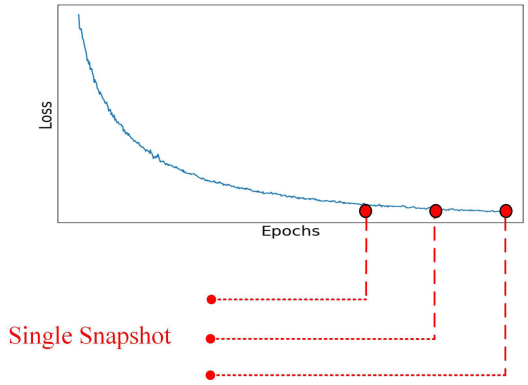

FIGURE 3. Save snapshot points when the model is about to converge.

D. THE ENSEMBLE STRATEGY

Ensemble strategies are widely used in machine learning. The ensemble results and generalization capability of multiple models are better than single models. The traditional ensemble method is to train various models with different initial weights. The results of the models converge to different local optimums. Finally, these different models are used to vote or average. The disadvantage of this ensemble method is that the training cost is too high. To make the performance of the ensemble model good, it is often necessary to train a lot of weak models. The author in [27] proposed a snapshot ensemble (Snapshot ensembling). This ensemble method uses a cyclic learning rate plan and only needs one training to obtain models that converge to multiple different minimum values. Reference [28] proposed the Fast Geometric Ensembling (FGE). FGE is very similar to Snapshots Ensembling. But they differ in two main ways. First, FGE uses linear piecewise periodic learning rate planning instead of cosine annealing. Second, the cycle length of the FGE is much shorter. FGE can improve model performance without consuming too many computing resources. The ensemble strategy is illustrated in Figure 3 in this paper.

We use Adam [33] (adaptive moment estimation) as the optimization method of the model. Since Adam has an adaptive learning rate, we do not follow the linear piecewise periodic learning rate planning or cosine annealing in [27], [28]. As the training progresses, the learning rate should gradually decrease. Use larger learning speeds at the beginning to speed up the best solution, and later use lower learning speeds to improve stability to avoid skipping the best solution. Save the model at this time, and multiply the learning rate of the model by a constant value[1] when the model runs to the snapshot point, and continue to run until the next snapshot point. After getting all the snapshot models, we average the output of the models and produce a final prediction. In this paper, a total of 7 snapshot points are saved,[2] and the final output depends on the average output of these seven snapshot points.

[1]The constant value in this article is 0.7.

[2]The model saves snapshot points when running 2000, 3000, 3500, 4000, 4500, 5000, 5500 epochs.

Finally, the MAPE (Mean Absolute Percentage Error) is the optimization goal of the model. MAPE is given by

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{6}$$

where m is the number of samples, y_i is the actual value of the sample, and $\hat{y}_i$ is the predicted value of the model. In this paper, we use the five commonly used evaluation metrics (including MAPE) to evaluate the performance forecasting models.

Root of mean squared error:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{7}$$

Mean absolute error:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i| \tag{8}$$

Normalized mean squared error:

$$NMSE = \frac{1}{\Delta^2 m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$
$$\Delta^2 = \frac{1}{m-1}\sum_{i=1}^{m}(y_i - \bar{y})^2 \tag{9}$$

Pearson's correlation coefficient:

$$R = \frac{\sum_{i=1}^{m}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{m}(y_i - \bar{y})^2 \sum_{i=1}^{m}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{10}$$

III. RESULTS AND DISCUSSION

In this section, the proposed model is compared with other methods based on two public data sets: ISO New England dataset[3] and North American electric utility dataset[4]. And the proposed method is tested using the actual load and temperature data. The two power companies have significant differences in power scale and temperature. And we evaluate the impact of weather forecast temperature error on model performance. The optimizer's initial learning rate is 0.001. All programs were conducted in *Tensorflow* 1.14 and *Keras* 2.24 based on Python 3.7. The model takes about four hours to train the 5500 generation on a personal computer with Intel i7-9750H and NVIDIA 1660ti. To evaluate the forecasting performance, we use MAPE as the error metric.

