Utilizing Block Characters for Real-time Load Forecasting in Energy Internet

Yangyang Ming
Department of Automation
Tsinghua University
Beijing，CHINA
yitianxk@126.com

Junwei Cao
Beijing National Research Center for Informance Science and Technology
Tsinghua University
Beijing, CHINA
jcao@tsinghua.edu.cn

***Abstract*—With the developing of Energy Internet (EI), its load forecast becomes more and more important and necessary. Although the load of EI has some temporal periodic features, its inherent uncertainties in different types of loads make accurate prediction more difficult. Sometimes even the typical neural network may produce apparent load forecast error and take non-negligible computing time. In EI load forecast with one step further real-time prediction, we apply the threshold factor of block characters into dimensions determination of ELM's (Extreme Learning Machine) training model size. Through corresponding simulations, the proposed algorithm shows more accurate results than ELM used alone, and also than ESN (Echo State Network), which exhibits large application potentials.**

***Keywords*—load forecast, Energy Internet, threshold factor, block characters, ELM**

I. INTRODUCTION

Through the smart using of related data in EI (Energy Internet) operation, its energy utilizing efficiencies can be largely promoted by adopting cyber-energy infrastructure integration in our previous studies [1-3], where load forecast becomes its basic and fundament function for the high effective operations of infrastructure integration. By using AI and machine learning related techniques [4-6], this function will ease the management and operation of EI and can be extended to other industries [7,8].

As in EI operation, due to its operation environment, the sampled data usually have complex operation characters, such as quasi-temporal period arises with random disturbs. Its complex is mainly due to multi-source energy generation such as wind, PV etc. renewable energy source [9,10] and multi-type consuming customers controlled by micro grid operation [11] and demand side management [12]. In this situation, even some high complex neural networks may not afford the load forecast task timely. To improve its performance, new neural network model maybe needed.

Under this working condition, many load forecast algorithms are proposed [13], which often utilize the potential temporal cycle and working day type characters, and typically adopt neural network like MLP (Multi-Layer Perception)[14-16] or other advanced machine learning algorithms [17-22]. But the neural node number in hidden layer are always fixed in one simulation, and the parameter configurations are modified though trial and error, its adaptive can't be ensured.

Current load forecast in EI mainly focuses on short term forecast [20-24]. This type of algorithm can be used in fault diagnosis, instant dispatch and system recovery, whose prediction accuracy largely influences the algorithm's robust and intelligent characters.

To alleviate the complexity and running delay of EI load forecast, ELM [25] (Extreme Learning Machine) liked neural network are introduced, which have characters like significantly reducing the running time delay and keeping comparable operation performance with those of complex networks. So, the introduction of block characters into adaptively deciding of ELM's training model size in every simulation step becomes an effective means, which can stir more innovations in this research domain.

II. PROPOSED ALGORITHMS

We utilizing different block characters to realize one-step further real time load forecast in EI system. The proceeding of this algorithm is shown as follows in Fig.1, and it includes following modules.

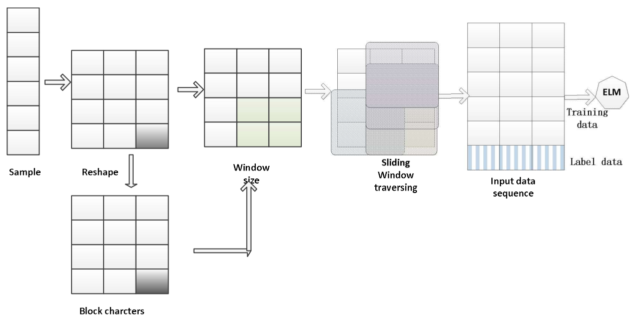


Fig. 1. The proceeding of this algorithm

A. Data Sampling Module

Before simulation, we need to get the history load data. Here we adopt a yearly load record for one industry park, and the original data sample frequency is one minute per sample. To reduce the algorithm's complexity, we get the mean value of data samples for every 15 minutes, so we can obtain 96 samples in one day, and total 35136 samples in one year (366 days). From

Fig.2 we can see, the load data have different characters at different time positions, which also have many exception data points (peak value or valley value) randomly distributed, which makes the traditional algorithms like neural network more hardly affordable for high precision forecasting demand due to irregular data distribution.

