Non-intrusive load decomposition model based on Group Bayesian optimization and post-processing

Tan Zhukui[1], Liu Bin[1], Zhang Qiuyan[1], Ding Chao[1], and Hu Houpeng [1]

[1]Electric Power Research Institute of Guizhou Power Grid Co., Ltd., 550002 Guiyang, Guizhou, China

Abstract: Non-intrusive load decomposition can decompose the power consumption of a single appliance from the household bus data, which is of great significance for users to adjust their own power consumption strategy. In order to solve the problem of large amount of computation in hyperparameter optimization of load decomposition model based on deep residual network, a Group Bayesian optimization method is proposed. This method can obtain better hyperparameter combination with less computational cost. In addition, in order to solve the problem of irrelevant activation of the model decomposition results, an improved post-processing method is proposed to improve the comprehensive performance of the model. Finally, the public data set REFIT is used to verify the proposed method, and the results show that the proposed method has a low decomposition error.

1 Introduction

One of the urgent needs of smart grid and energy Internet is to obtain the electricity consumption data of individual electrical appliance, based on which users can understand the energy consumption law of each electrical appliance and reduce energy consumption pertinently. This is an important step towards transparency and intelligence of the power grid. The current measurement technology can only automatically read the total power consumption data, and it is difficult to further obtain a user's internal load information. The load decomposition technology has become a major bottleneck in the development of smart grid.

Load decomposition techniques can be divided into Intrusive Load Decomposition (ILD) and Non-Intrusive Load Decomposition (NILD) method. NILD is a technology for estimating the power consumption of each electrical appliance of the user when the total power demand of the user bus is known. ILD technology needs to install measuring instruments on all the electrical appliances of the user, which leads to high investment cost and difficult maintenance. In contrast, NILD has the advantages of easy installation, convenient maintenance, simple hardware and so on. Correspondingly, NILD puts forward higher requirements for software algorithms, which has become the biggest obstacle for NILD to put into practical application.

Hart [1] put forward the concept of NILD for the first time and adopted the method of Combinatorial Optimization (CO) to solve this problem, creating a precedent in the field of NILD. Subsequently, a variety of machine learning methods have played an important role in the research of NILD, including k-Nearest Neighbor (KNN) algorithm, Support Vector Machine (SVM), Matrix Factorization and so on [2,3]. Among them, Hidden Markov Model (HMM) has become one of the more popular methods, because it can independently model the running states of a single electrical appliance, and achieve better decomposition results. In reference [4], a single electrical appliance is described as a HMM model, and NILD problem is described as a Factorial HMM (FHMM) problem. The model is solved by maximizing the logarithmic likelihood probability of the total power sequence. However, the traditional machine learning methods also have some limitations, and there is room for improvement in decomposition accuracy and generalization performance.

In recent years, as deep learning and artificial intelligence have become the focus of research, the research methods of NILD have gradually transferred from traditional machine learning to deep neural networks. The advantage of deep neural network is that a more complex and accurate model can be established through the superposition of neural network layers. In reference [5], a sequence-to-sequence (seq2seq) structure based on neural network is proposed and solved by RNN or CNN model, while in reference [6], the seq2seq method is improved and a sequence-to-point (seq2point) structure is proposed, which solves the problems such as disappearance of model gradient and inaccurate prediction of edge points in seq2seq structure. Reference [7] uses the same seq2point model as [6], which proves that this method has strong learning ability of cross-appliance and cross-domain transfer.

Although the above methods have achieved certain results, there are still two common problems to be solved in the current NILD algorithms:

1) The structure of the deep neural network is quite complex, so the hyperparameters in the network can easily reach millions. An efficient method is needed to determine a set of superior hyperparameters in numerous combinations with low time and hardware cost to improve the accuracy of the model.

2) One problem with NILD is that there may be many unreasonable prediction results in the decomposition results. These irrelevant activations often damage the performance of the model, but there is no simple and efficient method to deal with this phenomenon.

In view of this, we introduce the concept of group optimization on the basis of the traditional Bayesian optimization method, and proposes a Group Bayesian optimization method to realize hyperparameter optimization of deep residual network (ResNet), obtaining a better load decomposition model with a small computational cost; at the same time, a post-processing method for the decomposition results of the model is creatively proposed to eliminate unreasonable activation and further improve the comprehensive performance of the model. Finally, the public dataset is used to verify the method proposed in this paper, which proves the effectiveness of the method we propose.