[3]Available at http://www.iso-ne.com/isoexpress/web/reports/pricing/-/tree/zone-info

[4]Available at http://sites.google.com/site/fkeynia/loaddata

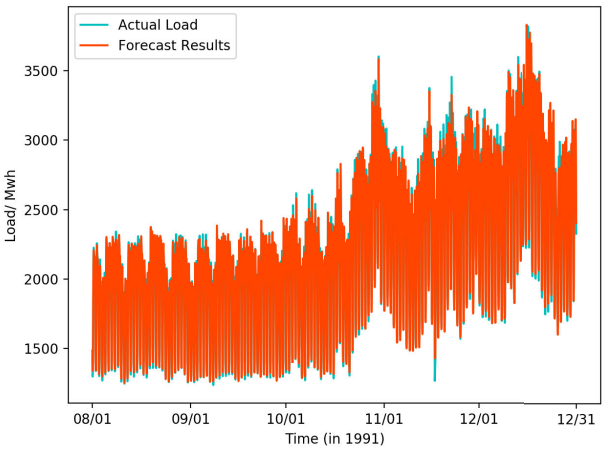

FIGURE 4. Forecast results and actual load from August to December in 1991.

A. THE PROPOSED MODEL TEST ON NORTH AMERICAN ELECTRICITY UTILITY DATASET

The first case uses the hourly data from North American electricity utility data. Collected data of this case covers January 1, 1985, to October 12, 1992. The data of the two-year period before October 12th, 1992, is used as the test set. Figure 4 shows the comparison between the forecast values of the proposed model from August 1 to December 31 in 1991 and the actual load data. In August and September, the model can well fit the real data distribution when the load is relatively stable. In November, the prediction accuracy decreased when the load fluctuated greatly, but the predicted value also reported the trend of the load. Figure 5 shows the distribution of the difference in the whole year of 1991. From Fig. 5, it can be easily illustrated that the median of the absolute difference between the actual and forecast value of each month is close to zero. In August and September, the forecast value was closest to the real load. There are some points with significant differences in November, which is consistent with the results in Figure 4.However, most of the differences between the forecast value and actual load are in the range $[-400, 400]$ in general.

We compare the results of the other methods in the same test set. In [34], a new hybrid prediction method was proposed, which mainly consisted of wavelet transform, neural network, and evolutionary algorithm. In [35], a novel load signal extension scheme was proposed. The advantage of this scheme is to deal with the border distortion problem. Reference [36] proposed a new method for day-to-day load forecasting. In [37], a parallel model was proposed, which consisted of 24 support vector machines. In [38], a method based on wavelet transform, extreme learning machine, and improved bee colony algorithm was proposed for STLF. In [39], to take into account both feature selection and parameter optimization, a method based on learning particle swarm optimization was proposed.

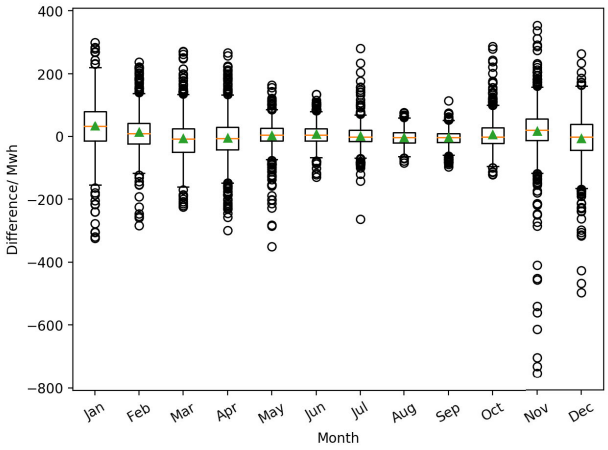

FIGURE 5. Distribution of difference in 1991.

TABLE 2. MAPE values (%) for North American electricity utility data.

