Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

Research Article

Prediction of Renewable Energy Generation and Power Load Based on the Integrated Model of the Improved Differential Evolution Algorithm and LSTM Neural Network

Linmao Ma[1,2*], Jingyi Xiao[1,2], Qingli Zhang[3]

[1] School of Management, South-Central Minzu University, Wuhan 430074, China
[2] Research Center of Digital Development and Governance in Minority Areas, South-Central Minzu University, Wuhan 430074, China
[3] College of Management, Shenzhen University, Shenzhen 508060, China
E-mail: lmma16@126.com

Received: 30 August 2025; **Revised:** 20 September 2025; **Accepted:** 24 October 2025

Abstract: The power system plays a crucial role in social development and national livelihood security. Recently, with the aggravation of air pollution from traditional power systems, the proportion of Renewable Energy (RE) in the power system is increasing. However, because of the unstable nature of RE generation, predicting electricity load and renewable energy generation can help optimize energy use and improve the power system's efficiency, greenness, and economy. A neural network model is established to solve the prediction problem of electricity. Considering the challenges associated with hyperparameter tuning in neural network construction, an improved differential evolution algorithm is developed to automatically adjust the parameters. The paper constructs three neural network automatic parameter tuning models based on the original Differential Evolution (DE), the Differential Evolution based on Elite Variation (EVDE) and the Differential Evolution based on Elite Variation and Self-Attention (SAEVDE). The parameter tuning model is employed to optimize the hyperparameters of the Backpropagation Neural Network (BPNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) models, specifically focusing on batch size, hidden layer size, time step, and input dimension. All the models, with the automatic adjustment of the parameters, are validated through comparative experiments. The results show that SAEVDE-LSTM has the best performance with an accuracy of up to 98%, demonstrating that the improved differential evolution algorithm is effective for the optimization strategy of the neural network and that the prediction model can meet the needs of power companies. Finally, the self-adaptive LSTM model is applied to forecast the Danish power load and wind power generation to obtain the supplementary supply scheme for a power system with a large proportion of renewable energy power.

***Keywords*:** power load forecasting, renewable energy, differential evolution algorithm, neural network, prediction model

MSC: 68T05, 90C59, 93C40

1. Introduction

The power system is the foundation of the country's operation and plays a pivotal role in the development of the country and the security of people's lives. With the continuous development of the market economy and infrastructure,

Copyright ©2026 Linmao Ma, et al.
DOI: https://doi.org/10.37256/cm.7320268458
This is an open-access article distributed under a CC BY license
(Creative Commons Attribution 4.0 International License)
https://creativecommons.org/licenses/by/4.0/

the development needs of the power industry are also increasing. Climate change has resulted in frequent occurrences of extreme weather, leading to a sharp increase in residential and industrial electricity consumption. To deal with the core demand of national power consumption, the power system with Renewable Energy (RE) must have a forward-looking supply and demand supporting system. So, it is valuable to help formulate a RE generation plan so that the system can accurately predict the load and RE generation in a particular future period to gain a balance between the power demand and the RE generation. In recent years, the environmental pollution caused by traditional thermal power generation has gradually aroused social concern, leading to an increase in the proportion of renewable energy power in the power system. The prediction of renewable energy generation plays a critical role in optimizing energy utilization, thereby enhancing the efficiency and economic performance of power systems. In terms of energy supply and demand balance, the prediction scheme of renewable energy power generation also makes it possible to reasonably adjust the supply plan of traditional energy and reduce the dependence on traditional energy, which can reduce greenhouse gas emissions and further promote the sustainable development of energy. Given that renewable energy generation is strongly influenced by factors such as seasonal variations and weather conditions, which introduce additional uncertainty into the power system, accurately forecasting power load demand and renewable energy generation is essential for optimizing power generation strategies and promoting the green and sustainable development of the power industry.

The effect of the power intelligent system currently on the market also reflects that there is still room for improvement in the accuracy and time span of power load prediction. The study found that the basic forecasting model typically exhibits poor fitting performance, primarily due to the strong nonlinear characteristics of power load, which is inherently time-series data. More and more studies have shown that for nonlinear sequence data, the Artificial Neural Network (ANN) can usually present better fitting results, such as deep learning models such as backpropagation neural networks, convolutional neural networks, and recurrent neural networks. However, although the neural network model can usually solve nonlinear time series data such as electric load better, its design has become a significant difficulty. To achieve a better fit, the artificial neural network is usually designed to be very deep, but this will result in many parameters, which are easy to overfit and result in poor generalization ability. Therefore, the structure of ANN is very crucial to the solution of the task, so a self-adaptive hyperparameter selection mechanism is needed. In addition to artificially setting hyperparameters based on experience, many optimization algorithms can be applied to solve this problem. For example, the Metaheuristic algorithm is a heuristic random optimization algorithm. Many studies have also shown that such algorithms are effective for hyperparameter optimization. Therefore, this research seeks to optimize the hyperparameters of the neural network model through the application of a swarm intelligence algorithm, thereby facilitating a more intelligent design of the model to address the challenges associated with power load forecasting.

The primary purpose is to propose an effective prediction method of power load and renewable energy generation for the intelligent improvement of power systems with a large proportion of RE. The specific task is to optimize the hyperparameters of the neural network model through the swarm intelligence algorithm. With the self-adaptive hyperparameter selection mechanism, the neural network model is established based on the optimal combination of hyperparameters to predict the power load and renewable energy generation. The prediction model can provide a more reasonable basis for power companies to rationally arrange power generation plans and prioritize the use of renewable energy to meet the power needs of various sectors of economic production, which also reduces the air pollution caused by power generation. Therefore, the forecast model is constructed for power load and renewable energy generation through a fully connected neural network, a traditional recurrent neural network and a long-term short-term memory network. Moreover, the results are compared to find the proper neural network. Furthermore, the hyperparameters of the neural network are optimized through an intelligent algorithm. Due to the difficulty of the hyperparameters selection in training the neural network, the differential evolution algorithm is adopted to optimize the hyperparameters of the neural network. In addition, to improve the differential evolution algorithm, the original differential evolution algorithm is enhanced by two different methods. Then, the improved algorithm is validated through the solution of two testing functions.

2. Literature review

The power system plays a core role in the country's infrastructure and economic development, which plays a vital role in the security of national life. The structure and operation mechanism of the power system has also attracted much attention from scholars in many domains, where the prediction of power load is a primary issue of the operation and management of the power system. With the advancements in machine learning techniques, power load prediction has seen significant improvements and has emerged as a critical research area in recent years [1]. Power load forecasting is typically categorized based on the prediction horizon, which includes ultra-short-term, short-term, medium-term, and long-term forecasts. Short-term load forecasting, in particular, focuses on providing predictions with higher resolution and accuracy, where the time intervals for forecasting can be as short as 15 minutes [2]. In contrast to short-term forecasting, medium-term and long-term predictions typically have a time interval of one hour. Therefore, medium-term to long-term prediction often faces more significant challenges since the data is more susceptible to external environmental factors such as weather and holidays [3, 4]. In addition, power load prediction is applied in some particular scenes, such as the prediction of daily power load in a specific area [5], the prediction of the peak daily power load [6–8], and the forecast of the power load in an industrial park [9]. Moreover, there is also a power load prediction under specific circumstances. Liu et al. explored whether the traditional short-term load prediction method can accurately capture the load mode during the COVID-19 period [10].

