Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org

Volume 26, Issues 3, pp. 1222–1230, DOI 10.12694/scpe.v26i3.4118

LOAD DEMAND PREDICTION BASED ON IMPROVED ALGORITHM AND DEEP CONFIDENCE NETWORK

WEI XU[*], YI YU [†], YAQIN QIAN [‡], XU HUANG [§], MING ZHANG [¶], AND ZHONGPING SHEN [‖]

Abstract. To address the issue of inaccurate load forecasting amidst the advancing smart grid technology and the widespread integration of various demand-side resources like controllable loads, distributed energy sources, and energy storage, the author proposes a deep confidence network based on improved algorithms for load demand forecasting. Firstly, the VMD algorithm is used to decompose the load data into different intrinsic mode functions (IMFs), Then combine the DBN network to predict each IMF, Finally, overlay the prediction results of each part to obtain the prediction results of the VMD-DBN model. The experimental results indicate that: The PSO-DBN model has good prediction results and fast convergence speed in power load forecasting. The MAPE is 1.03%, and the RMSE is 9.35MW, which verifies that the method has good prediction accuracy. Compared to the single use of DBN method and the combination of Empirical Mode Decomposition (EMD) DBN method, the proposed method by the author has a significant improvement in prediction accuracy.

Key words: Short term load forecasting, Generalized demand side resources, Deep belief network, Load aggregator

1. Introduction. Short term load forecasting, as the foundation of daily power grid planning, is an indispensable and important part of safe power operation [1]. With the reform and development of electricity, improving the accuracy of short-term load forecasting has become the primary task of researchers in this field. It is of great significance for how to arrange scheduling plans, ensure reliable operation of power systems, and maximize economic benefits [2]. The past is a prophet of the future, and the basic principle of forecasting lies in fully learning the past. Therefore, the premise of load forecasting is also to predict the historical data of regional electricity loads. The amount of historical data required varies depending on the prediction period (the two are usually in a certain positive proportion), and combined with local politics, economy, meteorology, social events that affect electricity consumption, and other factors that can significantly affect electricity use, explore the necessary potential connections between changes in electricity loads and these influencing factors, in order to find a certain future development law and trend of electricity loads [3].

Short term load forecasting is of great significance for the optimal combination of units, economic dispatch, and optimal power flow of the dispatch department, especially for the current and future electricity market. Precise load forecasting enables strategic scheduling of power generation units, optimizing their efficiency and the economic viability of grid operations. This, in turn, fosters stability and security within the power grid. Within the smart grid framework, the integration of controllable loads, distributed energy sources, energy storage, respond to demand in a flexible and diverse manner, enhancing load transfer capabilities and expanding the time range for transfer [4]. In the electricity market environment, users adjust controllable loads, distributed power sources, and energy storage resources reasonably based on different price signals and incentive mechanisms with the goal of electricity economy, changing load characteristics and changing patterns. In short-term electricity market forecasting, it's crucial to factor in diverse demand-side resources to enhance accuracy. As the power grid expands and information technology advances, smart grid dispatch systems are becoming more adept at collecting vast amounts of data on loads and related factors, fueling exponential growth in data availability [5]. However, the load forecasting methods in the above literature are mostly shallow three-layer networks, which

[*]School of Common Courses, Wannan Medical College, Wuhu, 241000, China. (WeiXu966@163.com)
[†]Anhui ARN Group Co., LTD, Anqing, 246000, China. (Corresponding author, YiYu6375@126.com)
[‡]School of Common Courses, Wannan Medical College, Wuhu, 241000, China. (YaqinQian9@163.com)
[§]School of Common Courses, Wannan Medical College, Wuhu, 241000, China. (XuHuang3@126.com)
[¶]School of Common Courses, Wannan Medical College, Wuhu, 241000, China. (MingZhang16@163.com)
[‖]School of Common Courses, Wannan Medical College, Wuhu, 241000, China. (MingZhang16@163.com)