Model	Actual temperature	Noisy temperature
[34]	2.64	2.84
[35]	2.04	-
[36]	2.37	2.53
[37]	1.99	2.03
[38]	1.87	1.95
[39]	1.80	1.85
GRU	4.58	4.67
LSTM	4.52	4.59
Basic Structure	3.24	3.29
Basic Structure+GRU	2.45	2.50
Basic Structure+LSTM	2.65	2.71
Proposed	**1.76**	**1.80**

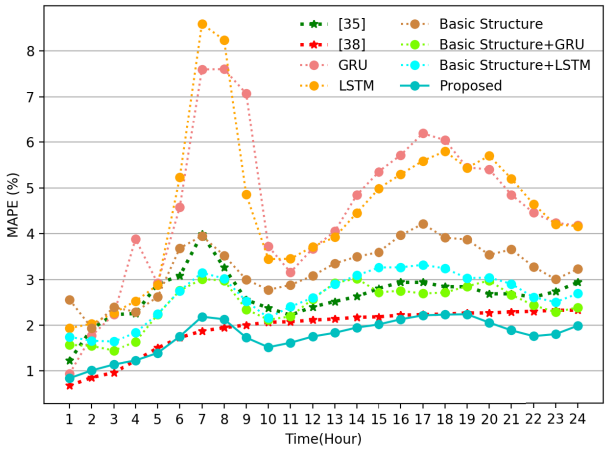

FIGURE 6. Hourly MAPE results of the forecasting models.

To evaluate the effect of weather forecasting error, we add the Gaussian noise with zero mean and standard deviation of 1°C to the temperature. The results of all models are shown in Table 2. It can be seen that the results obtained

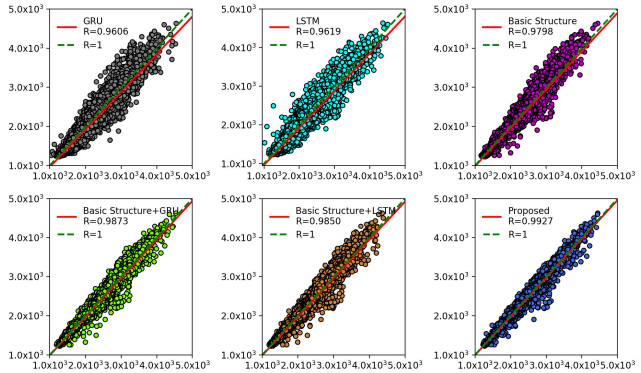

FIGURE 7. Relationship between actual load and forecasted load by six models for the NA dataset.

by the basic structure are better than GRU/LSTM under real temperature. And the MAPE obtained by the proposed model (Basic Structure + ResNet) is only 1.76, which is better than other methods. The residual network performs better than GRU/LSTM in deep neural networks. After adding noise, the proposed model outperforms the existing methods. The forecast results of the proposed method and the other methods with real temperatures at different forecast horizons are illustrated in Fig.6. For some papers that do not have this data, we only draw the data of [35], [38]. The input of the model depends on the previous 24 hours. For the later hours of the day, the model cannot get the accurate load value of the 24 hours, which will cause the accumulation of errors. At 1 am, the load data of the previous 24 hours of the model are all real data, and the model has the best performance. In 7 hours, the deviation reached the maximum, at which time MAPE was equal to 2.24. For anyone hour, this method is better than the other methods, and the MAPE is higher than the result in [38] only at the 1, 2, 7, and 8 am. Fig.7 shows the relationship between actual load and predicted load by all six models in which the proposed model shows a strong relation.

B. THE PROPOSED MODEL TEST ON ISO NEW ENGLAND DATASET

The second task is to estimate the generalization ability in various cases of the proposed model. In the section, a detailed analysis of the model's generalization capabilities is reported. ISO New England data covers data from March 1, 2003, to December 31, 2014. We did not adjust the hyper-parameters for this dataset, and the model is the same as when the mpdel run in North American electricity utility data.Hourly data set from years 2004–2007 and out-of-sample data from the years 2008 and 2009 are used for training and testing purposes. The results of different methods on the test set are shown in Table 3. Reference [5] proposed a novel multi-objective algorithm (MOFTL) based on Follow The Leader algorithm and comparing the results with three newly multi-objective algorithms. In [40], a method for augmented neural networks is proposed, which includes a set of iteratively trained artificial neural networks. Refer-

TABLE 3. Forecasting results of the models for ISO New England dataset.

