Multiple wavelet convolutional neural network for short-term load forecasting

Liao, Zhifang; Pan, Haihui; Fan, Xiaoping; Zhang, Yan; Kuang, Li

Published in:
IEEE Internet of Things Journal

DOI:
10.1109/JIOT.2020.3026733

Publication date:
2020

Document Version
Author accepted manuscript

Multiple Wavelet Convolutional Neural Network for Short-term Load Forecasting

Zhifang Liao, *Member, IEEE,* Haihui Pan, Xiaoping Fan, *Member, IEEE,* Yan Zhang, and Li Kuang, *Member, IEEE,*

***Abstract*—Although the accuracy of load forecasting has been studied by many works, the actual deployability of a model is rarely considered. In this work, we consider the actual deployability of a model from four aspects: the prediction performance of the model, the robustness of the model, the dependence of the model on external data and the storage size of the model. From these four aspects, we propose a multiple wavelet convolutional neural network (MWCNN) for load prediction. On two public datasets, we verified the performance performance and robustness of the MWCNN. The MWCNN only uses load data, and the storage size of the model is only 497 KB, which shows that MWCNN has good deployability. In addition, our MWCNN prediction results are interpretable. The experimental results show that the MWCNN can effectively capture the periodic characteristics of load data.**

***Index Terms*—Short-term load forecasting, convolutional neural network, wavelet reconstruction, deployability, interpretability.**

I. Introduction

SMART grid (SG), which is the intelligence of the power grid, is also an important part of the Internet of Things (IoT) [1]. Load forecasting is of great significance to the management and dispatching of SG. Generally, the amount of power generated should be as consistent as possible with the actual load demand. Therefore, accurate forecasting results will have a significant impact on power dispatching operations and management. However, the actual load demand is often affected by many factors, such as social, economic, and environmental factors, so accurate load prediction is difficult [2].

Many methods have been proposed and applied to short-term load forecasting. Early load forecasting models include linear or nonparametric regression [3], [4] and autoregressive models [5]. However, these statistical method-based model structures are usually simple and have a low load forecasting accuracy. With the maturity of expert system technology, some researchers have built expert systems to carry out load forecasting [6]. In recent years, support vector machines have been widely used in load forecasting. Researchers have improved load prediction performance by improving support vector regression (SVR) optimization techniques or SVR loss functions [7], [8]. However, as the number of training samples increases, support vector machines (SVMs) do not work well. Neural networks have attracted the attention of many researchers due to their excellent fitting capabilities. At present, many neural network-based load prediction methods have been extracted and obtained excellent results [9], [10], [11]. Benefiting from the success of deep learning [12], [13], some researchers have achieved better prediction performance by adopting advanced network structures and deep networks [14], [15], [16].

In previous studies, researchers often focused on improving the accuracy of load forecasting, but the actual deployability of the model was rarely considered. In this work, we mainly consider the deployability of the model from four aspects: the prediction performance of the model, the robustness of the model, the dependence of the model on external data and the storage size of the model. The prediction performance of the model is the main premise for the deployability of the model; that is, the more accurate the model's prediction accuracy is, the better the value that the model can create. The robustness of the model is the premise for the stable use of the model. Due to objective and uncontrollable factors such as data measurement or data recording deviations, there is usually a certain deviation between the obtained data and the actual data. This phenomenon requires that the proposed model have good robustness. That is, the slight disturbance of the model to the input should not cause too much difference in the prediction; the more external datasets the model needs to use, the more critical the model is to the deployed scenario. For example, in some closed scenarios or when the power system does not have an interface to obtain external data, these methods that require external data will not be adaptable; the less storage space the model has, the better it will be for actual deployment.

In view of the above four aspects, we first propose a feature engineering method based on wavelet reconstruction and then propose a multiple wavelet convolutional neural network (MWCNN) for load forecasting. Next, to further increase the prediction performance of the model, we propose an ensemble scheme based on multiple wavelets. Then, we verify the robustness of the model by perturbing the input to varying degrees. Finally, we present the interpretability of the prediction results of the MWCNN. The contribution of this work can be summarized into the following four parts:

- We propose a feature engineering method based on wavelet reconstruction, and based on this method, we propose the MWCNN for load prediction. The MWCNN uses only raw load series data and the storage space required for the model is only 497 KB which indicates

Zhifang Liao, Haihui Pan, Xiaoping Fan and Li Kuang are with the School of Computer Science and Engineering, Central South University, Changsha, 410083, P. R. of China. (e-mail: zfliao@csu.edu.cn; phhfly@csu.edu.cn; xpfan@mail.csu.edu.cn; kuangli@csu.edu.cn)

Yan Zhang is with School of Computing, Engineering and Built Environment, Department of Computing, Caledonian University, UK. (e-mail: yan.zhang@gcu.ac.uk)

(Corresponding author: Xiaoping Fan).

that the MWCNN has good actual deployability. On two public datasets, we validate the prediction performance of the MWCNN.

- To further improve the prediction performance of the model, we propose an ensemble scheme based on multiple wavelets. The experimental results show that the ensemble model has a better prediction performance than the single model. At the same time, we find that the ensemble model can also significantly reduce the range and standard deviation of the model's prediction bias.
- We add different degrees of disturbance to the original load data to verify the robustness of the model. We use various Gaussian distributions with different means and variances to generate 56 groups of noise and add these disturbances to the original load data. Compared to the model without added noise, the maximum increase in the MAE of the model with noise is only 4.72 and the maximum increase in the MAPE is only 0.03%, which shows that the MWCNN has good robustness.
- We use the attribution method to explain the prediction results of the MWCNN. On two public datasets, we find that the data point closest to the prediction has the largest impact on the final prediction result. The data closer to the prediction time point tend to have a greater impact on the final prediction result. Interestingly, we find that some time points with large data for prediction have periodic intervals, which shows that the MWCNN can capture the periodic characteristics in the load data.