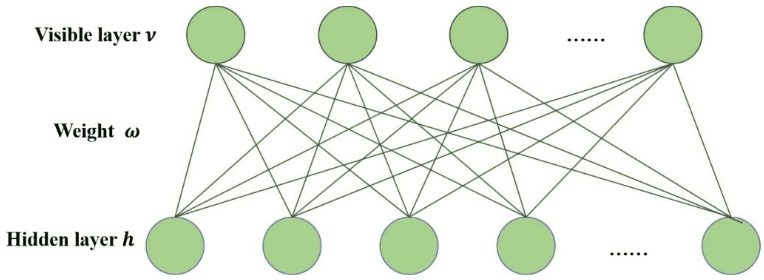

Fig. 3.1: RBM Structure

are difficult to handle the massive load dataset of the power grid today. Deep belief network (DBN) is an efficient deep learning algorithm composed of several stacked restricted Boltzmann machines (RBMs), which can be used for unsupervised learning and effectively process large power load data [6,7].

2. Literature Review. The flexible scheduling of controllable loads, distributed power sources, and energy storage, which are widely connected and participate in the electricity market, will inevitably affect the changes in electricity demand. Therefore, the author first constructs a contract based generalized demand side resource optimal scheduling model for three controllable resources: load curtailment (LC), load shift (LS), and energy storage (ES) [8]. This model aims to maximize the revenue of load aggregator (LA) and, under various constraints of the contract, solves the optimal scheduling strategy for generalized demand side resource participation in the electricity market based on real-time electricity prices. Yu, M. et al. introduced a short-term load forecasting model merging Fuzzy Exponential Weighting (FEW) with Improved Harmonic Search (IHS) algorithms. They validated the model's accuracy using fitness functions as evaluation criteria. Error analysis showed the model's effectiveness in predicting short-term electricity load data with strong stability and precision, offering valuable insights for implementing short-term forecasting in various industrial sectors[9]. To address data imbalance in ultra-short-term AC load forecasting, Tian, Z. et al. introduced a resistance-capacitance model featuring a two-phase parameter identification scheme[10]. In terms of short-term load forecasting in the power system, Jian, L. I. et al. tested a mutation model of RNN-LSTM. It effectively solves the problem of gradient explosion and disappearance caused by inputting a large amount of data in classical RNN [11]. Gao, W. et al. crafted a short-term load forecasting framework specifically tailored to accurately predict the cooling load of office buildings. They validated the framework's performance by assessing its ability to predict cooling loads, highlighting the importance of identifying key input features to enhance predictive accuracy[12].

Deep Belief Networks (DBN) are unsupervised learning models that converge faster and have higher prediction accuracy compared to traditional BP neural networks. A load forecasting algorithm based on DBN was proposed, combined with the fast optimization ability of particle swarm optimization, to achieve fast and accurate prediction of missing power data. Perform VMD on load data to obtain multiple sub sequences with distinct features, combine them with DBN prediction, and overlay each prediction result. By comparing the prediction results of a single DBN method and an EMD DNN prediction method, it was verified that the proposed method can explore the potential patterns of load data, reduce the computational scale, and reduce the generation of false IMFs, thereby improving prediction accuracy.

3. Method.

3.1. Restricted Boltzmann Machine. Deep Belief Networks (DBNs) consist of multiple Restricted Boltzmann Machines (RBMs) stacked together. Each RBM in the model comprises interconnected hidden and visible layers arranged in a random combination. The connections between units in the hidden and visible layers are fully linked, facilitating robust unsupervised learning for data. The network structure is depicted in Figure 3.1, illustrating the absence of connections within individual hidden and visible layers.

If RBM uses v to represent the visible layer and h to represent the hidden layer, then the energy equation of the system is

$$E(v,h|\theta) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_i w_{ij} h_j \tag{3.1}$$

In the formula, v_i represents the state of visible layer unit i; h_j is the state of hidden layer unit j; a_i is the bias of visible layer unit i; b_j is the bias of hidden layer unit j; The number of all units in the visible layer m; n is the number of all units in the hidden layer; w_{ij} is the connection weight between units i in the visible layer and units j in the hidden layer; θ is the set of all parameters $\{a_i, b_i, w_{ij}\}$.

The joint probability distribution of a given state is

$$P(v,h|\theta) = \frac{e^{-E(v,h|\theta)}}{Z(\theta)} \tag{3.2}$$

In the formula: $Z(\theta)$ is the partition function, represented as $Z(\theta) = \sum_v \sum_h e^{-E(v,h|\theta)}$.