2 Methods and Materials

2.1 NILD problem

NILD aims to predict the electricity consumption of a single appliance from the total electricity consumption sequence. Assuming that it is known that the bus power consumption of a power user in a certain period of time is $P_{sum} = [P_1, P_2, ..., P_T]$, where T is the length of time, the total power consumed by the user at a certain time can be expressed as:

$$P_t = \sum_{i=1}^{N} P_t^i + e_t \tag{1}$$

where $t = \{1, 2, \cdots, T\}$; $i = \{1, 2, \cdots, N\}$; N is the number of electrical appliances; P_t represents the total power at time t ; P_t^i represents the power of the appliance i at time t and e_t represents the noise of the model.

2.2 Deep ResNet

In NILD task, extracting the operating characteristics of the target electrical appliance from the bus power is the key to achieving load decomposition. Convolutional Neural Network (CNN) can realize feature extraction by constructing different convolution kernels [6,8]. We can understand the power sequence extracted from the user bus as a two-dimensional image that integrates the electrical behavior characteristics of multiple electrical appliances, and introduce the method of CNN to extract the unique operating characteristics of the target appliance to achieve load decomposition.

The performance of CNN is closely related to the depth of the network. It is found that with the increase of the number of network layers, the phenomenon of gradient disappearance and gradient explosion will hinder the convergence of the model. In order to solve this problem, we introduce the deep ResNet model proposed by He et al. [9].

The difference between the deep ResNet and the general CNN is that the former introduces the idea of fast connection. The original input skips multiple convolution layers through the fast connection and is directly superimposed with the residual into the output. This operation enables the deep ResNet to learn the original features directly from the input, so as to avoid the disappearance or explosion of gradients.

The schematic diagram of the residual block, the basic unit of the deep ResNet, is shown in Figure 1. It can be seen that the residual block is divided into two parts: the connection on the left is identical mapping, and the connection on the right is the residual part, which is composed of two convolution operations and their corresponding Batch Normalization and Relu activation process. The final output of the residual block can be obtained by superimposing the left and right parts and activating the Relu function. When the number of channels of the input and output is not the same, the identity mapping can not be used, and the 1×1 convolution operation is needed to increase or reduce the dimension. The mathematical expression of the residual block is:

$$y_l = F(x_l, \Theta_l) + h(x_l) \tag{2}$$

$$x_{l+1} = f(y_l) \tag{3}$$

where x_l represents the input of the residual block; x_{l+1} represents the output of the residual block and Θ_l is the weight matrix which represents the convolution operation while in CNN. When the number of input and output channels are the same, $h(\bullet)$ is an identity mapping, and is 1×1 convolution operation while they are inconsistent. $f(\bullet)$ is the activation function for which Relu function is generally used.

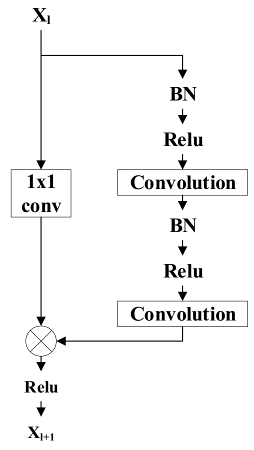


Fig. 1. Schematic diagram of the structure of residual block.

The application of residual block makes it possible to further deepen CNN, which is helpful to extract higher-level features from power sequence and improve the accuracy of load decomposition model.

2.3 Load decomposition model based on the deep ResNet

The structure of the deep ResNet finally built in this paper is shown in Figure 2. It uses the total power sequence intercepted by the sliding window as the input. The first convolution layer extracts the preliminary characteristics of the load, and then the output of the convolution layer is sequentially fed into the residual block to extract higher-level and more abstract power consumption behavior characteristics, and the final fully connected layer realizes the nonlinear mapping of the feature vector to the target appliance power value in the midpoint of the power sequence.