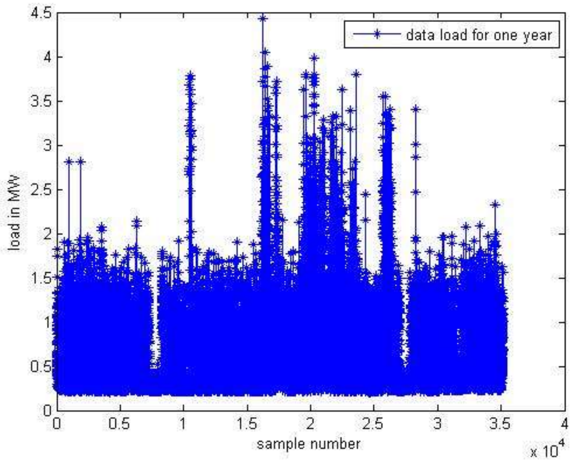


Fig. 2. load data in one year

B. Data Processing Module

a) Start index calculating module

We divide the load data into two sections: data sequence in training section and testing section. As the day sample number is 96, we fetch 96 last samples from the training section used as the test data sequence. And for every fetched data sample M in 96 samples， we calculate its corresponding start index (also named as character index) as $N = M - floor\left(\frac{M}{96}\right) * 96 + 1$,then we reshape the data sequence with index segments N:M into the data array with size $\left(\frac{M-N+1}{96}, 96\right)$ (row order first), which is notated as the preprocessing block L, and it's the source pool of every training data sequence.

b) Window size calculating module

For every block with its last data sample （from left to right, from top to bottom） located at the bottom right corner of block L (so its size varies), we calculate related block characters for every traversing block $(L((M-N+1)/96 - m1 + 1:(M-N+1)/96, 96 - n1 + 1:96))$, (characters may be the combined expressions of block mean value, standard deviation value and the block entropy, such as $mean + \sqrt{var}$, $mean + 3 * \sqrt{var}$, $mean + \sqrt{var} + entropy$, and so on), whose corresponding value is stored in $block\_e$ with the coordination index $(m1, n1)$ which represents its training window size, then we choose the coordination of data element in $block\_e$ with the minimum block element value which is bigger than the defined threshold calculated with certain factor, it can be expressed as:

$$max11 = \max(max(block\_e));$$

$$min11 = min(min(block\_e));$$

$$block\_f = min\,(min\,(block\_e(block\_e \geq k1 * (max11 - min11) + min11)));$$

$$[line1, col1] = find(block\_e == block\_f);$$

$$line(m) = line1(1);$$

$$col(m) = col1(1);$$

Where k1 is the threshold factor, which is between 0 and 1 (lower or equal than 0.2 in this experiment), and the coordination of selected element, i.e. $\big(line(m), col(m)\big)$, is chosen as the parameters of sliding window size with the elements of block $L((M-N+1)/96 - line(m) + 1:(M-N+1)/96, 96 - col(m) + 1:96)$.

c) ELM training module

For every test sample with original location $m2$,We calculated its related start index $n2 = m2 - floor(m2/96) * 96 + 1$, and took $(line(n2), col(n2))$ calculated in above step as its corresponding sliding window size.

We use the sample window with size $(line(n2), col(n2))$ traversed along the bottom line and right column of related reshaped matrix obtained from the data sequence (except the corner sample), and reshape the data array obtained in the window into one row of matrix H with size $(1, line(n2) * col(n2))$(row order first) ,which will form matrix H with size $(n, line(n2) * col(n2))$. After traversed all samples, H is transposed, then the bottom line of H is used as neural network output label, and other lines of H as neural network input data set.

We obtain the prediction input data block with a window size $(line(n2), col(n2))$ within transformed array L formed for this test sample, where the end location of the obtained block, i.e. its right bottom corner, is coincide with that of array L. And then, the block is reshaped which transforming into a vector with size$(1, line(n2) * col(n2))$, then this vector is also transposed. The last element of this vector is used as the real test data （for comparison), while other elements are used as neural network's prediction input data.

We input the output label and input data set along with prediction input data into ELM model for training and testing, which can predict the target value at the time location of real test data.