Model	MAPE (%)	RMSE (Mwh)	MAE (Mwh)	NMSE (Mwh)	R
[5]	3.07	594.3	458.16	0.0442	0.9741
[40]	1.79	-	-	-	-
[41]	1.75	-	-	-	-
[42]	1.55	-	-	-	-
[43]	1.54	-	-	-	-
[44]	-	651.8	458.4	0.0470	-
GRU	4.45	953.8	664.8	0.1181	0.9393
LSTM	4.23	920.0	637.2	0.1098	0.9436
Basic Structure	2.61	584.1	392.9	0.0443	0.9778
Basic Structure+GRU	2.16	485.9	324.5	0.0307	0.9846
Basic Structure+LSTM	2.07	471.5	309.4	0.0289	0.9855
Proposed	**1.50**	**346.3**	**227.69**	**0.0156**	**0.9925**

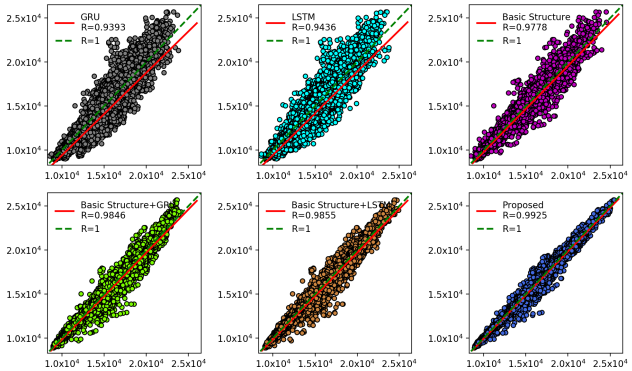

FIGURE 8. Relationship between actual load and forecasted load by six models for the ISO-NE dataset.

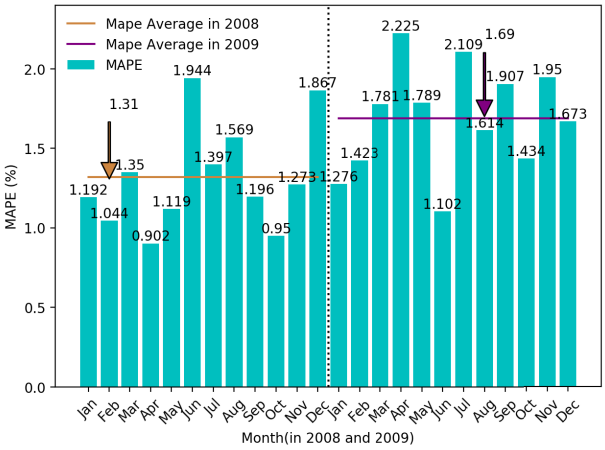

FIGURE 9. MAPE results for the ISO New England dataset in 2008 and 2009.

TABLE 4. MAPE results for ISO New England dataset in 2010 and 2011.

Model	2010	2011
[45]	1.80	2.02
[46]	1.75	1.98
[47]	1.50	1.80
[22]	1.50	1.64
Proposed	**1.49**	**1.61**
Proposed(10 extra months)	**1.46**	**1.54**