Due to the independence between the units in the visible layer and the units in the hidden layer of RBM, its conditional probability distribution is:

$$P(v|\theta) = \frac{\sum_h e^{-E(v,h|\theta)}}{Z(\theta)} \tag{3.3}$$

$$P(v|\theta) = \frac{\sum_v e^{-E(v,h|\theta)}}{Z(\theta)} \tag{3.4}$$

RBM adopts unsupervised greedy training algorithm for parameter training, with the training objective of maximizing the logarithmic likelihood function of the model, a_i, b_j, and $lgP(v|\theta)$. By taking partial derivatives of the likelihood function and combining Gibbs sampling, the updated iteration formulas for the parameters a_i, b_i and w_{ij} can be obtained as follows:

$$\Delta a_i = \epsilon(< v_i >_{data} - < v_i >_{recon}) \tag{3.5}$$

$$\Delta b_i = \epsilon(< h_i >_{data} - < h_i >_{recon}) \tag{3.6}$$

$$\Delta w_{ij} = \epsilon(< v_i h_j >_{data} - < v_i h_j >_{recon}) \tag{3.7}$$

In the formula, $< \cdot >_{data}$ represents the mathematical expected value of the model distribution; represents the mathematical expected value of the distribution after further reconstruction of the model; $< \cdot >_{recon}$ ϵ is the learning rate [13,14].

3.2. Deep Confidence Network. DBN consists of multiple RBMs arranged in layers, with each RBM's hidden layer serving as the visible layer for the subsequent RBM in the stack. This hierarchical structure, depicted in Figure 3.2, facilitates the learning of increasingly abstract representations of data as it progresses through the network [15]. DBN adopts a greedy layer by layer training algorithm to complete the cognitive and generation process of the model from bottom to top, and then completes backpropagation training and weight fine-tuning from top to bottom through feedback learning of the classic BP neural network at the top.

3.3. Training models and learning algorithms. The time series prediction model adopted by the author is based on the DBN neural network algorithm. The model uses a DBN algorithm composed of multiple RBMs stacked together to perform forward unsupervised learning on the initial weights and thresholds. The greedy layer by layer training algorithm iteratively optimizes the various initial parameters of the training model, and then fine tunes the model parameters through classical feedback learning, so that the training results converge to the optimal. The model training process is shown in Figure 3.3.