C. Performance Evaluation Module

We predict the load value with different parameter configurations, and evaluate the corresponding performance index as:

$$mse1 = sqrt(sum((predict - input\_1).* (predict - input\_1))/test\_count)/max\_in$$

where predict and $input\_1$ represents predicted load data sequence and real load data sequence, $test\_count$ represents the element number of the two sequences (size 200), max_in represents the max load data value. The less value of mse1, the more accurate prediction of the result.

III. SIMUATION RESULTS

A. Software Configuration

The simulation is running in MatLab 2014a and windows 10., with hardware setting of Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, 1.80 GHz.

B. Programming Length Setting

We estimate the load using the load data sequence just lying before results' location, which can be seen as a real time estimation and will have better precision performance than other time scales, the predicted load number is set as 200, and the whole data length is 35136, and the training data size is set as 34036.

C. Result analyse

We choose 6 block characters expressions as: $mean + 3 * \sqrt{var}, mean + \sqrt{var}, mean + 0.5 * \sqrt{var}, mean - 0.5 * \sqrt{var}, mean + 3 * \sqrt{var} - entropy$, $mean + \sqrt{var} + entropy$, where mean represents the mean value of selected block, var represents the variance of this block, and entropy represents the entropy value of this block, (equal interval division, with 20 levels). Then we choose the threshold factor k1 as 0.2, 0.1, 0.05, 0.02, 0.01, 0 for every block character.

a) Result analyse I

Here below table shows the results of above character expressions, whose best result as the minimum value of every line is notated with bold figures.

TABLE I. THE PERFORMANCE OF DIFFERENT CHARACTER EXPRESSIONS

$k1$ / mse	0.2	0.1	0.05	0.02	0.01	0
$mean - 0.5 * \sqrt{var}$	0.0688	0.0695	0.0639	0.0645	0.0665	**0.0635**
$mean + 3 * \sqrt{var} - entropy$	0.0615	0.0604	0.0575	0.0547	0.0579	**0.0553**
$mean + 3 * \sqrt{var}$	0.0568	0.0455	0.0413	0.0434	0.042	**0.0393**
$mean + \sqrt{var}$	0.0634	0.0527	0.0437	0.0439	0.042	**0.0408**
$mean + 0.5 * \sqrt{var}$	0.0694	0.0541	0.0456	0.0464	**0.0443**	0.0446
$mean + \sqrt{var} + entropy$	0.0417	0.042	0.0413	0.0419	0.0387	**0.0379**

From the simulation we can see, that for almost all block characters, the best performance usually located at $k1 = 0$ except for $mean + 0.5 * \sqrt{var}$, whose performance at $k1 = 0$ is also closest to the minimum value. Here, $k1 = 0$ means that we taking the coordination with minimum value of related characters as the training model size. And the performance of expression with addition signature is better than that with minus signature, so below figures we only consider addition signature. In these character expressions, $mean + \sqrt{var} + entropy$ shows the best results for all different $k1$, while it also has the most running time in table I.

b) Result analyse II

Below figures show the best simulation results for $mean + \sqrt{var} + entropy, mean + 3 * \sqrt{var}, mean + \sqrt{var}$, and $mean + 0.5 * \sqrt{var}$. And we compared the results with pure ELM, ESN (Echo state network) and traditional BPN (Back Propagating Network). Here simulated ELM windows' sizes were 6*6, 8*8, and 10*10. ESN's training window size is 150. The input data dimension of BPN is set as 5*5. It has two hidden layers with corresponding node number 21 and 5, its activation functions in turns are 'purelin', 'logsig' and purelin'.

From the results of Figs. 3-9, we can get below conclusions.

1) The raw data curve is a little random disturbed and is not so cyclical distributed especially at the peak section, so the accuracies of the estimation results can't be very high in many block characters.

2) From above results we can see that the predicted curve is more periodic repetition than that of the raw data, which is the basic characters of the neural network training, and is the source of prediction improvement, which deserves more deepen research.