II. Related Work

In [6], a computational intelligence method (ACO-GA) combining ant colony optimization, the genetic algorithm and fuzzy logic is proposed to build the expert system of load forecasting. The core of the expert system method lies in the construction of a related knowledge base, but the construction of a knowledge base is a time-consuming process; when new data is needed to update related rules, maintenance will become difficult, which leads to the lack of adaptability of the expert system. SVMs use the information provided by a limited sample to find an optimal compromise between model complexity and learning ability. The map the input data into a high-dimensional space, making classification and regression problems easier to solve. In [7], the feature selection algorithm was proposed for automatic model input selection, and particle swarm global optimization technology was used to optimize SVR hyperparameters to reduce the interaction between operators. In [8], an improved support vector regression method was proposed for the load-forecasting problem, and the loss function of the SVR was modified using local weighted regression. A weighted distance algorithm based on Markov distances was proposed to optimize the bandwidth of the weighting function. Although support vector machines have achieved good results in small-sample problems, as the number of training samples increases, the complexity of the model training time increases dramatically, which makes support vector machines difficult to implement in many cases. Load forecasting data usually contain both a global smooth trend and a sharp local change, that is, low-frequency and high-frequency components. A wavelet transform can effectively decompose a time series into its components, so it is an effective method to deal with nonstationary load behaviors. At present, many load prediction methods based on wavelet transforms have been proposed. In [9], a wavelet transform is used to decompose the original load sequence, and then multiple multilayer perceptrons (MLPs) are used to train and predict each decomposed component. Finally, the various subcomponents are combined to obtain the final prediction. In [10], the wavelet transform effectively decomposes the time series into its constituent parts. Each component is predicted by a combination of neural networks (NNs) and evolutionary algorithms (EAs), and then hourly load prediction is obtained by an inverse wavelet transform. In [11], a wavelet transform was used to decompose the load sequence to capture the complex features at different frequencies. Then, a combination model composed of a extreme learning machine (ELM) and the modified artificial bee colony (MABC) algorithm is used to predict each component of the load sequence. In [17], a similar daily load is selected as the input load, and it is decomposed into low-frequency components and high-frequency components by a wavelet transform. Then, the independent network is used to predict the two components of the future load. Benefiting from the development of deep learning, many researchers have improved load prediction performance by adopting more advanced network structures and building deeper networks. In [14], a deep neural network is used for load prediction. In [15], a short-term load forecasting method based on a deep residual network was proposed, and the generalization ability of the model was improved through a two-stage integration strategy. In [16], a WaveNet based on a dilated causal residual convolutional neural network (CNN) and long short-term memory (LSTM) layers was proposed for load prediction. In [18], a multilayer LSTM network was used for load prediction.

III. Method

A. Feature Engineering Based on Wavelet Reconstruction

The power load series usually contains both a global smooth trend and a sharp local change, that is, low-frequency and high-frequency components. Therefore, a wavelet transform can effectively decompose the power load series into its components. Let *L(t)* be a load sequence, which can be decomposed into

$$L(t) = \sum_{k} \phi_{j_0}(k) 2^{\frac{j_0}{2}} \varphi(2^{j_0} t - k) + \sum_{k} \sum_{j=j_0}^{\infty} \Phi_j(k) 2^{\frac{j}{2}} \psi(2^j t - k) \tag{1}$$

where $\psi(t)$ is the parent wave function, $\varphi(t)$ is the corresponding scale function, *t* is the time index, j_0 is a predefined scale, *j* and *k* are integer variables used for scaling and translation, respectively, and $\phi_{j_0}(k)$ and $\Phi_j(k)$ are approximate and detailed coefficients, respectively.

At present, many wavelet-based methods have been proposed and applied to load prediction. The core idea is to decompose the load sequence into several components and then send the components into one or more models for prediction [10], [11], [17]. Different from these existing methods,

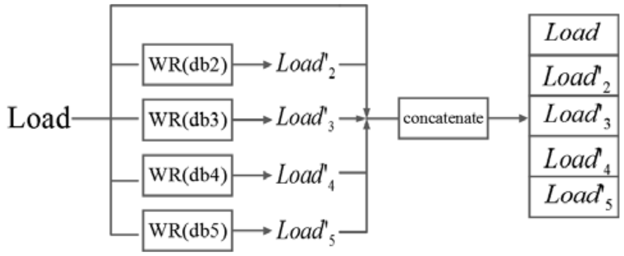

Fig. 1. Four different types of wavelets are used to reconstruct the load data, and the original load data are concatenated with four reconstructed data to obtain the final data. WR represents wavelet reconstruction, and the type of wavelet used is in brackets.

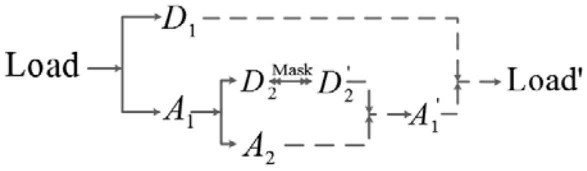

Fig. 2. Two-level wavelet decomposition and the reconstruction process. The solid line represents the wavelet decomposition process, and the dotted line represents the process of wavelet reconstruction. Mask means that the values of D_2 are all set to 0. By using different types of wavelets, we can obtain different reconstruction data.

TABLE I
ARCHITECTURE OF MWCNN

Type	Filters	Kernel size	Stride	Output size	Depth	Params
Convolution	30	1 × 8	1, 3	(5, 56, 30)	1	270
Convolution	30	2 × 2	1, 1	(5, 56, 30)	1	3630
Convolution	30	3 × 3	1, 2	(5, 28, 30)	1	8130
Convolution	30	3 × 3	1, 2	(5, 14, 30)	1	8130
Convolution	30	3 × 3	1, 2	(5, 7, 30)	1	8130
Convolution	16	2 × 2	1, 1	(5, 7, 16)	3	4016
Convolution	8	2 × 2	1, 1	(5, 7, 8)	3	1048
Convolution	4	2 × 2	1, 1	(5, 7, 4)	3	268
Convolution	1	2 × 2	1, 1	(5, 7, 1)	1	17
Avg pool	0	0	-	1	1	0

we mainly deal with the load sequence from the perspective of reconstruction, as the high-frequency components in the sequence are usually regarded as noise in the data, and more realistic and smooth data can be obtained by using the denoising reconstruction method. The whole process is shown in Figure 1. We first obtain the load data of 7 weeks before the forecast time point, i.e., 168 data points. Then, we use four different types of wavelets (db2, db3, db4, and db5) to decompose, denoise and reconstruct the load sequence to obtain four different groups of reconstruction data. Then, we concatenate the four groups of reconstruction data with the original data to obtain the final data, so the final processed data shape is 5×168. Since the level of decomposition and the method of denoising have a great influence on the reconstruction results, in the experiment, we explore the influence of the 1-3-level decomposition and soft threshold reconstruction method [19], hard threshold reconstruction method [19] and high-frequency zero reconstruction method on the load forecasting results. The specific experimental results are presented in Experiment B.