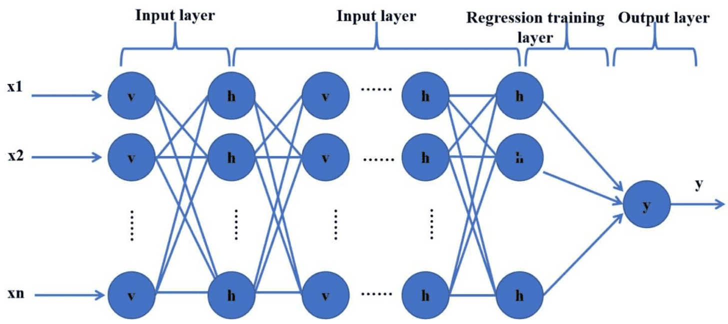

Fig. 3.2: DBN Model Structure

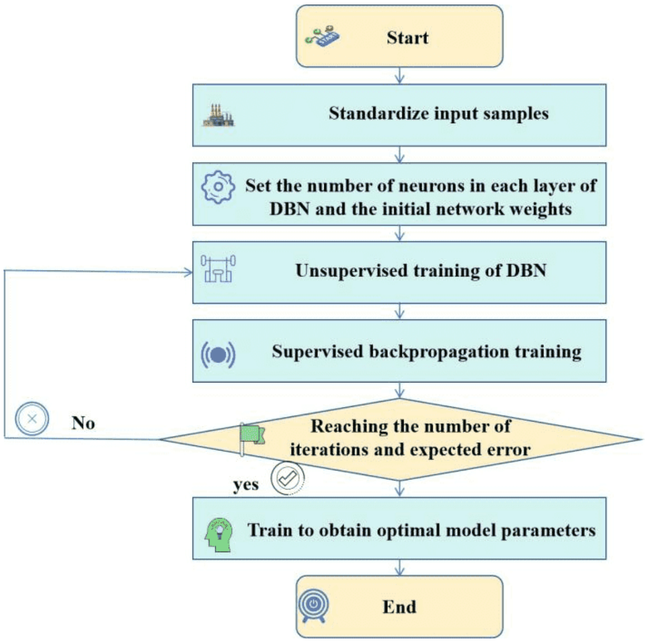

Fig. 3.3: DBN model training process

According to the temporal characteristics of load data, the experimental data is treated as a set of temporal data for model training. Assuming that the algorithm model has the $i^{\#}$-th input variable x_i^* and the i^*-th output variable y_i^*=x(t) at time t, among them, x(t) represents the timing value of the current time t, which is to use the value of that time as the output and the value of the previous $\hat{t}$ time periods as the input for prediction, that is

$$x_i^* = [x(t-\hat{t}), x(t-\hat{t}+1), \cdots, x(t-2), x(t-1)] \tag{3.8}$$

The specific training steps for the prediction model are as follows:

Step 1: Analyze and process the original power load data, and use the standard score formula to normalize the data to the [0,1] interval, as shown in equation 3.9. The normalized data can to some extent accelerate the convergence speed of the model and improve its accuracy.

$$x^* = \frac{x-\overline{x}}{\sigma} \tag{3.9}$$

In the formula, x^* represents the normalized data; x is the original experimental data; σ is a vector where each element represents the mean of the original data; is the standard deviation of the original experimental data.

Step 2: Use a DBN model stacked from multiple RBM models for data training, set the number of hidden layers n, and the learning rate of the BP neural network model during backpropagation training ϵ_{bp} and momentum factor α, and provide the number of DBN training times and the number of backpropagation algorithm training times. The condition for exiting the model iteration is to reach the maximum number of iterations or the expected error.

Step 3: To accomplish the unsupervised learning process in the DBN model, a greedy layer-by-layer training algorithm is employed. As the efficiency of Gibbs sampling diminishes with more sampling steps, the author opts for the Contrastive Divergence (CD) algorithm, introduced by Hinton, for swift parameter estimation.

The CD algorithm can obtain sufficiently good training model parameters through a one-step sampling method [16]. Based on the symmetric structure and independence of the model, the activation probability $P(h|v_0)$ and the initial state h0 of the hidden layer are obtained by using the initial state v_0 of the visible layer. After a step of Gibbs sampling, v_1 and h_1 can be obtained based on the initial state of the model. The specific sampling process is as follows.

The sigmoid function is used as the excitation function between the neurons in the hidden layer of the model to standardize the data. The formula for the processed function is

$$\hat{\sigma}(\dot{y}) = \frac{1}{1+e^{-y}} \tag{3.10}$$

In the formula, $\dot{y}$ represents the data to be subjected to sigmoid standardization processing.

From this, we can obtain the activation probability distribution of the visible layer and the hidden layer when they are turned on:

$$P(h_j = 1|v, \theta) = \hat{\sigma}(b_j + \sum_i v_i w_{ij}) \tag{3.11}$$

$$P(v_j = 1|h, \theta) = \hat{\sigma}(a_j + \sum_i w_{ij} h_j) \tag{3.12}$$

Finally, the various training parameters of the model can be updated according to equations 3.5-3.7 and 3.13-3.15 as follows:

$$w_{ij}^{k+1} = w_{ij}^k + \Delta w_{ij} \tag{3.13}$$

$$a_{ij}^{k+1} = a_i^k + \Delta a_i \tag{3.14}$$

$$b_{ij}^{k+1} = b_j^k + \Delta b_j \tag{3.15}$$

In the formula, k represents the number of iterations.

3.4. Prediction Model Based on Particle Swarm Optimization. The Particle Swarm Optimization (PSO) algorithm finds wide application in power data prediction tasks. It enhances the convergence speed and accuracy of training models by ensuring they reach global optimal solutions. PSO optimizes and updates model parameters, simulating the collective foraging behavior of birds. In this analogy, each particle in the PSO model represents an individual bird, navigating the solution space by adjusting its position and movement speed to find local optimal solutions. The particle swarm obtains a global optimal solution by sharing information between each particle. Assuming that the position of the k-th iteration of the particle swarm is $Z_i^k = (Z_{b1}^k, Z_{b2}^k, \cdots, Z_{bd}^k)$, the speed of movement is $U_b^k = (U_{b1}^k, U_{b2}^k, \cdots, U_{bd}^k)$, the optimal position of each particle is represented as $P_b^k = (P_{g1}^k, P_{g2}^k, \cdots, P_{gd}^k)$, and the optimal position of the entire particle swarm is represented as $P_b^k = (P_{b1}, P_{b2}, \cdots, P_{bd})$, the update formula for the position and movement speed of each particle can be obtained as follows:

$$Z_{bd}^{k+1} = Z_{bd}^k + U_{bd}^{k+1} \tag{3.16}$$

$$U_{bd}^{k+1} = \omega U_{bd}^k + c_1 r_1 (P_{bd} - Z_{bd}^k) + c_2 r_2 (P_{gd} - Z_{bd}^k) \tag{3.17}$$

Building on the characteristics of the PSO algorithm, the author introduces the PSO-DBN model. This model optimizes the parameters further following the unsupervised training of DBN. The formula involves model weights, learning factor constants (c1 and c2), random numbers (r1 and r2) within [0,1], the number of particles (d), and the number of iterations (k). Through this approach, the PSO-DBN model refines the parameter settings, enhancing the overall performance of the system. The initial particle swarm position of the PSO model is trained using DBN parameters, and then PSO iteratively optimizes the regression training layer of the model. The training parameters are the connection weight w1 and bias of the first layer, respectively 1, as well as the connection weight w_2 and bias θ_2 of the second layer, the update formula for the training parameters is:

$$w_1^{k+1} = \begin{bmatrix} Z_{b1}^k & \cdots & Z_{bs_1}^k \\ \vdots & \ddots & \vdots \\ Z_{b(s_1 s_2 + s_1 - 1)}^k & \cdots & Z_{b(s_1 s_2 + 2s_1)}^k \end{bmatrix} \tag{3.18}$$

$$\theta_1^{k+1} = \left[Z_{b(s_1 s_2 + 2s_1 + 1)}^k \cdots Z_{b(s_1 s_2 + 3s_1)}^k \right] \tag{3.19}$$

$$w_2^{k+1} = \left[Z_{b(s_1 s_2 + 2s_1 + 1)}^k \cdots Z_{b(s_1 s_2 + 3s_1)}^k \right] \tag{3.20}$$

$$\theta_2^{k+1} = Z_{b(s_1 s_2 + 3s_1 + 1)}^k \tag{3.21}$$

In the formula: k is the number of iterations; s_1 is the number of units in the first layer of DBN; s_2 is the number of units in the second layer of DBN [17,18]. The PSO-DBN model training flow is shown in Figure 3.4.

4. Results and Discussion. Modeling and analyzing the distribution network load data obtained from the project, mainly focusing on model training and prediction of historical values of distribution network load detection values. Due to the differences in data between different equipment points in substations, in order to make the prediction model universal for different point data in substations, the measurement data of a point in the database is selected as the experimental object, and the experimental data is normalized to improve the training speed and accuracy of the training model. In order to better evaluate the predictive accuracy of the prediction model in power load forecasting, two indicators, Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), were used to analyze the accuracy of the experimental results[19-20]. The specific indicators are calculated as follows:

$$\hat{E}_{RMSE} = \sqrt{\frac{\sum (x_e - x_e')^2}{n_e}} \tag{4.1}$$

$$\hat{E}_{RMSE} = \frac{1}{n_e} \sum \left| \frac{x_e - x_e'}{x_e} \right| \tag{4.2}$$

In the formula, x_e represents the real data; x_e' is the predicted data; n_e is the total number of data.

The simulation comparison results between two prediction models proposed by the author and the BP neural network model are shown in Table 4.1.

Figure 4.1 shows the convergence curves of three models. Based on the simulation results in Table 1 and the time complexity of the three models, it can be seen that the PSO-DBN model has good prediction results and fast convergence speed in power load forecasting. The MAPE is 1.03%, and the RMSE is 9.35MW, which verifies that the method has good prediction accuracy.