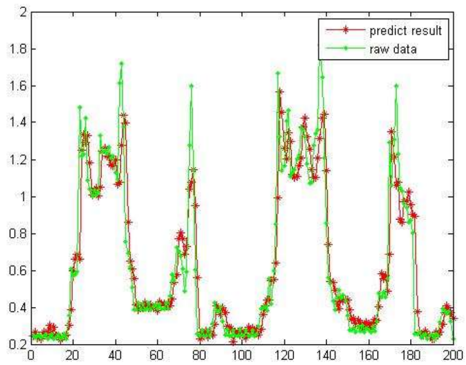


Fig. 3. Related results with $mean + \sqrt{var} + entropy$, mse =0.0379

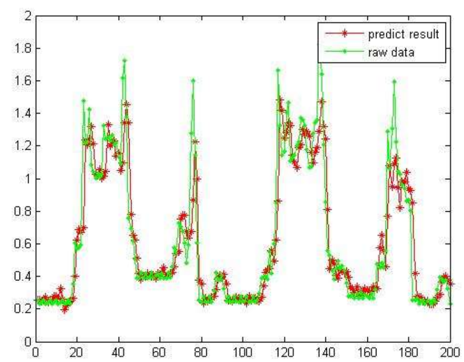


Fig. 4. Related results with $mean + 3 * \sqrt{var}$, mse =0.0393

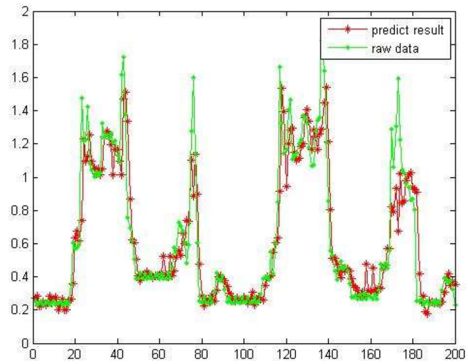


Fig. 5. Related results with $mean + \sqrt{var}$, mse =0.0408

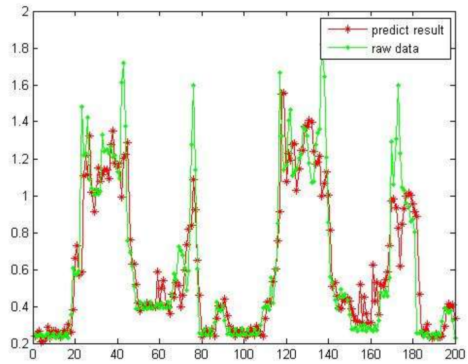
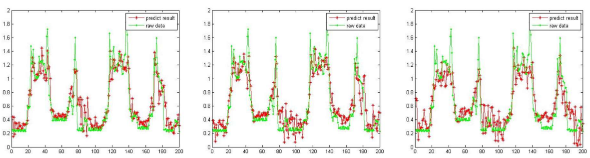


Fig. 6. Related results with $mean + 0.5 * \sqrt{var}$, mse =0.0446



(a) (b) (c)

Fig. 7. Pure ELM results, (a) line=col=6, mse=0.0455 (b) line=col=8, mse=0.0461 (c) line=col=10, mse=0.0532

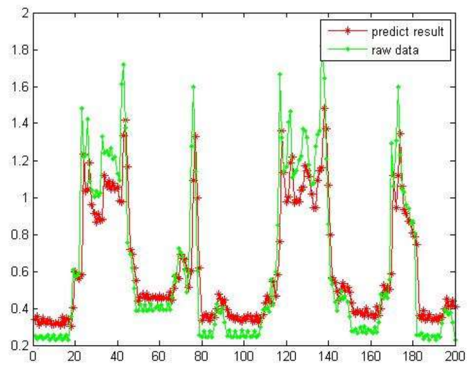


Fig. 8. Pure ESN results, washout=150, mse=0.0443

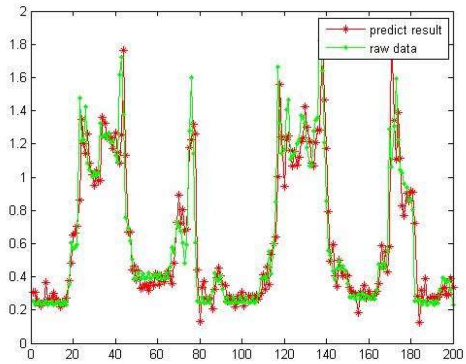


Fig. 9. BPN results, mse=0.0391

3) The results of our algorithm with addition signature usually have better performance than ELM algorithm with fixed training window size, which shows the advantages of our advanced algorithm.

4) The results of our algorithm with addition signature are better or comparable than the typical ESN in most scenes, both algorithms have quick running time when not considering block entropy.

5) The slightly inferior performance compared to our initial expectation may be due to unobservable samples' curve characters appeared in the training set, under these conditions, some peak value can't be predicted accurately. Its tendency of prediction result differs, too.