ence [41] present a load forecasting scheme using bagged neural networks that improves load forecasting accuracy. Reference [42] proposed method is based on a hybrid artificial intelligence system, a machine learning method combined with a trained rather simple neural network can get a more accurate solution. Reference [43] proposed to use bagged-boosted artificial neural networks for load prediction. In [44], a novel evolutionary algorithm based on follow the leader concept is developed and the proposed algorithm is integrated with neural network. The results obtained for the year 2008 and 2009 by all the forecasting models are shown in Table 3. It clearly shows that basic structure brings higher accuracy than GRU/LSTM in terms of all performance evaluation metrics mentioned earlier. The values of MAPE-1.50%, RMSE-346.3MWH, MAE-227.69MWH, NMSE-0.01560MWH, Pearson correlation coefficient (r)-0.9925 obtained by the proposed model makes it more reliable forecasting model. Fig8. shows the relationship between actual load and predicted load by all six models in which the proposed model shows a strong relation.Fig 9 shows monthly MAPE of the proposed model for the years 2008 and 2009. The comparison between the predicted load and the actual load of April 2008 and April 2009 is shown in Fig 10. The MAPE for April 2008 is the smallest of all months, and the MAPE for April 2009 is the largest of 24 months. The comparison of load obtained after forecast by six models with the actual load is shown in Fig 11. This comparison illustrates that the proposed model achieves the most accurate prediction values, and also load consumption during weekends is less compared to weekdays.We did not adjust any hyper-parameters on the ISO New England dataset. It can be seen that the proposed model has good generalization ability on different datasets.

We further estimate the performance of the proposed model on the dataset of 2010 and 2011. The data from 2004 to 2009 is used to train the proposed model. Table 4 shows the results in [22], [45]–[47]. These methods also use the data from the previous 5 years of the test set as the training set. With the same training set, the proposed model is better than the existing model in the 2010 and 2011 test sets. We have not modified any hyper-parameters here. The model is the same as when predicting North American electricity utility data. Besides, we added an extra 10-month sample number (the dataset start date is March 1, 2003), and the forecast results show that the proposed model performs better with additional training samples.

We have been using actual temperature values as input in this dataset. The results shown previously represent the upper

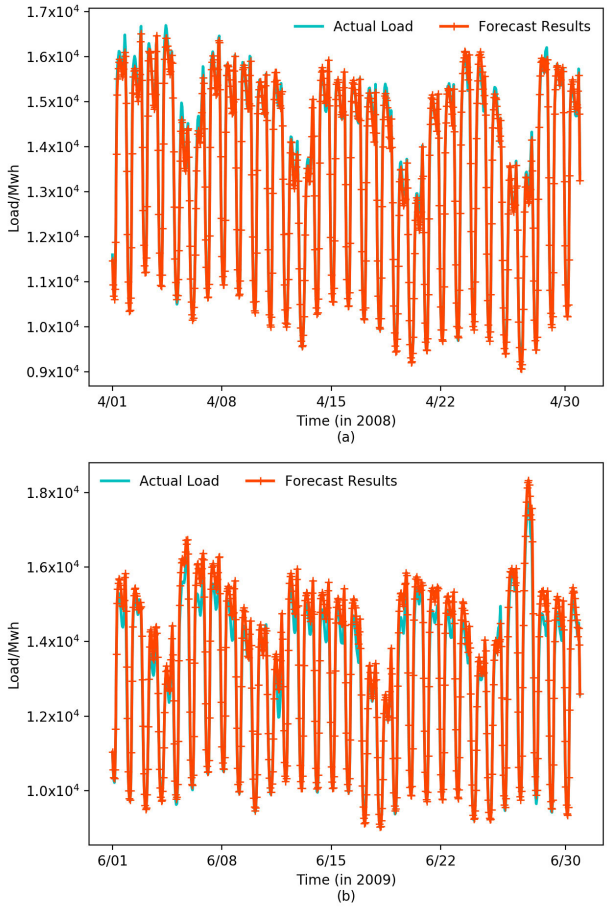

FIGURE 10. Forecast results of a month in April 2008 and June 2009.