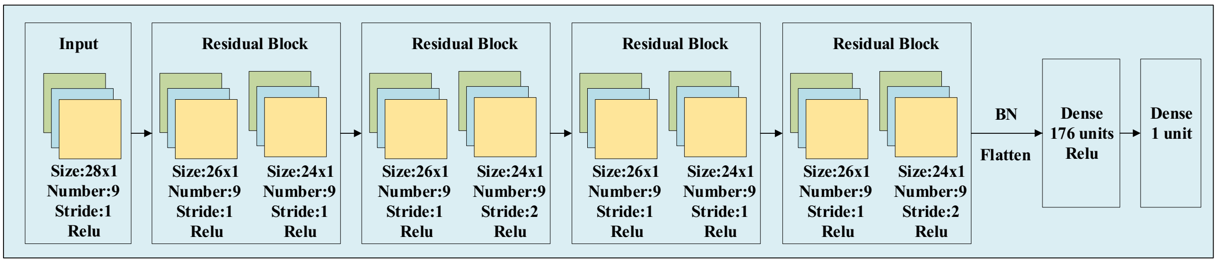


Fig. 2. Schematic diagram of the structure of the deep ResNet.

2.4 Bayesian optimization

The NILD model based on the deep ResNet introduced above has many adjustable hyperparameters, such as the size of the convolution kernel, the sliding step size, the number of neurons in the fully connected layer, and the size of the sliding window, etc. The selection of hyperparameters will greatly affect the performance of the model. In order to ensure a better model decomposition effect, we use Bayesian optimization to optimize the hyperparameters instead of manually setting them.

Bayesian optimization utilizes the Gaussian process. For load decomposition model $Y = f(X)$, we hope to determine the next search point through a combination of known hyperparameters. If t sets of hyperparameters $x_1, x_2, \cdots x_t$ and the values of the model corresponding to these sets of hyperparameters $f(x_{1:t}) = [f(x_1), f(x_2), \cdots f(x_t)]$ has been obtained. Then the Gaussian process assumes that this vector obeys the t-dimensional Gaussian distribution $f(x_{1:t}) \sim N(\mu(x_{1:t}), \Sigma(x_{1:t}, x_{1:t}))$, where $\mu(x_{1:t}) = [\mu(x_1), \mu(x_2), \cdots \mu(x_t)]$. The covariance matrix is expressed as:

$$\Sigma(x_{1:t}, x_{1:t}) = \begin{bmatrix} \mathrm{cov}(x_1, x_1) & \cdots & \mathrm{cov}(x_1, x_t) \\ \cdots & \cdots & \cdots \\ \mathrm{cov}(x_t, x_1) & \cdots & \mathrm{cov}(x_t, x_t) \end{bmatrix} = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_t) \\ \cdots & \cdots & \cdots \\ k(x_t, x_1) & \cdots & k(x_t, x_t) \end{bmatrix} \tag{4}$$

Covariance is calculated by Gaussian kernel function:

$$k(x_1, x_2) = \alpha_0 \exp(-\frac{1}{2\xi^2}\|x_1 - x_2\|^2) \tag{5}$$

where α_0 and ξ are parameters of the kernel function.

After obtaining t sets of candidate solutions, the corresponding Gaussian regression model is established to obtain the posterior probability of the model index value at any point, and the posterior probability is used to construct a collection function to determine the next set of hyperparameter combinations that need to be searched. The collection function expression is:

$$EI_{\mathrm{t}}(x) = E[\max(f(x) - \max(f(x_1), f(x_2), \cdots f(x_t))] \tag{6}$$

$$x_{t+1} = \arg\max EI_t(x) \tag{7}$$

where $E[\bullet]$ is expected value.

For the points x_{t+1} and $f(x_{t+1})$ that need to be searched next, assuming that after adding a new sampling point, $f(x_{1:t+1})$ obey the t-dimensional normal distribution. The mean vector and covariance matrix are divided into blocks, which can be written as:

$$\begin{bmatrix} f(x_{1:t}) \\ f(x_{t+1}) \end{bmatrix} \sim \left(\begin{bmatrix} \Sigma(x_{1:t}, x_{1:t}) & \kappa \\ \kappa^T & k(x_{t+1}, x_{t+1}) \end{bmatrix} \right) \tag{8}$$

where $\kappa = [k(x_{t+1}, x_1), k(x_{t+1}, x_2), \cdots k(x_{t+1}, x_t)]^T$.