6) The result of BPN algorithm' s prediction accuracy is approaching that of best result of $mean + \sqrt{var} + entropy$, but its total executing time is 4-6 times more than that of this type of block characters.

7) For most figures of our proposed algorithm, the bottom section of the raw data curve can be more accurately predicted than that of ELM and ESN. And the peak section of our predicting curve shows some latency compared to the raw data curve, which also has apparent lower value of estimated data sequence than the raw data sequence in the peak section accompanied with different fluctuating curve shapes.

8) Through deepen observing, the prediction error of ESN has some complement characters with ELM (such as different peak or valley forecasting error characters, and elements addition of two algorithms may reduce the errors), so its performance can be further improved by subtly combining different algorithms like using some types of thresholds for algorithm selection and/or weight averaging for prediction results.

9) By utilizing more training data (blocks in more bottom lines and more right columns being used for block training), although the running time increased, the performance is not apparently increased, which may be due to the less correlations of the raw data at different time windows.

IV. Further Research Directions

We need do below research works further.

1) Extending the neural network type and related models, other new real-time and low complex neural network model could be considered, which may further promote the algorithm's performance.

2) Optimizing the threshold setting, to show the superior of block characters in load forecast, threshold expression should be carefully optimized and consider more block characters, as it takes a vital function in our proposed algorithm.

3) More subtle operation proceeding, we need to further design more subtle operation proceedings to deep mine the advantages of block characters in load forecast and introduce more technique innovations based on the block characters like information entropy.

4) Considering environment data in this simulation, as weather data may further improve the performance of our algorithm [26], adding it into the simulation is a necessary step for future research. So, we need to integrate related weather data into the application scenes.

5) Comparing with more advanced algorithms, here we just compare the algorithm with typical algorithms like ELM, ESN and BPN. In the future, we will consider more advanced algorithms to verify the algorithm's performance.

6) Trying more steps further load forecast. We will modify the block structure to realize multi-steps forecast while keeping the advantages of block characters. So, improved block structure may be needed.

V. Conclusions

To improve the performance of load forecast in complex scenes like EI, we adopt the block characters in ELM neural model training window size selection, where minimum value with addition signature in block characters expression is more suited for ELM prediction. By selecting threshold factor intelligently and designing more rational factor threshold, the prediction precision can be largely improved. And the result analyses show great application potentials, which can be effectively used in EI application scenes such as source-network-load-storage collaboration optimizing.

Acknowledgment

This work is sponsored by Tsinghua-Toyota Joint Research Institute Inter-disciplinary Program.

References

[1] J. W. Cao, M. B. Yang, D. H. Zhang, Y. Y. Ming, and K. Meng, “Energy internet: an infrastructure for cyber-energy integration,” Southern Power System Technology, vol.8, no.4, pp. 1-10, 2014. (In Chinese)

[2] J. Y. Wang, K. Meng, J. W. Cao, Z. H. Cheng, L. C. Gao, and C. Lin, “Information technology for energy internet: A survey,” Journal of Computer Research and Development, vol.52, no.5, pp. 1109-1126, 2015. (In Chinese)

[3] J. W. Cao et al., “Information system architecture for smart grids,” Chinese Journal of Computers，vol.36, no.1, pp.143－167, 2013．(In Chinese)

[4] A. Baliyan, K., Gaurav, and S. K.Mishra, “A review of short term load forecasting using artificial neural network models,” Procedia Computer Science, 48, 121-125, 2015.

[5] H. Sun, D. Duan, H. Zhang, S. Choi, J. R. Luo, and L. Yang, “Regional medium-term hourly electricity demand forecasting based on LSTM,” 2023 IEEE Power & Energy Society General Meeting (PESGM), Orlando, FL, USA, 2023, pp. 1-5.

[6] J. Manju, R. B. Manjula, and R. Dash, “An efficient artificial bee colony based optimized model for load prediction in IoT enabled smart grid,” 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2023, pp. 129-136.

[7] A. Ahmad, N. Javaid, M.Guizani, N. Alrajeh, and Z. A.Khan, “An accurate and fast converging short-term load forecasting model for industrial applications in a smart grid,” IEEE Transactions on Industrial Informatics, 13(5), 2587-2596, 2017.