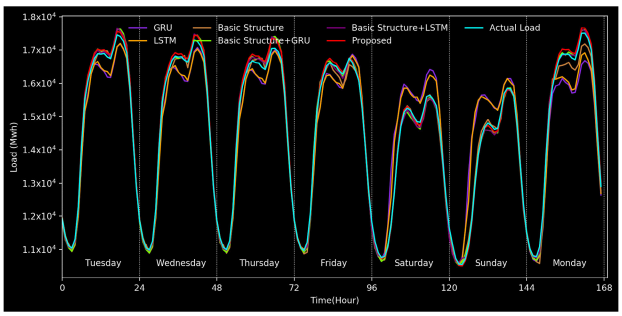

FIGURE 11. Comparison of actual load with forecast load by six models.

bound of the performance of the proposed model. We consider the effect of temperature error on STLF for ISO New England dataset and examine the performance of a single model and ensemble model in the face of temperature noise. To cover a wide range of temperature errors during weather forecasting, we consider a set of Gaussian noise with different means and standard deviations. All subsequent cases were repeated five times, and average the results. After adding different noise to the temperature, the MAPE values obtained by the two models are shown in Figure 12. It can be seen in Figure 12 that the proposed model (an ensemble model

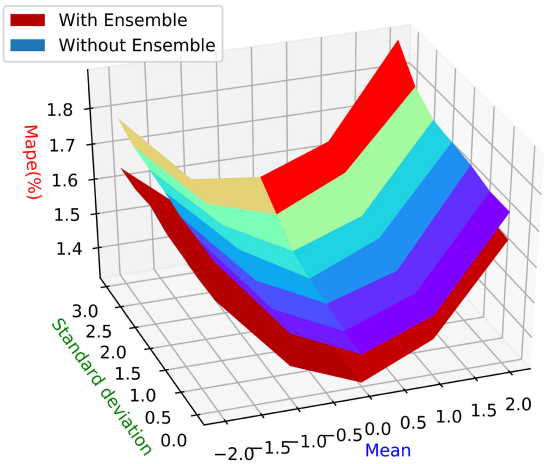

FIGURE 12. MAPE due to different Gaussian noises: means = (−2, −1, 0, 1, 2) and standard deviations = (0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0).

TABLE 5. Forecast results with zero-mean Gaussian noises.

Standard deviation	Temperature Error Range(°F)	Error Increment	Error Increment without Ensemble
0	-	0%	0%
0.5	$[-2.1, 2.2]$	0.15%	0.53%
1.0	$[-4.6, 4.5]$	0.49%	1.8%
1.5	$[-7.65, 6.33]$	1.98%	2.7%
2.0	$[-8.24, 9.23]$	2.62%	4.23%
2.5	$[-11.0, 10.51]$	3.09%	7.77%
3.0	$[-12.5, 12.3]$	4.56%	11.55%

of 7 single snapshot models) performs better than the single snapshot model under any deviation. Note that compared with the larger variance, the larger the noise of the mean will bring greater load prediction error.

We follow the method described in [34] for modifying the temperature value and consider the impact of the ensemble method when the mean is 0. The MAPE value (1.325) obtained with the actual temperature serves in 2008 as the reference (the MAPE results of the single snapshot model is 1.40 at the actual temperature serves in 2008). We compare the results of the proposed model with a single snapshot model, which is trained in the same epochs. It can be seen from Table 5, and the ensemble model dramatically reduces the impact of temperature noise. When the temperature error changes in the maximum interval $[-12.5, 12.3]$, the prediction error only increases by 4.56%.

IV. CONCLUSION

This paper proposes an STLF model based on ensemble residual networks. In terms of feature selection, we use historical load data to construct features. The model has a two-

stage network architecture of the basic structure and modified resnet model. And we use the ensemble model with learning rate decay to enhance model performance. The proposed model is tested on two public datasets. The results reveal that the proposed model is superior to existing models in prediction accuracy in various test cases, and it is superior in the robustness of temperature changes. If multiple models can be integrated based on the ensemble model, the model performance may be better. Since we only covered some of the latest technologies of deep neural networks, we did not use other deep neural network building blocks (such as CNN or Seq2seq). In the next step, we will try to apply it to STLF for comparison with existing methods.

REFERENCES

[1] E. Almeshaiei and H. Soltan, "A methodology for electric power load forecasting," *Alexandria Eng. J.*, vol. 50, no. 2, pp. 137–144, Jun. 2011.