Therefore, the search process of model hyperparameters can be described as finding the conditional distribution of $f(x_{t+1})$ when $f(x_{1:t})$ is known. According to the nature of the multi-dimensional normal distribution, the conditional distribution obeys one-dimensional normal distribution. The calculation formula is:

$$\mu = \kappa(\Sigma(x_{1:t}, x_{1:t}))^{-1}(f(x_{1:t}) - \mu(x_{1:t})) + \mu(x_{t+1}) \tag{9}$$

$$\sigma^2 = k(x_{t+1}, x_{t+1}) - \kappa(\Sigma(x_{1:t}, x_{1:t}))^{-1}\kappa^T \tag{10}$$

where μ is related to $f(x_{1:t})$ but σ^2 only related to the covariance value calculated by the Gaussian kernel function, and has nothing to do with $f(x_{1:t})$.

2.5 Group Bayesian optimization

On the basis of Bayesian optimization, this model introduces the idea of group optimization, and realizes the intelligence of problem solving through the search behavior of the body and the information interaction within the group. The specific steps of the Group Bayesian optimization method proposed in this paper are as follows:

Step1: Initialize the maximum number of searches and the total number of searched individuals, and give each individual an initial hyperparameter combination within a reasonable range.

Step2: Construct models separately according to the hyperparameter combination of each individual, use the same data set for training, and obtain the index value of the model under the hyperparameter combination corresponding to each individual.

Step3: For each individual, compare its hyperparameter combination with the hyperparameter combination corresponding to the individual with the best model index value in the group, and update it according to the following formula:

$$x_{id} = x_{id} + rand(0,1)(p_{id} - x_{id}) + rand(0,1)(p_{gd} - x_{id}) \tag{11}$$

where x_{id} is the value of the d dimension of the i-th individual; p_{id} is the optimal value of the d dimension of the i-th individual; p_{gd} is the optimal value of the d dimension in the group; $rand(0,1)$ is a random number from 0 to 1.

Step 4: If the end condition is not met, return to step 2, otherwise the algorithm ends, and the hyperparameter combination corresponding to the model with the best model index value in the group is the optimal solution searched out.

2.6 Training method

The NILD problem to be solved in this paper is a typical regression problem. Using the mean square error (MSE) as the loss function used in network training, that is, using the square of the difference between the real value and the predicted value and then summing the average, MSE can describe the total gap between the real value and the predicted value. The smaller MSE is, the higher the prediction accuracy of the model is, and the mathematical expression is as follows:

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2 \tag{12}$$

where y_n represents the real value of the model and $\hat{y}_n$ represents the predicted value of the model.

3 Results & Discussion

In this paper, python programming environment is used to develop the algorithm, and the deep ResNet model is built on the framework of tensorflow + keras. Model training and testing is completed on a server with 4 GeForce RTX 2080 Ti graphics cards, and we use GPU for hardware acceleration.

3.1 Dataset

We select the public data set REFIT to simulate the built model. The REFIT dataset is a public dataset from three universities Loughborough, Strathclyde and East Anglia which provides electricity use data of 20 households in England, covering the period from 2013 to 2015 [10]. These data include the total household electricity consumption data and the electricity consumption data of a single appliance with a frequency of 8s. We selected fridge, kettle, microwave, washing machine and dishwasher as the target research appliances which are the most used and accounted for the largest proportion of energy consumption in the households.

Table 1 shows the splits for the REFIT data in the simulation:

Table 1. Distribution of the REFIT dataset

Appliance	Training (House)	Validation (House)	Test (House)	Amount of data
Kettle	17,5,6,20,7,13,19	5	17	4000000
Washing machine	10,7,13,18,15,16,17	15	16	4000000
Microwave	4,10,12,17,19	17	4	2400000
Dishwasher	5,7,9,13,16,18,20	18	20	4000000
Fridge	2,5,9,12,15,3,10	12	15	4000000

3.2 Pre-processing

In order to avoid deviations in the results of load decomposition caused by different total power levels, we use the minimum-maximization normalization method to pre-process the original data and map the data to the [0,1] interval. The normalization function is as follows:

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{13}$$

where x^* represents the normalized data; $x_{\max}$ represents the maximum value of the power sequence data and $x_{\min}$ represents the minimum value of the power sequence data.

3.3 Post-processing

Post-processing refers to revising the decomposition results of the model based on prior knowledge. Literature [11] found through research that the decomposition results of the load decomposition model not only include the true on state of the electrical appliance, but also often include some sporadic “false” activations, that is, the model mistakes the electrical appliance in the on state, which leads to the final decomposition value bias high, which reduces the decomposition performance of the model.

For these "false" activations, simple logical judgments can often be used to eliminate them, thereby further improving the performance of the model. For this reason, we propose a post-processing method for the model decomposition results, including five steps:

Step1: Record the shortest activation time of the target appliance in the training data through the threshold method.