In the experiment, we find that 2-level and high-frequency zero reconstruction can achieve the best results. The processing process of this method is shown in Figure 2. First, the load sequence is decomposed into two components A_1 and D_1, where A_1 is the low-frequency component and D_1 is the high-frequency component. Next, the A_1 component is decomposed to obtain A_2 and D_2, and the high-frequency component D_2 of A_1 is set to 0 to obtain D_2'. Finally, D_2', A_2, and D_1 are reconstructed to obtain $Load'$. The difference between this method and the existing methods is that the component of the load sequence is not used as the input of the model, and we use many different types of wavelets to reconstruct the load sequence. We finally stitched together the 4 sets of reconstructed data with the original data to obtain the final data, which facilitated the introduction of the CNN.

B. Architecture of the MWCNN

Due to its good feature extraction capabilities, CNN has achieved success in many fields [20], [21]. In this paper, after reconstructing the original load sequence, we introduce a CNN to extract potential features from the reconstructed data. The design idea of the model structure mainly focuses on two points. On the one hand, the total number of parameters in the model should be as small as possible; that is, the storage space occupied by the model is small, which will be beneficial to the actual deployment of the model. On the other hand, the prediction accuracy of the model should be as high as possible, which is the premise that the model can be deployed. Research shows that the depth of the model and the structure of the model have a great impact on the final model performance [20], [22]. However, an increase in the model depth means an increase in the total number of model parameters, so we need to weigh the model's depth, that is, the total number of model parameters and the performance of the model when designing the model.

Due to the large difference between the length and width of the model input (the width is only 5, and the length is 168), the padding of all the convolution layers is set to the same value, and the height strip is set to 1 to ensure that the height of the input before and after the convolution remains the same, which makes the model deeper. The details of the architecture of the MWCNN model are shown in Table 1. Except for the final layer, which is a global pooling operation, all the other layers are convolution layers to better extract the features in the input. In the CNN model, setting the output layer to a global pooling layer rather than a fully connected layer usually achieves better generalization capabilities [23]. Another advantage of using global pooling is that it does not increase the number of training parameters. The number of model layers of the MWCNN is 16 (excluding the input layer), which ensures that the model has enough fitting ability. The total number of parameters in the final model are only 33,639, occupying only 497 KB of storage space. At the same time, the MWCNN only uses load data as input, which shows that the model is deployable in almost any scenario. The general procedures of the MWCNN is shown in Figure 3.

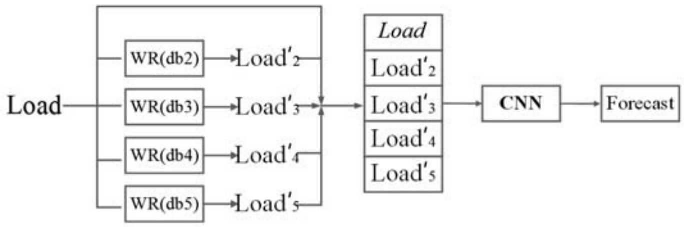

Fig. 3. The general procedures of the MWCNN. Four different types of wavelets (db2, db3, db4 and db5) are used to reconstruct the original load data. The original load data are concatenated with four reconstructed data as the input of MWCNN.

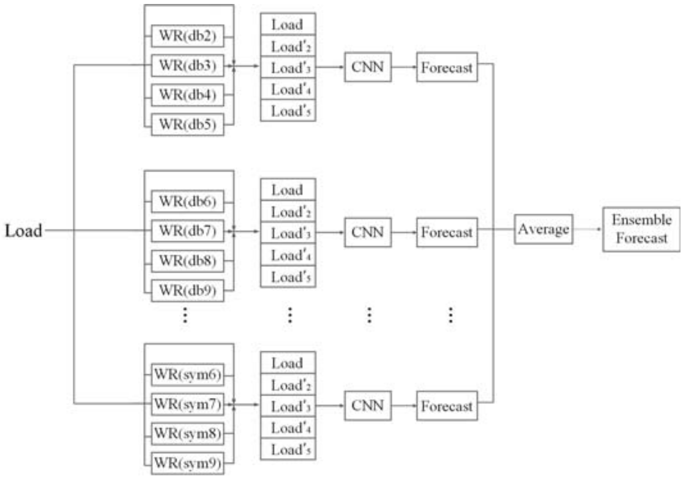

Fig. 4. Structure diagram of the integrated method based on multiple wavelet reconstruction. For each CNN, we will use different types of wavelet clusters to reconstruct the original load sequence. After all the CNNs are trained, the prediction results of the CNNs are averaged to obtain the final prediction results.

C. Ensemble Method Based on Multiple Wavelet Reconstruction

To improve the prediction performance of the model, multiple models are usually integrated to improve the prediction effect. However, using the integrated method requires two premises: the prediction performance of a single model should be as high as possible, and the differences between the models should be as large as possible. A common method is to ensure the diversity of the model as much as possible by randomly selecting the training dataset. However, this method reduces the number of training samples, so the performance of a single model will be reduced. To ensure the prediction accuracy of a single model and the differences between models, this paper proposes an integrated method based on multiple wavelets. This method uses different types of wavelets to reconstruct the original load data to obtain different model inputs. The overall structure of the method is shown in Figure 4. For each CNN, we will use multiple types of wavelets to reconstruct the original data. The load sequence is reconstructed to obtain different inputs. After training all the CNNs, we average the prediction results of each CNN to obtain the final integrated prediction result. There are two obvious benefits to using this method: first, the dataset does not need to be randomly divided, and all the datasets can be used to train the model, which can ensure that the prediction accuracy of a single model is as high as possible. Second, different wavelet clusters are used to reconstruct the original data to obtain different inputs, which can ensure the diversity of the model. In Experiment C, we find that when the integration scale is 6, the integration performance is the best. The wavelet clusters used by these 6 models are (db2, db3, db4, and db5), (db6, db7, db8, and db9), (db10, db11, db12, and db13), (db14, db15, db16, and db17), (sym2, sym3, sym4, and sym5), and (sym6, sym7, sym8, and sym9).