[2] J.-F. Chen, W.-M. Wang, and C.-M. Huang, "Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting," *Electr. Power Syst. Res.*, vol. 34, no. 3, pp. 187–196, Sep. 1995.

[3] Y.-M. Wi, S.-K. Joo, and K.-B. Song, "Holiday load forecasting using fuzzy polynomial regression with weather feature selection and adjustment," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 596–603, May 2012.

[4] S. Fan and L. Chen, "Short-term load forecasting based on an adaptive hybrid method," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 392–401, Feb. 2006.

[5] P. Singh and P. Dwivedi, "A novel hybrid model based on neural network and multi-objective optimization for effective load forecast," *Energy*, vol. 182, pp. 606–622, Sep. 2019.

[6] M. Rejc and M. Pantos, "Short-term transmission-loss forecast for the slovenian transmission power system based on a fuzzy-logic decision approach," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1511–1521, Aug. 2011.

[7] Y. Chen, P. B. Luh, C. Guan, Y. Zhao, L. D. Michel, M. A. Coolbeth, P. B. Friedland, and S. J. Rourke, "Short-term load forecasting: Similar day-based wavelet neural networks," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 322–330, Feb. 2010.

[8] B. Li, J. Zhang, Y. He, and Y. Wang, "Short-term load-forecasting method based on wavelet decomposition with second-order gray neural network model combined with ADF test," *IEEE Access*, vol. 5, pp. 16324–16331, 2017.

[9] F. M. Bianchi, E. De Santis, A. Rizzi, and A. Sadeghian, "Short-term electric load forecasting using echo state networks and PCA decomposition," *IEEE Access*, vol. 3, pp. 1931–1943, 2015.

[10] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[11] Z. A. Bashir and M. E. El-Hawary, "Applying wavelets to short-term load forecasting using PSO-based neural networks," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 20–27, Feb. 2009.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[13] Z. Deng, B. Wang, Y. Xu, T. Xu, C. Liu, and Z. Zhu, "Multi-scale convolutional neural network with time-cognition for multi-step short-term load forecasting," *IEEE Access*, vol. 7, pp. 88058–88071, 2019.

[14] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[15] 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.

[16] G. Gong, X. An, N. K. Mahato, S. Sun, S. Chen, and Y. Wen, "Research on short-term load prediction based on Seq2seq model," *Energies*, vol. 12, no. 16, p. 3199, 2019.

[17] L. Han, Y. Peng, Y. Li, B. Yong, Q. Zhou, and L. Shu, "Enhanced deep networks for short-term and medium-term load forecasting," *IEEE Access*, vol. 7, pp. 4045–4055, 2019.

[18] J. Bedi and D. Toshniwal, "Empirical mode decomposition based deep learning for electricity demand forecasting," *IEEE Access*, vol. 6, pp. 49144–49156, 2018.

[19] T. Hossen, S. J. Plathottam, R. K. Angamuthu, P. Ranganathan, and H. Salehfar, "Short-term load forecasting using deep neural networks (DNN)," in *Proc. North Amer. Power Symp. (NAPS)*, Sep. 2017, pp. 1–6.

[20] 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.

[21] 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.

[22] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3943–3952, Jul. 2019.

[23] J. W. Taylor and R. Buizza, "Neural network load forecasting with weather ensemble predictions," *IEEE Power Eng. Rev.*, vol. 22, no. 7, p. 59, Jul. 2002.

[24] R. E. Abdel-Aal, "Improving electric load forecasts using network committees," *Electr. Power Syst. Res.*, vol. 74, no. 1, pp. 83–94, Apr. 2005.

[25] M. De Felice and X. Yao, "Short-term load forecasting with neural network ensembles: A comparative study [application notes]," *IEEE Comput. Intell. Mag.*, vol. 6, no. 3, pp. 47–56, Aug. 2011.

[26] M. Alamaniotis, A. Ikonomopoulos, and L. H. Tsoukalas, "Evolutionary multiobjective optimization of kernel-based very-short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1477–1484, Aug. 2012.