[8] M. Jayashankara, P. Shah, A. Sharma, P. Chanak, and S. K. Singh, “A novel approach for short-term energy forecasting in smart buildings,” in IEEE Sensors Journal, vol. 23, no. 5, pp. 5307-5314, 1 March1, 2023.

[9] X. Sun et al., “Multi-energy load forecasting for integrated energy systems based on causal-wavelet neural networks,” 2024 IEEE 2nd International Conference on Power Science and Technology (ICPST), Dali, China, 2024, pp. 1773-1778.

[10] A. Kaur, L. Nonnenmacher, and C. F. M. Coimbra, “Net load forecasting for high renewable energy penetration grids,” Energy, 114(nov.1), 1073-1084, 2016.

[11] V. Prema and K. U. Rao, “Predictive models for power management of a hybrid microgrid — A review,” 2014 International Conference on Advances in Energy Conversion Technologies (ICAECT), Manipal, India, 2014, pp. 7-12..

[12] K. Shahryari and A. Anvari-Moghaddam, “Demand side management using the Internet of energy based on fog and cloud computing,” 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 2017, pp. 931-936.

[13] N. Lin et al., “Medium- and long-term load forecasting of residential user groups based on graph convolutional neural network and DBSCAN clustering,” 2024 IEEE 2nd International Conference on Power Science and Technology (ICPST), Dali, China, 2024, pp. 2030-2034.

[14] D. L. Marino, K. Amarasinghe, and M. Manic, “Building energy load forecasting using deep neural networks,” IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 2016, pp. 7046-7051.

[15] J. Llanos, D. Sáez, R. Palma-Behnke, A. Núñez, and G. Jiménez-Estévez, “Load profile generator and load forecasting for a renewable based microgrid using self organizing maps and neural networks,” The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, QLD, Australia, 2012, pp. 1-8.

[16] I. M. Coelho, V. N. Coelho, E. J. D. S. Luz, L. S. Ochi, F. G. Guimares, and E. Rios, “A gpu deep learning metaheuristic based model for time series forecasting,” Applied Energy, 412-418, 2017.

[17] Z. Ming-Yue, “A new method for short-term load forecasting based on fractal interpretation and wavelet analysis,” International Journal of Electrical Power & Energy Systems, 2015.

[18] V. N. Coelho, I. M. Coelho, B. N. Coelho, A. J. R. Reis, R. Enayatifar, and M. J. F. Souza, et al., “A self-adaptive evolutionary fuzzy model for load forecasting problems on smart grid environment,” Applied Energy, 169(may1), 567-584, 2016.

[19] B. Zhao, L. Stankovic, and V. Stankovic, “On a training-less solution for non-intrusive appliance load monitoring using graph signal processing,” in IEEE Access, vol. 4, pp. 1784-1799, 2016.

[20] Wang Xin and Liu Junjie, “Short-term load forecasting based on bayes' theorem support vector machine,” The Journal of New Industrialization, 2014, 4 (12): 20-24. (In Chinese)

[21] Li, Y, “Short-term load forecasting based on Bayes and RS,” In: Zeng, Z., Wang, J. (eds) Advances in Neural Network Research and Applications. Lecture Notes in Electrical Engineering, vol 67. Springer, Berlin, Heidelberg, 2010.

[22] T. S. Janjanam, K. S. Siram, and P. K. Kollu, "Cloud resources forecasting based on server workload using ML techniques," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 427-433.

[23] S. Subbayya, J. G. Jetcheva, and W. -P. Chen, "Model selection criteria for short-term microgrid-scale electricity load forecasts," 2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 2013, pp. 1-6.

[24] Z. Yu, H. Du, Y. Lei, and M. Shan, "Short-term load forecasting of power system based on weighted improved adaptive neuro-fuzzy inference system and random forest," 2024 IEEE 2nd International Conference on Power Science and Technology (ICPST), Dali, China, 2024, pp. 2024-2029.

[25] W. Jiahui, M. Weishi, and Z. Zhiyu, "Short-term load forecasting based on GA-PSO optimized extreme learning machine," 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 2018, pp. 1-9.

[26] C. Na and F. Haoshuo, "Short-term load forecasting for microgrid considering weather characteristics and source-load correlation," 2024 IEEE 2nd International Conference on Power Science and Technology (ICPST), Dali, China, 2024, pp. 1160-1169.