D. Interpretability of Load Forecasting

Although deep neural networks (DNNs) have made great achievements, they have always been regarded as a black-box method, which has caused some people to worry about the application of neural networks, and it also shows that we are not clear about the working mechanism behind neural networks. To explain the decision results of neural networks, many studies have attributed the prediction results of deep networks to the problem of their input characteristics [23], [24]. Assuming that $F(x)$ is a deep neural network and $x \in R^n$ is the input, the prediction attribution of input x to benchmark input x' is a vector $A_F(x, x') = (a_1, ..., a_n)$, where a_i is the contribution of x_i to prediction $F(x)$. In general, the prediction at the baseline should be close to zero $F(x') \approx 0$.

In [25], an attribution method of integral gradient is proposed. The integral gradient is defined as the path integral from the baseline x' to the gradient of the input x along a straight path. Specifically, the integral gradients of input x and baseline x' along the i-th dimension are defined as follows:

$$IG_i(x) = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha \qquad (2)$$

where $\frac{\partial F(x)}{\partial x_i}$ is the gradient of $F(x)$ along the i-th dimension. Integral gradients have two good properties: sensitivity and implementation invariance. Sensitivity means that if a change in a feature changes the final prediction results, then the feature needs to be assigned an attribution attribute; achieving nondeformation means that if two models have the same function (all the same inputs have the same output) but have different model structures, the interpretable method's interpretation of the two models should be consistent.

In this work, we use integral gradients to explain the prediction results of the MWCNN. Specifically, we first use the integral gradient to calculate the contribution $A \in R^{5\times168}$ of each dimension of the input to the final prediction, and then we average each column of A to obtain the effect of each historical time point of the input on the final prediction. In Experiment F, we explain the prediction of the model on two public datasets.

E. Implementation Details

In all experiments, ReLU [26] is selected as the activation function of the model and the loss function of the model is the mean absolute error. The selection of the optimizer plays an important role in the final convergence performance and convergence time of the model[27], [28]. Since Adam[29] can promote the convergence of deep neural network, we choose Adam as the optimizer. To make the training of the model more stable, we set the learning rate plan for the optimizer. The Adam optimizer's initial learning rate is 0.001, and the learning rate is divided into 10 after every 600 iterations. The total training time of the model is 1200, and the training batch size is 256. He normal initialization [30] is used to initialize the parameters of all the models. To reduce the impact of random initialization, all the random initialization seeds are set to 0. The deep learning framework we use is Keras 2.1.0 with TensorFlow-GPU 1.15.0 as the backend [31], [32]. All the experiments are implemented in the Python 3.6 environment. We use the mean absolute percentage error (MAPE), mean absolute error (MAE) and root mean squared error (RMSE) to evaluate the performance of the model.

$$ReLU(x) = max\{0, x\} \tag{3}$$

$$MAPE = \frac{1}{M}\sum_{i=1}^{M}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{4}$$

$$MAE = \frac{1}{M}\sum_{i=1}^{M}|y_i - \hat{y}_i| \tag{5}$$

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2} \tag{6}$$

where *M* is the number of samples, y_i is the actual load value, and $\hat{y}_i$ is the predicted load value.

IV. EXPERIMENT

A. Datasets

We use two public datasets, the Independent System Operator New England (ISO-NE) dataset and the North American Utility (NAU) dataset, to verify the validity of the model. Both the ISO-NE dataset and the NAU dataset contain one-hour resolution load data. The time range of the ISO-NE dataset is from March 2003 to December 2014, and that of the NAU dataset is from January 1985 to October 1992.

B. Effectiveness of Feature Engineering and the Model Architecture

1) Effectiveness of Wavelet-based Feature Engineering: The quality of feature engineering has a great influence on the model performance[33], [34]. In this case, we mainly verify the validity of the proposed wavelet-based feature engineering and the validity of the MWCNN structure design. We perform experiments on the ISO-NE dataset, where the training set ranges from 2010 to 2011, and the last month of the training set is used as the validation set; the range of the test set is 2012. We compare the proposed feature engineering method based on wavelet reconstruction with the soft-threshold-based wavelet reconstruction method, the hard-threshold-based wavelet reconstruction method, and the unprocessed raw load dataset. The model structure of all the feature engineering methods are shown in Table 2. Since the original load data are one-dimensional, the ANN model structure is adopted. For the sake of fairness, the total number of parameters of the ANN model are as close as possible to those of the MWCNN. Since the level of wavelet decomposition has a great influence on the final prediction results, we decompose all the feature engineering methods from the 1-3 level. The experimental results are shown in Table 2. The best results are obtained by using the wavelet reconstruction method with 2-level high-frequency zeroing. Compared with the ANN, the MAPE is improved by 55.26% and the MAE is improved by 55.21%, which shows that compared with using only the original load data, the potential features in the load sequence can be extracted better by combining a CNN and the high-frequency zeroing wavelet reconstruction method. In the following experiments, we use the wavelet reconstruction method with 2-level high-frequency zeroing. Then, we find that the level of wavelet decomposition has a certain influence on the method of high-frequency zeroing but has a minimal influence on the method of hard threshold. Finally, we notice that the reconstruction method based on soft thresholding will lead to a bad result.

TABLE II
LOAD PREDICTION RESULTS OF THE DIFFERENT DECOMPOSITION LEVELS AND DIFFERENT WAVELET RECONSTRUCTIONS

	MAPE(%)	MAE
ANN	0.76	112.73
MWCNN-1-level	0.39	57.72
MWCNN-2-level	**0.34**	**50.49**
MWCNN-3-level	0.36	52.98
MWCNN-hard-1-level	0.36	53.47
MWCNN-hard-2-level	0.36	52.69
MWCNN-hard-3-level	0.36	52.89
MWCNN-soft-1-level	16.32	2248
MWCNN-soft-2-level	16.29	2246
MWCNN-soft-3-level	16.25	2240

2) Effectiveness of Architecture of the MWCNN: To verify the validity of the MWCNN model structure, in addition to comparing the proposed model with the ANN, we also add one layer or remove one layer to the structure of the MWCNN to show that our proposed model structure is locally optimal. In addition, research shows that adding different attention mechanisms to the channel of the convolution layer can improve the performance of CNN [35], so we also add an SE operation to each convolution layer of the MWCNN. The experimental results are shown in Figure 5. We find that all the CNN-based models are significantly better than the ANN. At the same time, the effect of the MWCNN is slightly better than that of MWCNN1 (one layer is removed on the basis of the MWCNN) and MWCNN2 (one layer is added on the basis of the MWCNN), which indicates that the model result of the MWCNN is locally optimal. Interestingly, increasing the SE does not increase the final prediction performance. We think that this finding is because the model introduces additional training parameters and leads to overfitting.