In the past decade, driven by the emphasis on sustainable development, renewable energy generation has increasingly contributed to the power system, helping to mitigate air pollution from the energy sector. Meanwhile, the number of products using electricity as the leading energy is also growing, such as electric vehicles. Arias et al. [11] and Moon et al. [12] predicted the demand for power load for electric vehicles, while Cesaro et al. predicted the dispatching of green power in the power sector [13]. For the relationship between renewable energy generation and the power system, the development of renewable energy generation have significant impacts on system reliability [14]. Such research on renewable energy generation helps to design the plan and dispatch scheme of the renewable energy grid. Regarding the integration of renewable energy generation and dispatch, Arias et al. [15] proposed a novel planning method through a spatial load prediction model to guide the expansion and distribution of the power system with renewable energy. Since renewable energy generation highly fluctuates with the weather, Serrano-Guerrero et al. [16] proposed a practical implementation method to obtain data in a prediction period of 15 days, which can limit uncertainty and prove feasibility in applying the artificial neural network.

There are numerous methods used for power load prediction. The mainstream methods are primarily based on machine learning algorithms, which can be categorized into traditional machine learning and deep learning techniques. The main prediction methods based on conventional machine learning are linear regression, logistic regression, decision trees, random forests and support vector machines. Bracale et al. [17] established a multivariate quantile regression model to achieve power load prediction. Liu and Li [18] proposed the support vector regression method with a nonlinear semi-exponential loss function to enhance the accuracy of the power load prediction. Gao et al. [19] developed an improved random forest regression algorithm for ultra-short-term power load forecasting, utilizing the Gaussian mixture-based tree-structured Parzen estimator algorithm. Chen et al. [20] proposed a boosted clustering model based on the traditional k-means algorithm, which facilitates the clustering of consumers and enhances the accuracy of electricity consumption prediction. Furthermore, with the advancement of artificial intelligence technologies, several researchers have demonstrated that deep learning techniques offer promising solutions to the challenges of power load forecasting. The deep learning technique is a popular branch of machine learning that essentially processes unstructured data by building a more elaborate network with more layers, where Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) neural networks can deal with long sequences well for the prediction problem [21]. Kolokythas et al. [22] proposed an artificial neural network model to enhance the accuracy of wind power generation forecasting. In addition, the use of convolutional neural networks also is applied in this field. The time-based convolutional neural network was employed to forecast the timely power load, built using the idea of causal convolution with the achievement of temporal information processing [23–25]. Geng et al. [26] used the particle swarm algorithm for tuning the hyperparameter of the time convolution network to predict power load. Comparatively, recurrent neural networks are more proper to process such time-series data as power load, especially

for the Long Short-Term Memory (LSTM) neural networks widely used in this field [27–31]. Ageng et al. [28] proposed an hourly load forecasting framework based on the two-stack LSTM. Alonso et al. [32] presented a LSTM model capturing individual consumption patterns to achieve the prediction of electricity consumption. Besides the usage of the single neural network, the combination of different neural networks also is used to solve this issue [33–35]. Wan et al. [36] added the attention mechanism to convolutional neural networks and proposed a CNN-LSTM model to extract spatiotemporal features for power load prediction, which proved that the model performed well under different conditions. Qu et al. [37] assembled CNN, LSTM and attention mechanisms in a multi-correlation prediction model to achieve hourly PV power prediction.

Although neural network models have performed well in many domains, research on improving neural network algorithms has also gained increasing attention to achieve better performance. An alternative method of improvement of neural networks is to incorporate them with intelligent algorithms to optimize the hyperparameter combination of the network. Shakir et al. [38] applied a genetic algorithm to optimize the LSTM neural network for non-intrusive load prediction in smart microgrids. Similarly, Yu et al. [39] constructed parallel models for control chart recognition using GA-based CNN and LSTM. These studies demonstrate that the performance of the optimized neural network is superior to that with the corresponding hyperparameters randomly obtained. In addition, ensemble learning models that combine deep learning and other machine learning approaches have been widely applied in energy forecasting. For instance, some studies have proposed ensemble frameworks integrating Random Forest and XGBoost [40], hybrid machine learning approaches combining MLP, SVR, and CatBoost [41], and models that incorporate regional clustering with LSTM networks [42]. Ge et al. [43] improved K-means clustering to identify load characteristics across different regions, industries, and production modes, and proposed a hybrid forecasting method that integrates reinforcement learning and Particle Swarm Optimization to enhance Least Squares Support Vector Machines (PSO-LSSVM). Their results verified the high accuracy and practical value of the hybrid model. Song et al. [44] developed a Super Learner-based ensemble model for full-load power prediction in combined-cycle power plants, integrating six base models through cross-validation weighting. The results showed that boosting-based algorithms contributed most significantly to the ensemble performance. Goswami and Kandali [45] constructed a stacking model with neural networks as base learners and CatBoost as the meta-learner, demonstrating that machine learning methods substantially outperformed traditional statistical models.

It's evident that the swarm intelligence algorithm is the most commonly used among the optimization algorithms used in the mentioned research above. In fact, these algorithms were initially used to solve optimization problems for various types of functions. The butterfly algorithm is proposed to solve a kind of high-dimensional complex function [46] and the particle swarm optimization with improved parameters is constructed to solve large-scale combinatorial optimization problems [47]. In addition, a hybrid metaheuristic algorithm is adopted to solve the multi-objective appliance optimization scheduling problem [48]. To address the randomness inherent in the Extreme Learning Machine (ELM), recent studies have introduced intelligent optimization algorithms to enhance its performance. For example, a Nonlinear dynamic convergence factor-based Chicken Swarm Optimization algorithm (NCSO) [49] and an improved Local escape Dwarf Mongoose Optimization Algorithm (LDMOA) [50] have been employed to fine-tune the parameters of ELM, both of which significantly reduced forecasting errors and improved overall predictive accuracy. Recently, swarm intelligence algorithms have been increasingly integrated with machine learning methods to improve their performance, with the differential evolution algorithm being commonly used due to its simple structure and ease of implementation. The enhanced differential evolution algorithm has been applied to optimize the identification of parameters in solid oxide fuel cells [51]. Xiong et al. applied the combination of the differential evolutionary algorithm and whale optimization algorithm to solve the parameter extraction of solar photovoltaic models [52]. The differential evolution algorithm, in conjunction with support vector regression, has been employed to address the power load forecasting problem [53] and building energy consumption prediction [54].