Step2: Use the threshold method to record the duration of each activation of the target electrical power decomposition value.

Step3: Eliminate the activations whose activation duration is less than the shortest activation time in the power decomposition value of the target appliance.

Step4: Judge the total load power section corresponding to the remaining activation. If the total load power in this section has corresponding power rises and drops, then the activation is considered reasonable; otherwise, the activation is considered unreasonable, and it is removed.

3.4 Metrics

We use three typical metrics to evaluate the performance of the model. The first metric is the Mean Absolute Error (MAE), which is defined as the average value of the absolute error between the predicted value of the target electrical appliance and the true value at each moment. The mathematical expression is as follows:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}\left|\hat{y}_t - y_t\right| \tag{14}$$

where y_t represents the real power of the target appliance at time t and $\hat{y}_t$ represents the predicted power of the target appliance at time t.

The second metric is the normalized signal aggregation error (SAE), which is defined as the relative error of the total energy between the predicted value of the target appliance and the true value. The mathematical expression is as follows:

$$SAE = \frac{\left|\hat{E} - E\right|}{E} \tag{15}$$

where E represents the total energy consumption of the appliance, that is $E = \sum_{t=1}^{T} y_t$, and $\hat{E}$ represents the predicted appliance energy consumption, that is $\hat{E} = \sum_{t=1}^{T} \hat{y}_t$.

The third metric is the standard decomposition error (NDE), which is defined as the normalized error of the squared difference between the predicted value of the target electrical appliance power and the true value. The mathematical expression is as follows:

$$NDE = \frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{\sum_{t=1}^{T} {y_t}^2} \tag{16}$$

3.5 Experimental results

In order to build a better load decomposition model, we first use Group Bayesian optimization to find a set of suitable hyperparameters for the load decomposition models of kettle, microwave, fridge, washing machine and dishwasher. The hyperparameters that need to be optimized here include the size of the convolution kernel of the convolution layer, the sliding step size, the number of neurons in the fully connected layer, and the size of the sliding window. Figure 3 shows the results of Group Bayesian optimization.

After constructing the optimal load decomposition model for each electrical appliance, the test set is used to test the decomposition performance of the model, and the data of the test set is not involved in model training. Figure 4 shows the true and predicted values of the above electrical power. The results show that the deep ResNet we proposed can extract the operating characteristics of electrical appliances and almost accurately capture every activation. However, it can be found that the model also predicts some irrelevant activations, which will damage the decomposition accuracy of the model. After using the post-processing algorithm we proposed to deal with this problem, it can be found from Figure 4 that these irrelevant activations have been eliminated, which can prove the effectiveness of our post-processing method. Table 2 shows the comparison results of various metrics between the method in this paper and the method in [7]. It can be found that the deep ResNet we proposed has achieved optimal values in MAE, SAE and NDE metrics compared with ordinary CNN. This is due to the fact that the deep ResNet can be stacked very deep and it can extract deeper features to achieve load decomposition. Especially after post-processing, the performance of the model can be further improved on the original basis.