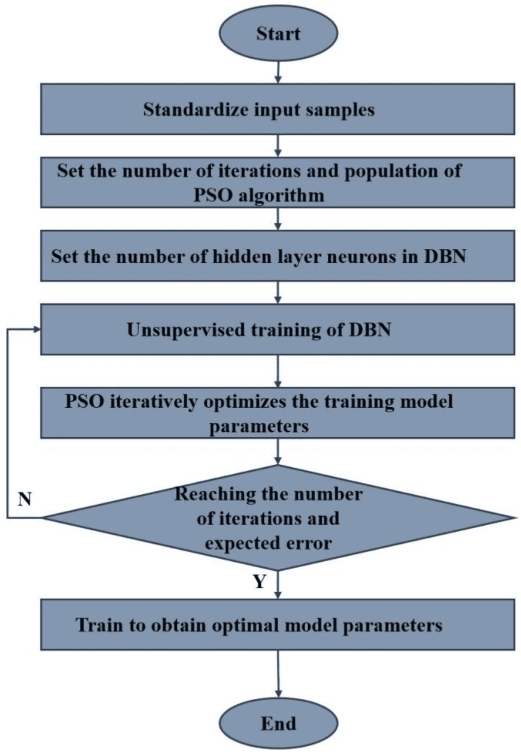

Fig. 3.4: PSO-DBN model training process

Table 4.1: Simulation Results

training model	RMSE	MAPE
BP	2.2338	0.0052
DBN	1.3063	0.0031
PSO-DBN	1.2763	0.0028

5. Conclusion. The author proposes a load demand prediction based on an improved algorithm using deep confidence networks. The participation of a large number of generalized demand side resources in the electricity market has higher requirements for short-term load prediction accuracy. At the same time, the massive dataset generated by the smart grid dispatch system provides a data foundation for the use of deep learning. Hence, the author initially integrates generalized demand side resources into market operations via load aggregation merchants, establishing a contract-based scheduling model for generalized demand side resources to derive optimal scheduling strategies. Then, the optimal scheduling scheme for generalized demand side resources is used as input for a load prediction model. In this paper, a DBN short term load forecasting model, which includes generalized demand side resources, is developed and compared with the BP neural network and the DBN model. The empirical results demonstrate the efficacy of the demand response resource scheduling model, centered on electricity price contracts, in maximizing revenue for load aggregators. It effectively adapts to real-time market electricity prices, determining optimal participation times for various resources. Moreover, integrating the optimal scheduling plan for generalized demand side resources into the prediction model proves advantageous, enhancing prediction accuracy and reducing errors.

Acknowledgement.

1. 2020 Anhui Province Quality Engineering Project: Exploration and Practice of Innovation and En-

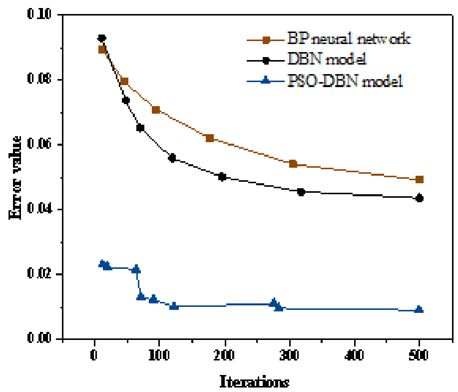

Fig. 4.1: Convergence curves of three models

trepreneurship Training of regional medical college students from the perspective of Yangtze River Delta Integration Development (2020jyxm2076).

2. 2022 Wannan Medical College Teaching Quality and Teaching Reform Project Research topic: Medical College moral education and Innovation and entrepreneurship education integration practice path and evaluation reform research (2022jyxm21).