Given the critical importance and complexity of power systems, the prediction of power load and renewable energy generation has been a focal point of research for many years. There are many applications for power load prediction, such as regional and industry load, load prediction in particular environments, and renewable energy generation prediction. For these issues, the utilization of machine learning methods for prediction is the leading solution. Besides the prediction method based on traditional machine learning, such as support vector machine and regression methods, most studies have

turned to deep neural networks to seek a more accurate prediction model. Compared with the deep neural network, the fully connected neural network, such as the backpropagation neural network, is relatively simple to the point of being unable to produce an acceptable prediction result in such a complicated environment. Many studies have shown that recurrent neural networks have better performance for data with temporal information like power load and renewable energy generation. So, LSTM and other kinds of recurrent neural networks have become the primary approaches for dealing with the power load prediction problem. Based on the analysis of the previous literature, the differential evolutionary algorithm has a faster speed and simple structure, which is more suitable for multi-parameter optimization of deep neural networks. Therefore, the differential evaluation algorithm is adopted to build the self-adaptive tuning mechanism of the hyperparameter for the neural networks. Finally, a self-adaptive LSTM neural network integrating the differential evolutionary algorithm is established to solve the problem of power load and renewable energy generation prediction.

3. Method

This section briefly introduces the basic process of the Differential Evolution (DE) algorithm and two improvement strategies for DE, as well as the procedure of the backpropagation neural network, RNN and LSTM neural network. Meanwhile, the construction process of these neural networks optimized by differential evolutionary algorithms to achieve adaptive tuning of their hyperparameters is presented in detail.

3.1 *Differential evolution algorithm*

Differential Evolution (DE) is a heuristic-based global optimization algorithm derived from the genetic algorithm and contains four operational processes: initialization, variation, crossover and selection. The core operation of DE is described as follows.

Population initialization. Suppose that there are D variables and the population contain N individuals. The i-th individual is shown in equation (1). The value of each variable is located in a limited range of $(x_j^{\min}, x_j^{\max})$, then the assignment is made as equation (2).

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{iD}), \quad i = 1, 2, \ldots, N \tag{1}$$

$$x_{ij} = x_j^{\min} + \text{rand}(0, 1) \times \left(x_j^{\max} - x_j^{\min}\right), \quad i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, D \tag{2}$$

Where rand is a random number with a uniform probability distribution.

Mutation operation. If an individual is variated, three random individuals x_{p1}, x_{p2}, x_{p3} are selected. The mutation operation is then computed according to equation (3).

$$H_i = x_{p1} + F \times (x_{p2} - x_{p3}) \tag{3}$$

Where F is the scaling factor used to adjust the influence of the difference vector.

Crossover operation. The crossover operation is then performed to generate offspring candidates according to equation (4).

$$v_{ij} = \begin{cases} H_{ij}, & \text{rand}(0, 1) < CR \text{ or } j = k \\ x_{ij}, & \text{else} \end{cases} \tag{4}$$

Where the mutant individuals and the target individuals is represented as H_i and x_i, respectively. The offspring candidates is noted as v_i and CR is the crossover probability.

Selection operation. The selection operation is to compare the fitness of the target individuals x_i with that of the generated individuals v_i to select the better individuals for the next generation as shown in equation (5).

$$x_i \text{ (next generation)} = \begin{cases} v_i, & f(v_i) < f(x_i) \\ x_i, & \text{else} \end{cases} \tag{5}$$

After the presence of the main operation of DE, the specific process is illustrated as follows and the flowchart is depicted in Figure 1a.

Step 1: set the initial parameters: the population size NP, the maximum number of iterations maxiter, the variance scaling factor F, and the crossover rate CR.

Step 2: produce the population. This stage yields NP individuals according to equation (2) and calculates the fitness function values of each individual.

Step 3: perform mutation and crossover operations. At this stage, mutation operation is performed according to equation (3) and the crossover operation is processed by equation (4) to obtain the offspring candidate individuals.

Step 4: selection operation. The better individuals are selected to form the population for the next generation according to equation (5).

Step 5: update the global optimal individual. If the termination condition is met, the best individual is viewed as the final result; otherwise, return to Step 3.

3.2 *Differential evolution algorithm with elite variation*

The traditional differential evolution algorithm is a DE/rand/1 mutant approach, in which only one differential vector is generated and used to create mutant individuals with the randomly selected individual. As to the improvement of DE, the mutation method of DE/best/1 is applied, in which the generated mutant individual and the global best individual gained from the previous generation are adopted to produce the mutant individual. This approach is more conducive to the approach to the global optimum. However, this approach will result in a certain similarity of mutant individuals arising from each mutant individual in the population, which is not conducive to the diversification of the population. Therefore, these two kinds of mutation operation are integrated to propose a Differential Evolution algorithm based on Elite Variation (EVDE). Firstly, the weight factor w varies with the number of iterations and is determined according to equation (6).

$$\lambda = (\text{maxiter} - \text{generation})/\text{maxiter} \tag{6}$$

Where maxiter is the maximum number of iterations and t is the current number of iterations.

Then, the mutant individual H_i is produced by combining the differential vector in line with equation (7).

$$H_i = \lambda x_{p1} + (1 - \lambda) x_{gbest} + F \times (x_{p2} - x_{p3}) \tag{7}$$

Where the global optimal individual is denoted as x_{gbest} and the randomly selected individual is noted as x_{p1}.

Therefore, at the initial iteration, the mutant individual depends more on the randomly selected individuals, which is more random and beneficial to enhance the population diversity. As the number of iterations increases, the mutant individual is gradually more dependent on the global optimal individual because of the decrease of the weight factor. This improvement accelerates the process of finding the global optimization, which achieves the diversity of the population in the early stage to improve the global search capacity and gradually reduce the step size in the later stage to ensure the stability of the local search.

Then, the process of the EVDE algorithm is constructed with the specific algorithm flow shown in Figure 1b, which is roughly similar to the construction process of DE in Section 3.1.

3.3 *Differential evolutionary with elite variation and self-attention*

As an extension of the previous improvement strategy, the adaptive adjustment of the two hyperparameters of the mutation scaling factor and the crossover probability is proposed to enhance the original algorithm. A Differential Evolution based on Elite Variation and Self-Attention (SAEVDE) is presented.

The mutation scaling factor determines the degree to which an individual is affected by the difference vector in mutation. The larger the mutation scaling factor, the more the mutated individuals are affected by the difference vector. In the case of the randomly selected individuals close to each other, the fitness function values are similar, making the difference vector small. In order to avoid a tiny difference vector having almost no effect on the mutation operation, the expectation is to make the mutation scaling factor larger to some extent to control the degree of the mutation operation. Therefore, the fitness function values of three individuals are calculated to yield the new mutation scaling factor, as shown in equation (8).

$$F = f_{\min} + (f_{\max} - f_{\min}) \times \frac{f_2 - f_1}{f_3 - f_1} \tag{8}$$

Where the range of mutation scaling factor is in $[f_{\min}, f_{\max}]$. f_1, f_2, and f_3 respectively represent the fitness function values of the individuals whose fitness function values range from the best to the worst.

The crossover probability determines how much information the new individual inherits from the mutant individual during the crossover. The smaller the crossover probability, the smaller the proportion of generated individuals inherited from variant individuals, which makes population diversity smaller and results in an easier convergence to the local optimal solution. The larger the crossover probability, the more significant the proportion of new individuals inherited from mutant individuals, which produces improved population diversity and more remarkable global search ability, leading to a faster convergence rate. However, the influence of an overlarge crossover probability on the population may be more significant than the impact of the mutation operation, leading to a slower convergence. Therefore, the preferred method to set the crossover probability is to be adaptively and regularly updated with the generation of iterations, as shown in equation (9).

$$CR = CR_{\min} + (CR_{\max} - CR_{\min}) \times \sin\left(\pi \times \frac{\text{generation}}{\text{maxiter}}\right) \tag{9}$$

At the early stage, the crossover probability is updated from a low to a high value, which gradually increases the population diversity and makes the global exploration ability more robust. Meanwhile, at the later stage, the crossover probability is refreshed from high to low, gradually decreasing the population diversity and ensuring stable convergence to the global optimal solution. Then, the process of the SAEVDE algorithm is illustrated in Figure 1c. The primary operations are similar to the process of DE in Section 3.1, so these are not repeated here.