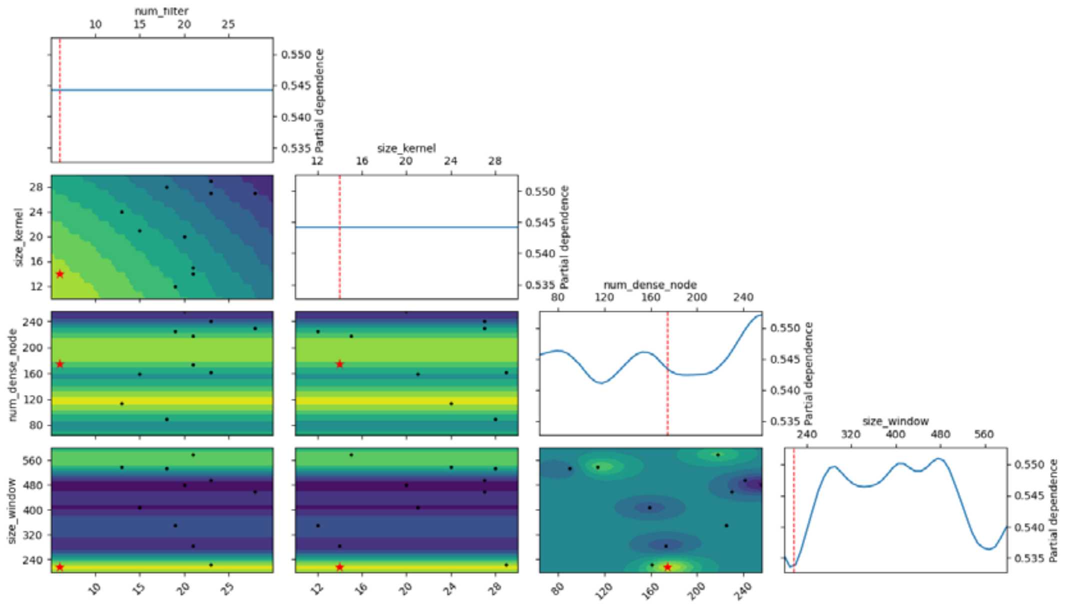
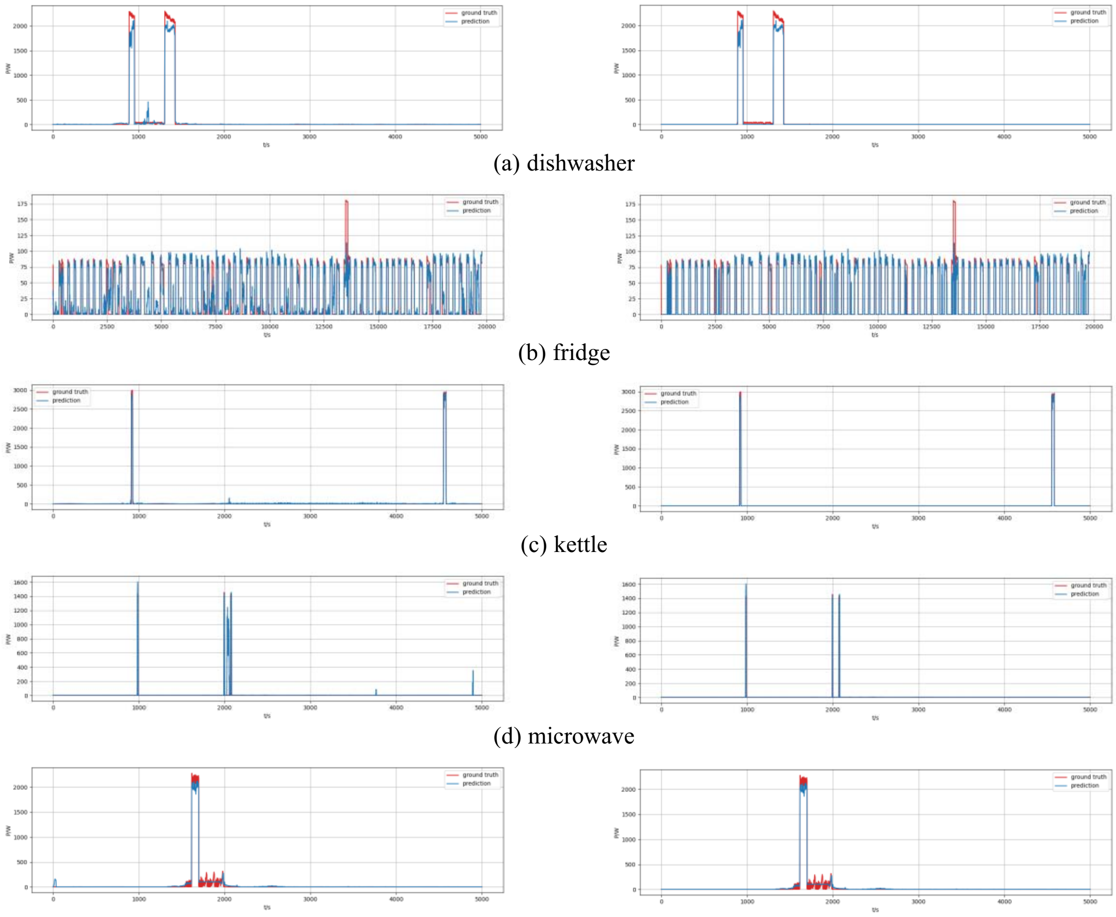


Fig. 3. the results of Group Bayesian optimization.