REFERENCES

[1] Sun, F., Huo, Y., Fu, L., Liu, H., Wang, X., & Ma, Y. (2023). Load-forecasting method for ies based on lstm and dynamic similar days with multi-features. Global Energy Internet: English, 6(3), 285-296.
[2] Zhang, R., Yu, M., & Zhang, C. (2022). A similar day based short term load forecasting method using wavelet transform and lstm. IEEJ Transactions on Electrical and Electronic Engineering, 17(4), 506-513.
[3] Hao, L., Linghua, Z., Cheng, T., & Chenyang, Z. (2023). Short-term load forecasting model based on gated recurrent unit and multi-head attention. Journal of China University of Posts and Telecommunications: English Edition, 30(3), 25-31.
[4] Guo-Feng Fan, Li, Y., Xin-Yan Zhang, Yi-Hsuan Yeh, & Wei-Chiang Hong. (2023). Short-term load forecasting based on a generalized regression neural network optimized by an improved sparrow search algorithm using the empirical wavelet decomposition method. Energy Science And Engineering, 11(7), 2444-2468.
[5] Xiao, L. I., & Xianling, L. U. (2022). Ethod for forecasting short-term power load based on dual-stage attention mechanism and gated recurrent unit network. Computer Engineering, 48(2), 291-296.
[6] Zhang, R., Yu, M., & Zhang, C. (2022). A similar day based short term load forecasting method using wavelet transform and lstm. IEEJ Transactions on Electrical and Electronic Engineering, 17(4), 506-513.
[7] Xin, L., Haidong, S., Hongkai, J., & Jiawei, X. (2022). Modified gaussian convolutional deep belief network and infrared thermal imaging for intelligent fault diagnosis of rotor-bearing system under time-varying speeds:. Structural Health Monitoring, 21(2), 339-353.
[8] Su, S., Hu, G., Li, X., Li, X., & Xiong, W. (2023). Electricity-carbon interactive optimal dispatch of multi-virtual power plant considering integrated demand response. Energy Engineering (English), 120(10), 2343-2368.
[9] Yu, M., Zhu, J., & Yang, L. (2023). Short-term load prediction model combining few and ihs algorithm. Archives of Electrical Engineering, 9(2), 433-443.
[10] Tian, Z., Lin, X., Lu, Y., Song, W., & Niu, J. (2023). Imbalanced data-oriented model learning method for ultra-short-term air conditioning load prediction. Energy and buildings, 11(5), 695.
[11] Jian, L. I., Peng, Y., Cheng, Z., Yuwei, L. I., Fan, J., & Chen, J. (2022). Prediction and analysis on short-term load of power system based on lstm. Meteorological and Environmental Studies: English Version (004), 013.
[12] Gao, W., Huang, X., Lin, M., Jia, J., & Tian, Z. (2022). Short-term cooling load prediction for office buildings based on feature selection scheme and stacking ensemble model. Engineering computations: International journal for computer-aided engineering and software(5), 39.
[13] Zhang, R., Yu, M., & Zhang, C. (2022). A similar day based short term load forecasting method using wavelet transform and lstm. IEEJ Transactions on Electrical and Electronic Engineering, 17(4), 506-513.
[14] Dong, S., & Xia, Y. (2023). Network traffic identification in packet sampling environment. Digital Communication and Networking: English Version, 9(4), 957-970.

[15] Gowri, S., Subhashini, R., Mathivanan, G., Jabez, J., Vigneshwari, S., & Vimali, J. S. (2024). A descriptive framework for information retrieval using crawler based clustering and effective search algorithm. International Journal of Information Technology & Decision Making, 23(02), 993-1016.

[16] Kayij, E. N., Joél Lema Makubikua, & Busili, J. D. K. (2023). New hybrid algorithm based on bicriterionant for solving multiobjective green vehicle routing problem. American Journal of Operations Research, 13(3), 33-52.

[17] Babu, R. G., Amudha, V., Chellaswamy, C., & Kumar, K. S. (2022). Retracted article: lot based residential energy management system for demand side response through load transfer with various types of domestic appliances. Building Simulation (English), 15(9), 1.

[18] Chen, G., Huang, W. X., Ronch, A. D., Pecora, R., Kim, D., & Liu, Y., et al. (2023). Bp neural network-kalman filter fusion method for unmanned aerial vehicle target tracking:. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 237(18), 4203-4212.

[19] Yang, J., Zhang, L., Liu, G., Gao, Q., & Qian, L. (2022). Sintered silicon carbide grinding surface roughness prediction based on deep learning and neural network. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 44(7), 1-13.

[20] Ma, G., Wang, Z., Liu, W., Fang, J., Zhang, Y., & Ding, H., et al. (2023). Estimating the state of health for lithium-ion batteries:a particle swarm optimization-assisted deep domain adaptation approach. Journal of Automation: English Edition, 10(7), 1530-1543.

Edited by: Bradha Madhavan
Special issue on: High-performance Computing Algorithms for Material Sciences
Received: Jun 10, 2024
Accepted: Aug 2, 2024