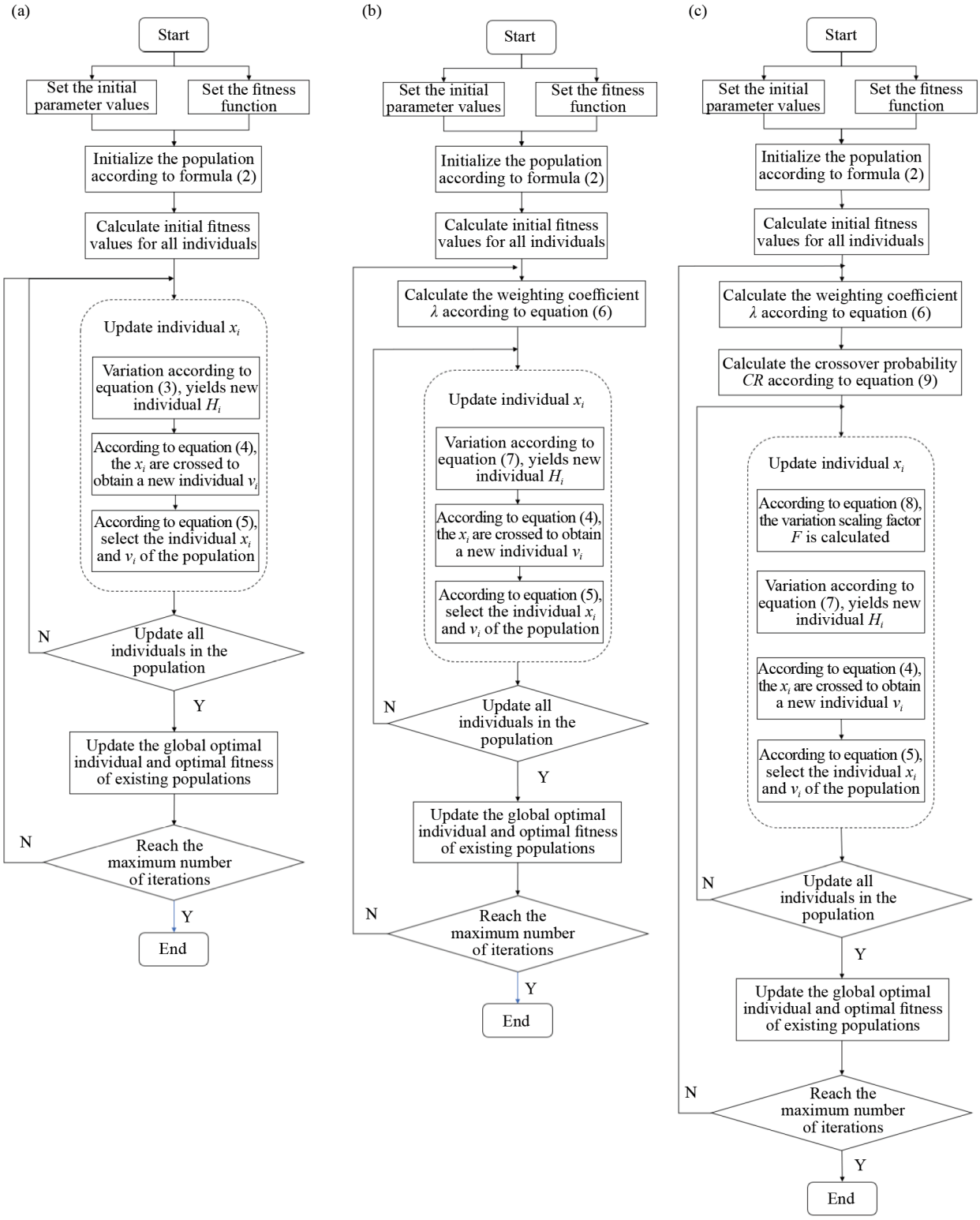


Figure 1. The process of DE (a), EVDE (b), SAEVDE (c)

3.4 *Recurrent neural network*

Tracing back to the 1980s, Recurrent Neural Networks (RNNs) have been pivotal in sequence data processing, with Elman's model standing out as particularly influential [55]. These networks are intricately designed to tackle sequence-dependent tasks prevalent in various real-world scenarios, such as text translation, content generation, and electric load forecasting. Their unique structure enables the integration of current inputs with historical data, thus providing the networks with the profound ability to remember and utilize extensive sequence information.

As illustrated in Figure 2, the RNN enhances a basic feedforward neural network by integrating a hidden state. This hidden state functions to store and convey information from each moment, thus preserving past data. The network's configuration includes x_{t-1} as the previous moment's input, x_t as the current input, and the hidden state H_t, whose dimensions are adjustable and analogous to the neurons in hidden layers of the backpropagation neural network. The activation function is represented by φ, and O_t signifies the output at each timestep. This hidden state, being continually updated, contributes to the sequential computations, embodying the network's temporal depth. RNNs introduce the notion of time steps, signifying the extent of the sequential data being processed, with Figure 2 showcasing 3 RNN cells handling three-time steps of data. The RNN operates through the following principal procedure:

(1) Initiate the process by parameter initialization. This step involves setting up the connection weights from the input layer to the hidden layer, alongside the hidden state's weights and biases, and the output layer's weights and biases, which are specified as W_{xh}, W_{dh}, W_{hq}, b_h, b_q. Furthermore, it's essential to initialize the initial timestep's hidden state in the RNN.

(2) Proceed to compute the hidden state's output. The present hidden state is formulated from the preceding moment's hidden state and the current input, as illustrated in formula (10).

$$H_t = \varphi(X_t W_{xh} + H_{t-1} W_{hh} + b_h) \tag{10}$$

(3) Ascertain the output at this timestep. Drawing on the calculated hidden state, the output for this specific timestep is ascertained following formula (11), ensuring an output at each timestep.

$$O_t = H_t W_{hq} + b_q \tag{11}$$

Based on the output results calculated previously, the error can be computed, allowing for the application of backpropagation to modify the network parameters. Importantly, the Recurrent Neural Network maintains a consistent parameter set across all timesteps, thereby preventing any increase in parameter overhead with extended timesteps. In contrast to traditional backpropagation neural networks, the Recurrent Neural Network uniquely integrates a hidden component to encapsulate and relay historical data, ensuring that the ultimate output reflects all prior moments' information, embodying the Recurrent Neural Network's primary functionality.