C. Multiple Wavelet Reconstruction Ensemble Scheme

The number of integrated models has a large impact on the final prediction result, so we first determine the optimal number of integrated models through experiments. Later, we

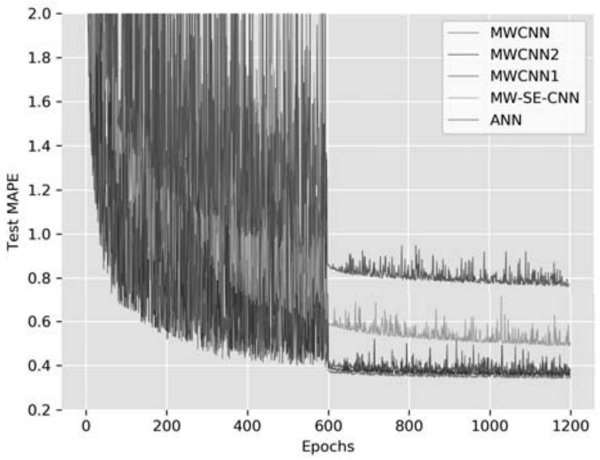

Fig. 5. Test loss values of the MWCNN and other model structures on the ISO-NE dataset. MWCNN1 means that one layer is removed from the model structure of the MWCNN, MWCNN2 means that one layer is added from the model structure of the MWCNN, and MW-SE-CNN is meas that one SE block is added to each convolution kernel in the structure of the MWCNN.

also explore the differences between the integrated model and the single model in predicting the range and standard error of prediction errors. The reason we are concerned about the forecast range and forecast standard deviation is that the energy management efficiency of smart grids will be strongly affected by the forecast range, which is because underestimating the load demand will cause a power shortage, and overestimating the load demand will cause overproduction. In both cases, the larger the forecast range is, the higher the management cost of the smart grid. Therefore, in some cases, managers will choose a predictor with low prediction variance but high average error instead of a predictor with high prediction variance and low average error.

We use the ISO-NE dataset to explore the effectiveness of the multiwavelet-based integration method. The training set ranges from 2007 to June 2008, and the last month is used as the validation set. The test set ranges from July 2008. From 1 to July 31, 2008. We specifically explore the MAPE, the MAE, the maximum deviation of the prediction, and the standard deviation of the prediction deviation from a single model to the integration of 16 models. The experimental results are shown in Table 3. When 6 models are integrated, the prediction performance is the best, and the predicted range and standard deviation of the prediction deviation are much lower than those of the single model. We find that the performance of model prediction does not increase with increasing integration scale, but all the results of the integration model are better than those of the single model, which indicates that the integration method does improve the performance of the model. Although the model predicts the MAX and SD when the integration size is either 10 of 12, they are not too different from the MAX and SD when the integration size is 6. At the same time, from the perspective of actual deployment, it is relatively simple to deploy 6 models, so in the subsequent experiments, the integration size is set to 6.

TABLE III
THE EFFECT OF THE ENSEMBLE SIZE

Ensemble Size	MAPE	MAE	MAX	SD
1	0.3578	61.9005	491.1133	57.3087
2	0.3416	59.0441	461.0059	54.5438
3	0.3400	58.6264	430.6973	52.8673
4	0.3365	58.0773	423.5898	52.5574
5	0.3300	56.9975	421.1367	52.3546
6	**0.3275**	**56.6630**	414.0840	52.0145
7	0.3296	57.0019	419.1855	51.6429
8	0.3300	57.0421	400.1797	51.5771
9	0.3303	57.0216	401.0078	51.5537
10	0.3304	57.0651	**391.5430**	51.5339
11	0.3301	56.9951	393.9629	51.5792
12	0.3313	57.2475	393.4883	**51.4599**
13	0.3336	57.6321	399.2305	51.7851
14	0.3335	57.6187	403.1914	51.6984
15	0.3335	57.6000	403.5293	51.7416
16	0.3335	57.6652	402.4609	51.9615

D. Performance of the Proposed Model on the Public Datasets

To verify the predictive performance of the model, we compared the proposed model with existing methods on the two public datasets. In addition to comparisons with existing methods, we also add an ANN as the baseline. In all the comparisons, the predictive performance of single and integrated models is reported. In the ISO-NE dataset, we performed two performance comparisons due to the different test set ranges selected by the existing methods. We compare the proposed model with the three methods [36], [37], [11], where the training set ranges from January 1, 2007, to June 30, 2008, and the last month of the dataset is used as the validation set. The test set range is from July 1, 2008, to July 31, 2008. The experimental results are shown in Table 4. Our single model and integrated model are better than the other methods. Specifically, compared to WT-ELM-MABC, our single model improves the MAPE by 20% the MAE by 16.81%, and our integrated model improves the MAPE by 26.7% and the MAE by 24.5%. We will also compare the proposed method with the methods of [38], [39], [16], where the training set ranges from January 1, 2004, to December 31, 2005, and the last month of the dataset is used as the validation set and the test set. The range is from May 1, 2006, to May 31, 2006. The experimental results are shown in Table 5. Our single model and integrated model are better than the other methods. Specifically, compared to the method proposed by Pramono et al.[16], our single model improves the MAPE by 21.74%, the MAE by 21.84%, and the RMSE by 24.2%; our ensemble model improves the MAPE by 30.4%, the MAE by 29%, and the RMSE by 29.5%.