[27] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get M for free," 2017, *arXiv:1704.00109*. [Online]. Available: http://arxiv.org/abs/1704.00109

[28] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of DNNs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8789–8798.

[29] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst. (GIS)*, 2016, p. 92.

[30] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 971–980.

[31] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8609–8613.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 630–645.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[34] A. J. Rochareis and A. P. AlvesdaSilva, "Feature extraction via multiresolution analysis for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 189–198, Feb. 2005.

[35] N. Amjady and F. Keynia, "Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm," *Energy*, vol. 34, no. 1, pp. 46–57, Jan. 2009.

[36] A. Deihimi and H. Showkati, "Application of echo state networks in short-term electric load forecasting," *Energy*, vol. 39, no. 1, pp. 327–340, Mar. 2012.

[37] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4356–4364, Nov. 2013.

[38] S. Li, P. Wang, and L. Goel, "Short-term load forecasting by wavelet transform and evolutionary extreme learning machine," *Electr. Power Syst. Res.*, vol. 122, pp. 96–103, May 2015.

[39] Z. Hu, Y. Bao, and T. Xiong, "Comprehensive learning particle swarm optimization based memetic algorithm for model selection in short-term load forecasting using support vector regression," *Appl. Soft Comput.*, vol. 25, pp. 15–25, Dec. 2014.

[40] A. S. Khwaja, X. Zhang, A. Anpalagan, and B. Venkatesh, "Boosted neural networks for improved short-term electric load forecasting," *Electr. Power Syst. Res.*, vol. 143, pp. 431–437, Feb. 2017.

[41] A. S. Khwaja, M. Naeem, A. Anpalagan, A. Venetsanopoulos, and B. Venkatesh, "Improved short-term load forecasting using bagged neural networks," *Electr. Power Syst. Res.*, vol. 125, pp. 109–115, Aug. 2015.
[42] S. Brodowski, A. Bielecki, and M. Filocha, "A hybrid system for forecasting 24-h power load profile for polish electric grid," *Appl. Soft Comput.*, vol. 58, pp. 527–539, Sep. 2017.
[43] A. S. Khwaja, A. Anpalagan, M. Naeem, and B. Venkatesh, "Joint bagged-boosted artificial neural networks: Using ensemble machine learning to improve short-term electricity load forecasting," *Electr. Power Syst. Res.*, vol. 179, Feb. 2020, Art. no. 106080.
[44] P. Singh and P. Dwivedi, "Integration of new evolutionary approach with artificial neural network for solving short term load forecast problem," *Appl. Energy*, vol. 217, pp. 537–549, May 2018.
[45] H. Yu, P. D. Reiner, T. Xie, T. Bartczak, and B. M. Wilamowski, "An incremental design of radial basis function networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1793–1803, Oct. 2014.
[46] C. Cecati, J. Kolbusz, P. Rózycki, P. Siano, and B. M. Wilamowski, "A novel RBF training algorithm for short-term electric load forecasting and comparative studies," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6519–6529, Oct. 2015.
[47] S. Li, L. Goel, and P. Wang, "An ensemble approach for short-term load forecasting by extreme learning machine," *Appl. Energy*, vol. 170, pp. 22–29, May 2016.

QINGSHAN XU received the B.E. degree in electrical engineering and its automation from Nanchang University, China, in 2018, where he is currently pursuing the M.S. degree in electrical engineering. His research interests include load forecasting, machine learning, and nature language processing.

XIAOHUI YANG received the B.Sc., M.Sc., and Ph.D. degrees from Nanchang University, Nanchang, China, in 2003, 2006, and 2015, respectively.

He has been with the Department of Electronic Information Engineering, School of Information Engineering, Nanchang University, since 2006, where he is currently an Associate Professor. He has published over 30 research articles. His current research interests include intelligent control, process control, fault diagnosis, and stochastic nonlinear systems.

XIN HUANG received the B.E. degree in electrical engineering and its automation from Nanchang University, China, in 2018, where he is currently pursuing the M.S. degree in electrical engineering. His research interests include energy management and load forecasting.