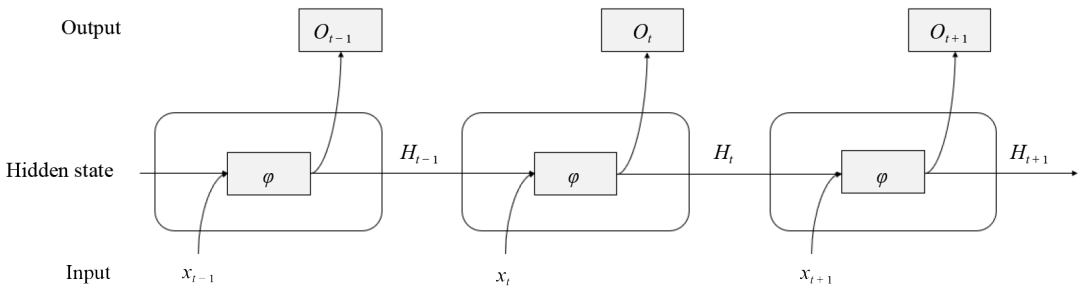


Figure 2. The architecture of RNN

3.5 *Long and short-term memory network*

LSTM, introduced by Hochreiter and Schmidhuber in 1997, revolutionized recurrent neural networks by incorporat-ing a "gate" architecture that addresses the challenge of handling long-term dependencies in time series, a limitation of standard RNNs [21]. This model enhances the RNN with memory cells, paralleling hidden states yet specifically devised for chronicling temporal details. LSTM's design is tailored to elucidate long-term dependencies among the data, incorporating three pivotal gates: forget, input, and output.

Its architecture aligns with RNN, unfolding through a sequence of computational iterations, each encapsulating a distinct computational unit. Distinguished by its complexity, the architecture of each LSTM unit is showcased in Figure 3. The operational dynamics of LSTM is shown as follows:

(1) The forget gate, governed by formula (12), orchestrates the retention or omission of information from the hidden state.

$$F_t = \sigma\left(X_t W_{df} + H_{t-1} W_{hf} + b_f\right) \tag{12}$$

where W_{df}, W_{hf}, b_f denote the forget gate's connection weights and bias, and σ represents the sigmoid function.

(2) The input gate, delineated in formula (13), ascertains the pertinence of the input data.

$$I_t = \sigma\left(X_t W_{di} + H_{t-1} W_{hi} + b_i\right) \tag{13}$$

where W_{di}, W_{hi}, b_i represent the input gate's weights and bias.

(3) Calculation of candidate memory cell. The candidate memory cell contains information from the current input and the previous hidden state. Its computation is specified in the formula (14).

$$\tilde{C}_t = \tanh\left(X_t W_{dc} + H_{t-1} W_{hc} + b_c\right) \tag{14}$$

where W_{dc}, W_{hc}, b_c denote the input connection weights, hidden state connection weights, and biases for this unit.

(4) Calculation of the output memory cell. The memory cell output of this unit is computed using the following formula (15), involving point-wise multiplication of the forget gate with the previous memory cell and the input gate with the candidate memory cell.

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t \tag{15}$$

where $\odot$ represents the point-wise multiplication operation of two vectors.

(5) Proceeding through the output gate. The output gate decides whether to use the hidden state, computed as shown in the formula (16).

$$O_t = \sigma\left(X_t W_{do} + H_{t-1} W_{ho} + b_o\right) \tag{16}$$

where W_{do}, W_{ho}, b_o represent the input connection weights, hidden state connection weights, and biases for the output gate.

(6) Calculation of the hidden state output for this unit. The hidden state output for this time step is computed following formula (17), involving element-wise multiplication of the memory cell after applying the tanh activation function and the result from the output gate.

$$H_t = O_t \odot \tanh(C_t) \tag{17}$$

(7) Calculation of this unit's output. The output of this unit is computed as specified in formula (18).

$$O_t = H_t W_{hq} + b_q \tag{18}$$

It's noteworthy that the memory cell in LSTM serves two main purposes: first, to design mechanisms that balance the use of current moment information and historical data; second, the hidden state values are constrained between -1 and 1, while the memory cell values are not limited, allowing storage of more information and effectively preventing gradient explosions.

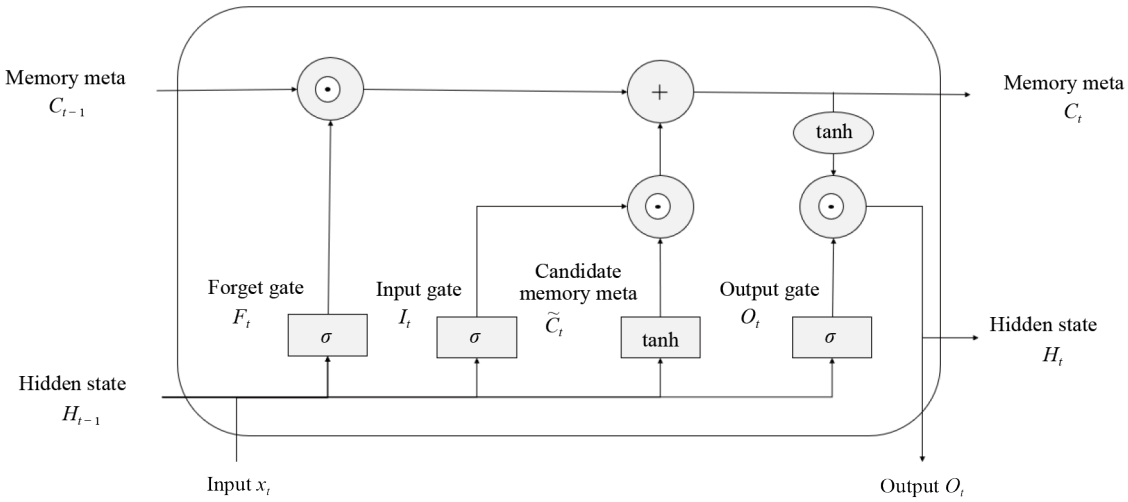


Figure 3. The unit of LSTM

3.6 *Self-adaptive neural network models with DE algorithm*

Considering the complexity inherent in neural network models, setting numerous hyperparameters, such as the number of hidden layer neurons, timestep intervals, and batch sizes, is a critical phase in model development. Inadequate hyperparameter configurations can compromise network performance, and setting these parameters is often a subjective process based on personal experience. This study advances the functionality of neural networks by optimizing their hyperparameters through differential evolution.

Previously, the neural network's algorithmic framework was delineated. This section outlines the procedure post-integration of the algorithm, depicted in Figure 4. The method to optimize the neural network's hyperparameters through DE, EVDE, and SAEVDE involves the following steps:

Step 1: Commence with initializing the differential evolution algorithm. Here, each dimension signifies a neural network hyperparameter, with the dimension's optimization scope representing the potential value range for that

hyperparameter. The algorithm's fitness function is gauged by the predictive error on the test dataset, aiming to minimize this predictive error.

Step 2: The neural network is trained to derive the fitness function value. The neural network is constructed with the hyperparameter set provided by the evolutionary algorithm's individual, calculating the error of the network's output on the test dataset.

Step 3: Ascertain the optimal hyperparameter set. The model is trained accordingly with each iteration of the differential evolution algorithm. Upon hitting the maximum iteration count, the ultimate hyperparameter values are preserved.