(a) dishwasher

(b) fridge

(c) kettle

(d) microwave

(e) washing machine

Fig. 4. Comparison of predicted and true power values of different electrical appliances.
(left: without post-processing, right: with post-processing)

Table 2. Comparison of decomposition errors of different methods

Appliance	CNN			without post-processing			with post-processing		
	NDE	SAE	MAE	NDE	SAE	MAE	NDE	SAE	MAE
washing machine	0.54	2.61	16.85	0.26	0.28	3.73	0.19	0.03	2.04

kettle	0.52	0.13	6.83	0.31	0.1	11.22	0.19	0.07	9.79
fridge	0.43	0.33	20.02	0.37	0.25	15.12	0.22	0.23	10.96
microwave	0.71	0.17	12.66	0.57	0.15	4.67	0.51	0.13	4.38
dishwasher	0.44	0.26	12.26	0.31	0.08	9.3	0.3	0.04	8.36

4 Conclusions

In this paper, based on the theory of CNN and residual mechanism, a NILD method based on deep ResNet and improved post-processing mechanism is proposed. In this method, the power sequence is converted into feature images, and the behavior characteristics of many kinds of electrical appliances are expressed in the way of images, so as to realize load decomposition. The results of numerical examples show that the proposed method achieves lower error level and higher decomposition accuracy. The contributions of this paper are as follows:

1) Based on the CNN theory, the power sequence to be decomposed is transformed into a two-dimensional feature image, and the load decomposition model based on deep ResNet can fully extract the characteristics of multi-appliance behavior contained in the power sequence, so as to obtain the decomposition power of the target appliance.

2) Based on the theory of group optimization and Bayesian optimization, a Group Bayesian optimization method is proposed to realize hyperparameter optimization more efficiently.

3) In view of the unreasonable activation in the results of load decomposition, an improved post-processing method is proposed in this paper to improve the comprehensive performance of the model in an all-round way.

The transfer and learning ability of deep ResNet model can be further studied in the future.

Acknowledgments

Funding: The National Natural Science Foundation of China (No. 51777078), Guizhou Power Grid Co., Ltd. Technology Project (No. 066600KK52180051)

References

1. Hart, G.W. (1992) Nonintrusive appliance load monitoring. Proceedings of the IEEE, 80(12): 1870–1891.
2. Faustine, A., Mvungi, N.H., Kaijage, S., Michael, K. (2017) A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem. arXiv 2017, arXiv: 1703.00785.
3. Rahimpour, A., Qi, H., Fugate, D., Kuruganti, T.

   Non-Intrusive Energy Disaggregation Using Non-Negative Matrix Factorization With Sum-to-k Constraint. (2017) IEEE T rans. Power Syst, 32: 4430–4441.
4. Kong, W.C., Dong, Z.Y., Ma, J., Hill, D.J., Zhao, J.H., Luo, F.J. (2018) An Extensible Approach for Non-Intrusive Load Disaggregation With Smart Meter Data. IEEE TRANSACTIONS ON SMART GRID, 9(4): 3362-3372.
5. Kelly, J., Knottenbelt, W. (2015) Neural NILM: Deep neural networks applied to energy disaggregation. In: Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments. Seoul. pp. 55–64.
6. Zhang, C., Zhong, M., Wang, Z., Goddard, N., Sutton, C. (2018) Sequence-to-point learning with neural networks for nonintrusive load monitoring. In: The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18). New Orleans. pp. 2604-2611.
7. D'Incecco, M., Squartini, S., Zhong, M. J. (2020) Transfer Learning for Non-Intrusive Load Monitoring. IEEE TRANSACTIONS ON SMART GRID, 11(2): 1419-1429.
8. Shin, C., Joo, S., Yim, J., Lee, H., Moon, T., Rhee, W. (2019) Subtask gated networks for non-intrusive load monitoring. In: The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19). Honolulu. pp. 1150-1157
9. He, K.M., Zhang, X.Y., Ren, S.Q., Sun, J. (2016) Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. Seattle. pp: 770-778
10. Murray, D., Stankovic, L., Stankovic, V. (2017) An electrical load measurements dataset of united kingdom households from a two-year longitudinal study. Scientific data, vol. 4, p. 160122.
11. Rafiq, H., Shi, X., Zhang, H., et al. (2020) A Deep Recurrent Neural Network for Non-Intrusive Load Monitoring Based on Multi-Feature Input Space and Post-Processing. Energies, 13(9): 2195.