In the NAU dataset, we compare the proposed model with five existing methods [40], [41], [10], [7], [11]. The training set range is from January 1, 1988, to October 11, 1990, and the last month of the dataset is used as the validation set. The test set range was from October 12, 1990, to October 12, 1992. To verify that the model has good robustness to temperature noise, Gaussian disturbances are added to the original temperature data, and the model's robustness is verified by calculating the model performance changes before and after the distur-

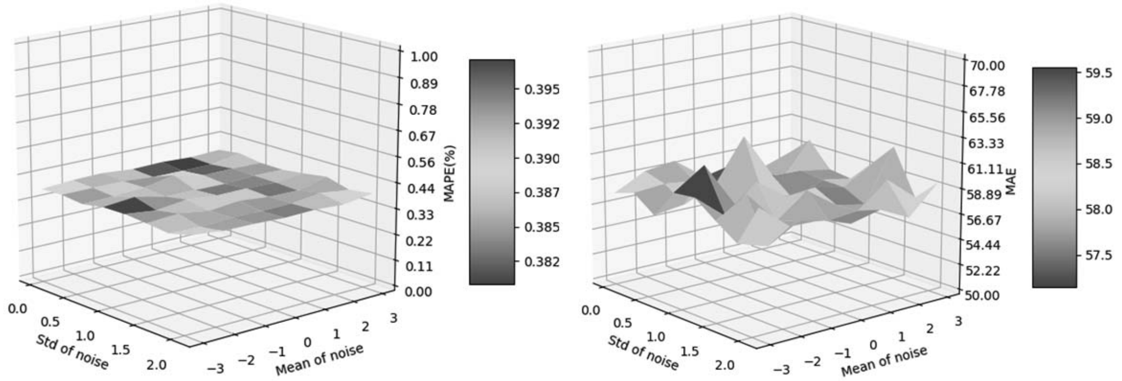

Fig. 6. The MAPE and MAE after adding noise disturbance to the model input. We use Gaussian distributions with a mean in (-3, -2, -1, 0, 1, 2 and 3) and a variance in (0, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8 and 2.1) to generate 56 sets of noise perturbations and add them to the training set. After training, the MAE and MAPE of the undisturbed test set are calculated.

TABLE IV
THE MAPE (%) AND MAE OF ONE-HOUR-AHEAD LOAD FORECASTING ON THE ISO-NE DATASET. + REPRESENTS THE RESULTS OF THE ENSEMBLE METHOD

	MAPE	MAE
ANN	0.79	138.30
ISO-NE[36]	0.81	138
WNN[37]	0.49	84
WT-ELM-MABC[11]	0.45	74.41
MWCNN	0.36	61.90
MWCNN$^+$	**0.33**	**56.19**

TABLE V
THE MAPE (%), MAE AND RMSE OF ONE-HOUR-AHEAD LOAD FORECASTING ON THE ISO-NE DATASET. + REPRESENTS THE RESULTS OF THE ENSEMBLE METHOD

	MAPE	MAE	RMSE
ANN	0.62	86.55	120.34
Tian et al[38]	0.66	88.07	141.97
Kong et al[39]	0.48	65.12	100.50
Wavenet[16]	0.57	78.02	125.11
Pramono et al[16]	0.46	62.23	88.31
MWCNN	0.36	48.64	66.94
MWCNN$^+$	**0.32**	**44.22**	**62.27**

bance addition. Since our proposed method does not require the use of a temperature dataset, increasing the temperature noise perturbation will not cause changes in the prediction results. The experimental results are shown in Table 6. Our single model has the same prediction performance as WT-ELM-MABC. However, because our method does not require temperature data, our model predicts performance when noise is added to the temperature data. Our model is lightly better than WT-ELM-MABC. The prediction performance of our integrated model is better than that of WT-ELM-MABC. At actual temperatures, the MAPE is improved by 4.5%, and at noisy temperatures,the MAPE is improved by 7.2%. The experimental results show that our proposed model has good prediction performance on both public datasets.

TABLE VI
THE MAPE (%) OF ONE-HOUR-AHEAD LOAD FORECASTING ON THE NAU DATASET. + REPRESENTS THE RESULTS OF THE ENSEMBLE METHOD

	Actual temperature	Noisy temperature
ANN	0.85	0.85
ESN[40]	1.14	1.21
M2[41]	1.10	1.11
WT-NN-EA[10]	0.99	-
SSA-SVR[7]	0.72	0.73
WT-ELM-MABC[11]	0.67	0.69
MWCNN	0.67	0.67
MWCNN$^+$	**0.64**	**0.64**

E. Robustness Analysis of the Proposed Model

In the actual deployment environment of load forecasting, due to objective and uncontrollable factors such as data measurement errors or data record deviations, there is usually a certain deviation between the data we finally obtain and the actual data. This phenomenon requires that our proposed model have good robustness; that is, the slight disturbance of the model to the input should not cause excessive prediction differences. Let $f(x)$ be the load prediction model and $\Delta\delta$ be a slight disturbance; then, the robustness of the model can be expressed as $f(x+\Delta\delta) \approx f(x)$. In this experiment, we verify the robustness of the model by adding different degrees of perturbations to the input of the model. Specifically, we use Gaussian distributions with a mean in (-3, -2, -1, 0, 1, 2 and 3) and variance in (0, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8 and 2.1) to generate 56 groups of disturbance noise, and the range of the generated disturbance noise is [-11.45, 11.74]. We only add noise to the training set and evaluate the test set after training the model. We use ISO-NE as the dataset, where the range of the training set is 2006-2007, and the last month is used as the test set. The range of the test set is 2008. The experimental results are shown in Figure 6. We find that adding noise to the data does not have a large impact on the prediction results of the model. In fact, compared to the model without added noise, the maximum increase in the MAE of the model with only noise is only 4.72, and the maximum increase in the