Step 4: Formulate the optimized neural network model. Utilize the finest hyperparameter set to construct the neural network model, which is then applied to predict real-world data.

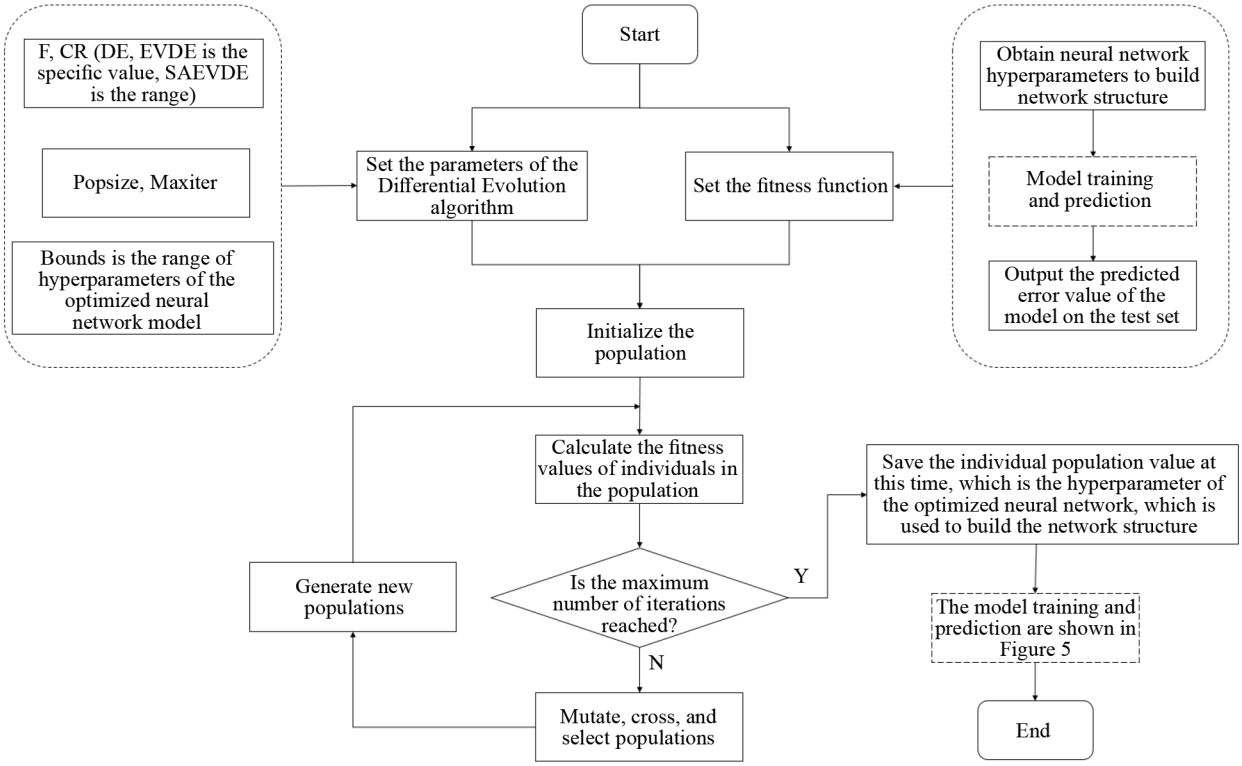


Figure 4. Flowchart of self-adaptive LSTM neural network with DE algorithm

4. Numerical experiments

4.1 *Data presentation*

To validate the performance of the constructed model, the power load data from a region in China is selected for demonstration. The data is a sample of power load data from 1 January 2018, to 31 August 2021, with a period of 15 min. So, we have 128,544 data points after mean-filling the 390 missing values. This section aims to compare the prediction effectiveness of 12 models on such time-series data of power load. Specifically, the data of the last 10 days is used as the test set, while the rest is viewed as the training data. During the training process, 10% of the data from the training set is randomly selected as the validation set to assess the model's accuracy.

Figure 5 presents the data from the most recent ten-day period, elucidating the pronounced periodicity inherent in the power load data under examination. Each day within this decadal dataset embodies a distinct periodicity, exhibiting robust

time-series attributes and epitomizing the characteristics quintessential to power load forecasting. The repetitive pattern observed underscores the data's periodicity, rendering it a paradigmatic example for analyzing and modelling within the context of predictive power load assessments.

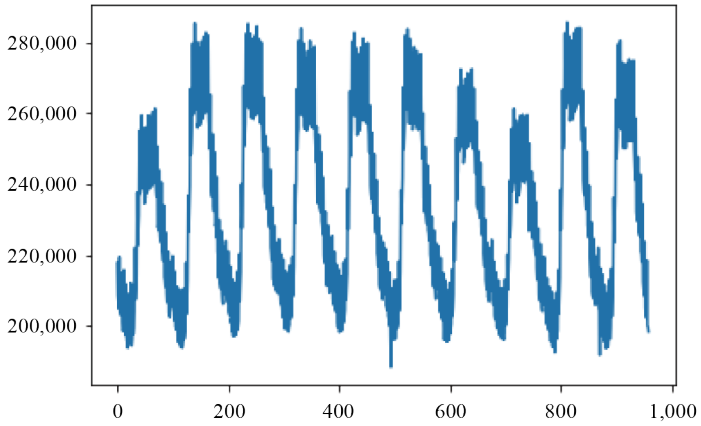


Figure 5. The data of the last 10 days

4.2 *Evaluation indicators*

The models delineated above have yielded predictive outcomes on the test dataset, denoted as $\hat{y} = \{\widehat{y_1}, \widehat{y_2}, \cdots, \widehat{y_n}\}$, alongside the actual values marked as $y = \{y, y_2, \cdots, y_n\}$. Several metrics are available to gauge the efficacy of the model, with four predominant indicators commonly employed for evaluating the performance of a predictive model.

(1) Root Mean Square Error (RMSE)

RMSE represents the square root of the averaged squared discrepancies between the forecasted and the actual values. This metric is quantified on the same scale as the data under consideration, whereby a diminutive RMSE value signifies enhanced accuracy of the model. The computational formula is articulated in Equation (19).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2} \tag{19}$$

(2) Mean Absolute Error (MAE)

MAE denotes the mean of the absolute deviations between the actual and the predicted values. Analogous to RMSE, this metric is scaled to the magnitude of the data, with a higher MAE reflecting greater predictive error. The formula for its calculation is presented in Equation (20).

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y_i} - y_i| \tag{20}$$

(3) Coefficient of Determination

This metric assesses the degree of fit between the model predictions and the actual data, with values ranging from 0 to 1. A value proximal to 1 denotes a model yielding highly accurate predictions. The formula for computation is delineated in Equation (21).

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \widehat{y_i})^2}{\sum_{i=1}^{n} (y_i - y_y)^2} \tag{21}$$

(4) Mean Absolute Percentage Error (MAPE)

MAPE assesses the relative error magnitude between the predicted and the actual data, where a lower MAPE value indicates minimized predictive discrepancies. The calculation methodology is explicated in Equation (22).