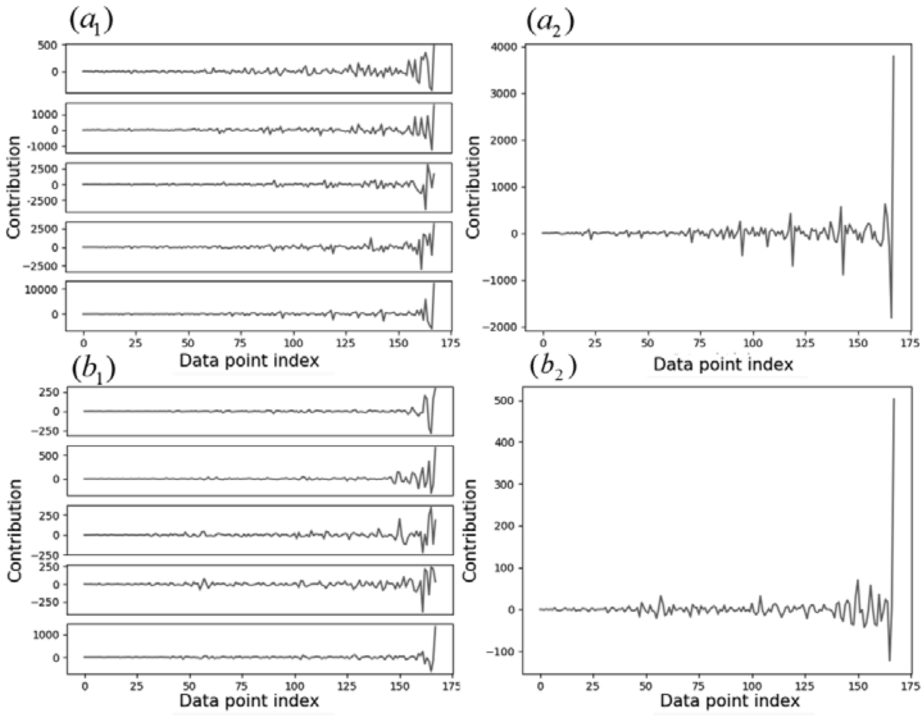

Fig. 7. The figure above uses the integral gradient to calculate the importance of each dimension of certain ISO-NE data, where (a_1) is the contribution value of each dimension to the final prediction and (a_2) is the average of each column of a1 to obtain each historical time contribution of points to the final prediction. The following figure shows the calculation results for the NAU data.

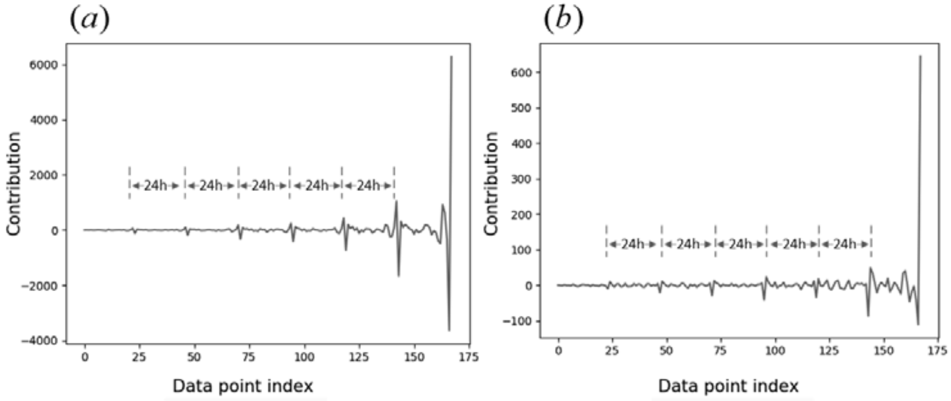

Fig. 8. (a) is the contribution value of historical data points to the final prediction on the ISO-NE dataset, and (b) is the contribution value of historical data points to the final prediction on the NAU dataset. We select 100 data points from the two datasets, calculate the contribution value of the historical data points of a single data point to the final prediction, and average the contribution values at the same time point to obtain the final result.

MAPE is only 0.03%. When a Gaussian disturbance with a mean of -2 and a variance of 1.8 is added to the model, the model's MAE is reduced by 1.24. This finding shows that our proposed model has very good robustness.

F. Interpretability of Load Forecasting

Although many neural network-based methods are currently used for load prediction [11], [15], [16], these methods do not provide any explanation for the results of load prediction. In this use case, we use the integral gradient [25] to explain that our model relies on that input to make the final predictions. We provide attributional explanations for the prediction results of the MWCNN model in Experiment D on the two public datasets. Specifically, we randomly select one dataset from ISO-NE and NAU and provide the prediction basis of the model. The experimental results are shown in Figure 7. The figure above uses the integral gradient to calculate the importance of each dimension of certain ISO-NE data, where (a_1) is the contribution value of each specific dimension to the final prediction and (a_2) is the (a_1). Each column is averaged to obtain the contribution value of each historical time point to the final prediction. The following figure shows the calculation results of the NAU data. From (a_2) and (b_2), we can clearly find that the data closer to the prediction time point have a greater impact on the final prediction, but this impact has both positive and negative effects. At the same time, both the ISO-

NE and NAU datasets show that the first data point near the prediction time has the largest impact on the final prediction and is positive.

Considering that the interpretation given for the prediction of singleton data may have randomness, we make predictions for more data to obtain more general interpretation conclusions. We first select 100 data points on the ISO-NE and NAU datasets and calculate the historical time point of each data point to the final prediction value. The calculation results are shown in Figure 7 (a_2). Then, we averaged the contribution values of the final predictions at the 100 same historical time points to obtain the contribution value of the final predictions at each historical time point of the many datasets. The experimental results are shown in Figure 8, where Figure 8 (a) is the result calculated on the ISO-NE dataset, and Figure 8 (b) is the result calculated on the NAU dataset. From the figure, we can obtain the same predictions as the singleton prediction interpretation. The data closer to the prediction time point have a greater impact on the final result, and the first data point near the prediction time has the largest impact on the final prediction and is positive. In addition, we also find that although the data points that are farther away from the prediction time point have a weaker contribution to the prediction, some local areas have some data points that contribute far more than other points in the area, as shown in Figure 8. The right-most value in the red interval region, and the maximum value in the interval decreases with time, which also indicates that the data with a longer isolation prediction time have less contribution to the prediction. Interestingly, the interval from the maximum value of one region to the maximum value of another region is exactly 24 h, which shows that the MWCNN model can well capture the local periodicity in historical prediction data. We hypothesize that the reason why the MWCNN model has better prediction performance than the other models is because the MWCNN model can better capture the potential periodic features in the load data.