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{\widehat{y_i} - y_i}{y_i} \right| \tag{22}$$

4.3 *Parameters for the neural network and differential evolutionary algorithm*

To strengthen the comparative analysis and substantiate the effectiveness of the developed models, a conventional Backpropagation Neural Network (BPNN) is utilized in the experiments for comparison. The fundamental hyperparameters for the BPNN, RNN, and LSTM neural networks are provided in Table 1. These models are implemented using the PyTorch library. For coherence with subsequent RNN and LSTM analyses, the input dimension is equated to the time step in this context. The three hyperparameters of input dimension, the hidden layer size, and batch size were optimized using the DE, EVDE, and SAEVDE algorithms, respectively. The DE algorithm's parameters were configured as follows: a mutation factor (F) of 0.5, a crossover probability (CR) of 0.7, a population size of 10, and a maximum iteration count of 20. To be fair in the comparation of the performance of the three optimization algorithms, EVDE and SAEVDE has the same parameters with the original DE algorithm. In order to fairly compare the performance of the three optimization algorithms, the parameters of EVDE and SAEVDE are the same as those of the original DE algorithm, but the variance factor is set to $[0.1, 0.9]$ and the crossover probability is set to $[0, 1]$ according to the rationale of the SAEVDE algorithm. The optimization criterion was the Mean Absolute Percentage Error (MAPE) of the model's predictions on the test set. This approach ensures that the original neural network models are rigorously optimized for its hyperparameters, enhancing its predictive accuracy and reliability in power load forecasting. The utilization of optimization methods for hyperparameter optimization aims to achieve a more efficient and effective neural network model, potentially surpassing the conventional neural network in terms of forecasting precision and generalization capability.

Table 1. Hyperparameters for the neural network models

Hyperparameters	BPNN/RNN/LSTM
The input dimension	16
The batch size	64
The hidden layer size	100
The maximum iteration	200

4.4 *Results and discussion*

Utilizing the DE, EVDE, and SAEVDE algorithms for the hyperparameter optimization of the neural network, refined hyperparameters are employed to construct various neural network models. Subsequently, the predictive efficacy of each model is meticulously compared and analyzed, with the findings presented in Table 2. Meanwhile, each model was evaluated through twenty independent trials, and the Mean Absolute Percentage Errors (MAPE) obtained across these runs were aggregated to construct boxplots for comparative analysis, as illustrated in Figure 6.

Table 2. The performance of all predictive models

Model	RMSE	MAE	R^2	MAPE
BPNN	13,439.17	11,278.27	0.7130	0.0511
DE-BPNN	11,024.35	11,024.35	0.7200	0.0511
EVDE-BPNN	12,851.63	10,496.47	0.7379	0.0483
SAEVDE-BPNN	9498.44	8788.19	0.8503	0.0346
RNN	11,735.12	10,175.93	0.7812	0.0462
DE-RNN	11,730.30	9413.19	0.7818	0.0401
EVDE-RNN	10,047.28	7882.30	0.8385	0.0367
SAEVDE-RNN	6785.49	5790.80	0.9270	0.0248
LSTM	11,735.12	10,175.93	0.7812	0.0462
DE-LSTM	10,972.07	9065.18	0.8093	0.0415
EVDE-LSTM	9346.75	7742.60	0.8611	0.0350
SAEVDE-LSTM	6695.49	5539.28	0.9290	0.0137

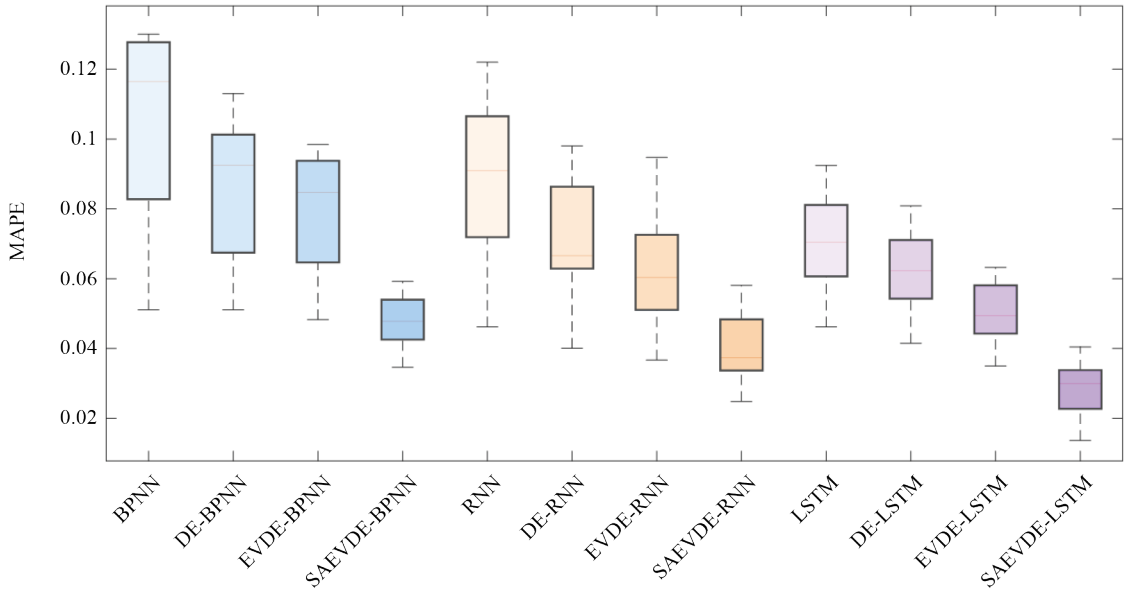


Figure 6. Stability of the predictive models

A thorough comparative evaluation of the predictive performance across all models reveals a discernible hierarchy in effectiveness; notably, the LSTM model outperforms the RNN and BPNN in predictive accuracy. Moreover, the application of the Differential Evolution (DE) algorithm for hyperparameter optimization facilitates the identification of an optimal hyperparameter configuration, thereby enhancing the resultant model's performance. Among the optimization algorithms, the SAEVDE algorithm demonstrates a more pronounced refinement capability than the DE and EVDE algorithms. It is particularly evident from the evaluation of the four key metrics that the SAEVDE-integrated neural network models exhibit more minor prediction errors and higher accuracy than their counterparts combined with other algorithms. A comprehensive analysis of the evaluation metrics unequivocally suggests that the SAEVDE-LSTM model is the superior model among all considered. Hence, this model is designated for subsequent predictive applications in empirical analysis,

given its exemplary performance and accuracy in forecasting. As illustrated in Figure 6, neural networks optimized using DE-based algorithms exhibit higher stability in terms of MAPE across multiple experimental runs. Among these models, the SAEVDE-LSTM demonstrates the most consistent performance, indicating that the SAEVDE algorithm achieves more effective hyperparameter optimization than the two baseline DE approaches. These findings further confirm the robustness of the proposed optimization strategy and highlight the superior forecasting capability of the LSTM network compared with the RNN and BPNN models for this type of predictive task.

4.5 *Comparative verification*

To further substantiate the performance of the algorithms employed in this study, additional comparative experiments were conducted. These experiments aimed to evaluate the optimization capability of the proposed SAEVDE algorithm and to assess the predictive performance of the SAEVDE-LSTM model in greater depth.