V. Conclusion and Discussion

In this work, we propose a feature engineering method based on wavelet reconstruction, and based on this, we propose an MWCNN for short-term load prediction. To further improve the prediction performance, we propose an integrated scheme based on multiple wavelets. The MWCNN does not use external datasets, and the storage size of the model is only 497 KB. The MWCNN has superior prediction performance and good robustness, which we have verified on two public datasets. In terms of the predictive performance of the model, the robustness of the model, the dependence of the model on external data, and the size of the model storage, the MWCNN has good practical deployability.

We believe that the effectiveness of MWCNN mainly comes from the following three aspects. Firstly, we propose a feature engineering method based on wavelet reconstruction. In this method, different types of wavelet are used to decompose the original load series, and then the high frequency component is set to 0 for reconstruction. The results of *Experiment B* show the effectiveness of our feature engineering. Secondly, we use CNN to extract the potential features of the reconstructed load data. The results of experiment F show that MWCNN can well extract the potential periodic characteristics of load data. We think this is the main reason why WMCNN can achieve superior performance. Finally, we adopt the wavelet-based ensemble scheme to further improve the prediction performance. There are two obvious benefits to using this method: first, the dataset does not need to be randomly divided, and all the datasets can be used to train the model, which can ensure that the prediction accuracy of a single model is as high as possible. Second, different wavelet clusters are used to reconstruct the original data to obtain different inputs,which can ensure the diversity of the model. The results of the *Experiment D* show the effectiveness of our ensemble method.

References

[1] N. Tang, S. Mao, Y. Wang, and R. Nelms, "Solar power generation forecasting with a lasso-based approach," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1090–1099, 2018.

[2] H. Hahn, S. Meyer-Nieberg, and S. Pickl, "Electric load forecasting methods: Tools for decision making," *European journal of operational research*, vol. 199, no. 3, pp. 902–907, 2009.

[3] K.-B. Song, Y.-S. Baek, D. H. Hong, and G. Jang, "Short-term load forecasting for the holidays using fuzzy linear regression method," *IEEE transactions on power systems*, vol. 20, no. 1, pp. 96–101, 2005.

[4] W. Charytoniuk, M. Chen, and P. Van Olinda, "Nonparametric regression based short-term load forecasting," *IEEE transactions on Power Systems*, vol. 13, no. 3, pp. 725–730, 1998.

[5] J. W. Taylor, "Short-term electricity demand forecasting using double seasonal exponential smoothing," *Journal of the Operational Research Society*, vol. 54, no. 8, pp. 799–805, 2003.

[6] A. Ghanbari, S. Abbasian-Naghneh, and E. Hadavandi, "An intelligent load forecasting expert system by integration of ant colony optimization, genetic algorithms and fuzzy logic," in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE, 2011, pp. 246–251.

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

[8] E. E. Elattar, J. Goulermas, and Q. H. Wu, "Electric load forecasting based on locally weighted support vector regression," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 4, pp. 438–447, 2010.

[9] B.-L. Zhang and Z.-Y. Dong, "An adaptive neural-wavelet model for short term load forecasting," *Electric power systems research*, vol. 59, no. 2, pp. 121–129, 2001.

[10] 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, 2009.

[11] S. Li, P. Wang, and L. Goel, "Short-term load forecasting by wavelet transform and evolutionary extreme learning machine," *Electric Power Systems Research*, vol. 122, pp. 96–103, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] S. Ryu, J. Noh, and H. Kim, "Deep neural network based demand side short term load forecasting," *Energies*, vol. 10, no. 1, p. 3, 2017.

[15] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3943–3952, 2018.

[16] S. H. Pramono, M. Rohmatillah, E. Maulana, R. N. Hasanah, and F. Hario, "Deep learning-based short-term load forecasting for supporting demand response program in hybrid energy system," *Energies*, vol. 12, no. 17, p. 3359, 2019.

[17] 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 Transactions on Power Systems*, vol. 25, no. 1, pp. 322–330, 2009.

[18] S. Muzaffar and A. Afshari, “Short-term load forecasts using lstm networks,” *Energy Procedia*, vol. 158, pp. 2922–2927, 2019.

[19] D. T. Lee and A. Yamamoto, “Wavelet analysis: theory and applications,” *Hewlett Packard Journal*, vol. 45, pp. 44–44, 1994.

[20] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[21] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.

[22] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*, 2013.

[23] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, “Not just a black box: Learning important features through propagating activation differences,” *arXiv preprint arXiv:1605.01713*, 2016.

[24] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, “Evaluating the visualization of what a deep neural network has learned,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2660–2673, 2016.

[25] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic attribution for deep networks,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3319–3328.

[26] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for lvcsr using rectified linear units and dropout,” in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 8609–8613.

[27] L. Abualigah, “Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications,” *Neural Computing and Applications*, pp. 1–21, 2020.

[28] L. M. Abualigah and A. T. Khader, “Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering,” *The Journal of Supercomputing*, vol. 73, no. 11, pp. 4773–4795, 2017.

[29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[30] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[31] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[32] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[33] L. M. Q. Abualigah, *Feature selection and enhanced krill herd algorithm for text document clustering*. Springer, 2019.

[34] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, “A new feature selection method to improve the document clustering using particle swarm optimization algorithm,” *Journal of Computational Science*, vol. 25, pp. 456–466, 2018.

[35] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[36] P. Shamsollahi, K. Cheung, Q. Chen, and E. H. Germain, “A neural network based very short term load forecaster for the interim iso new england electricity market system,” in *PICA 2001. Innovative Computing for Power-Electric Energy Meets the Market. 22nd IEEE Power Engineering Society. International Conference on Power Industry Computer Applications (Cat. No. 01CH37195)*. IEEE, 2001, pp. 217–222.

[37] C. Guan, P. B. Luh, L. D. Michel, Y. Wang, and P. B. Friedland, “Very short-term load forecasting: wavelet neural networks with data pre-filtering,” *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 30–41, 2012.

[38] C. Tian, J. Ma, C. Zhang, and P. Zhan, “A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network,” *Energies*, vol. 11, no. 12, p. 3493, 2018.

[39] 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 Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2017.

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

[41] A. R. Reis and A. A. Da Silva, “Feature extraction via multiresolution analysis for short-term load forecasting,” *IEEE Transactions on power systems*, vol. 20, no. 1, pp. 189–198, 2005.