(1) Comparison with Other Optimization Algorithms

To verify the effectiveness of the SAEVDE algorithm in optimizing the hyperparameters of deep learning models, several widely recognized optimization algorithms were selected for comparison. These included Bayesian Optimization, Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Kepler Optimization Algorithm (KOA), and the Runge-Kutta Optimization Algorithm (RUN). During the hyperparameter tuning of the LSTM network, all algorithms were configured with consistent parameter settings wherever possible, while maintaining the same population size and maximum iteration count as those used in the SAEVDE algorithm to ensure a fair comparison.

The predictive outcomes of the LSTM model optimized by each algorithm are summarized in Table 3. As indicated, the SAEVDE algorithm achieved markedly superior optimization performance compared with the other heuristic approaches, demonstrating its enhanced capability in fine-tuning deep learning models. These results further validate the robustness and effectiveness of the proposed SAEVDE method for addressing complex optimization problems in predictive modeling.

Table 3. The performance of all optimization algorithms

Model	RMSE	MAE	R^2	MAPE
Bayes-LSTM	9065.23	8501.36	0.8901	0.4901
PSO-LSTM	7906.49	6987.12	0.9012	0.0449
GWO-LSTM	8751.47	7852.49	0.9001	0.0523
KOA-LSTM	9825.49	8923.55	0.8859	0.5121
RUN-LSTM	8743.29	7980.14	0.9123	0.0499
SAEVDE-LSTM	6695.49	5539.28	0.9290	0.0137

(2) Comparison with Other Models

In order to evaluate the predictive performance of the proposed model, we conducted a comparative analysis with several well-established deep learning models commonly used for short-term electricity demand forecasting. These models include the Gated Recurrent Unit (GRU), Bidirectional Long Short-Term Memory (BiLSTM), Convolutional Neural Network (CNN), Temporal Convolutional Network (TCN), and Transformer. The results of these models are presented in Table 4.

The results in Table 4 clearly demonstrate that the optimized LSTM model outperforms the baseline models in terms of prediction accuracy. This underscores the significant enhancement in model performance achieved through hyperparameter optimization using SAEVDE. Of particular note, the GRU and BiLSTM models exhibited relatively suboptimal performance, likely due to the inherent characteristics of the time-series data, which may not align well with the strengths of these models. The CNN model, which is primarily designed for image-related tasks, would require additional preprocessing of the time-series data to adapt it for forecasting purposes. In contrast, the Transformer model performed

below expectations, reinforcing conclusions found in existing literature that its complexity necessitates a more meticulous design of its architecture to adequately address forecasting challenges.

Table 4. The results of the comparative experiments

Model	RMSE	MAE	R^2	MAPE
GRU	11,409.01	9056.23	0.7823	0.1025
BiLSTM	9886.34	8765.89	0.8956	0.0823
CNN	12,027.69	12,036.32	0.7589	0.1122
TCN	10,075.25	9875.56	0.8156	0.0836
Transformer	16,570.00	13,500.15	0.6954	0.1385
SAEVDE-LSTM	6695.49	5539.28	0.9290	0.0137

5. Case study on the power system in Danish

As technological advancements continue, electricity generation methods are increasingly diversified, incorporating wind, hydro, thermal, and solar power, which are now widely utilized. This diversification has substantially addressed national electricity demands and enhanced energy utilization efficiency. However, the associated challenge of resource allocation has emerged, prompting a need to optimize renewable energy use by tailoring it to the specific characteristics of each electricity generation method, aiming to balance electricity supply and demand, and striving for carbon neutrality within the power system. Denmark, renowned for its highly developed renewable energy sector, especially wind power, which represents the largest share, underscores the importance of precise predictions of electricity demand and wind power generation. Such forecasts are vital for power companies to devise rational power generation plans. Hence, this section will predict Denmark's electricity demand and wind power generation to evaluate power supply augmentation strategies.

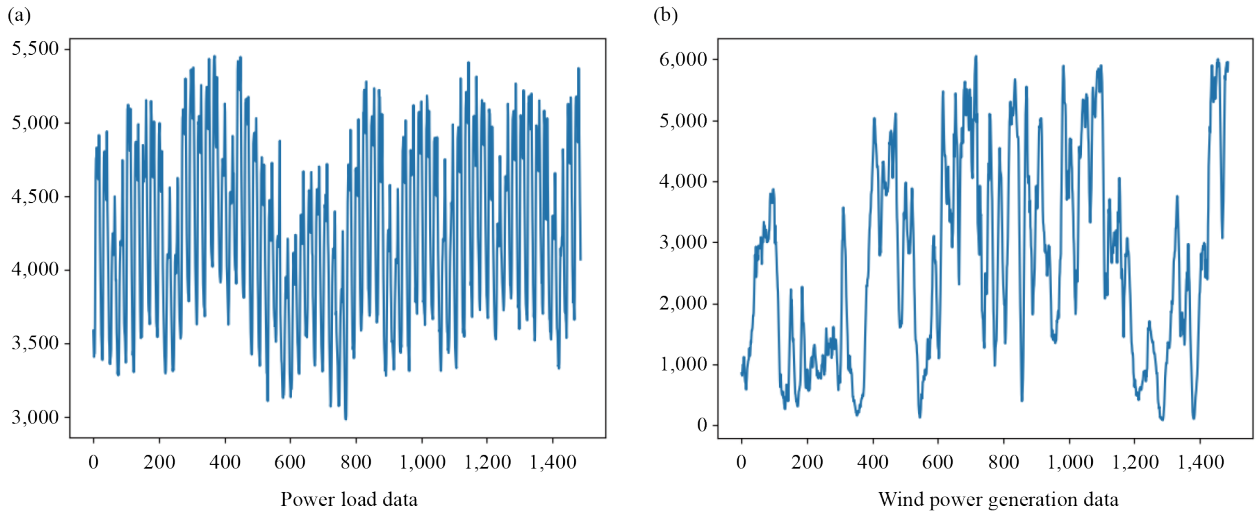


Figure 7. Description of the empirical data set

Data encompassing Denmark's power load and wind power generation from 1 January 2021, to 31 January 2023, were gathered, including hourly total power loads and wind power generations (both onshore and offshore) across two regions. This data was then amalgamated to construct the total wind power production. Subsequently, the aggregation of

these regional figures provided a complete profile of total electricity demand and overall wind generation. The refined dataset contains 18,260 data points, each indicating the power load and the aggregate wind power generation. For practical model training and forecasting, the data from the final two months, or the latest 1488 data entries, were designated as the test set, with the preceding data serving as the training set. The power load and total wind generation are depicted in Figures 7a and 7b, exhibiting slightly periodic trends with observable fluctuations.

5.1 *Prediction of power load*

In the forecast of power load, the SAEVDE algorithm was initially utilized to optimize three parameters of the LSTM, aligning with the methodologies detailed in Section 4. The tuned hyperparameters were identified as a batch size of 31, a timestep of 38, and 81 neurons in the hidden layer. The LSTM model was constructed using these parameters. The training progression is depicted in Figure 8a, while the predictive results on the test dataset are illustrated in Figure 8b. The model's evaluation indicators, presented in Table 5, demonstrate that the self-adaptive parameters model has a high prediction precision and can effectively address the demand fluctuations within the electricity system and fulfil the Danish power grid's forecasting requisites.

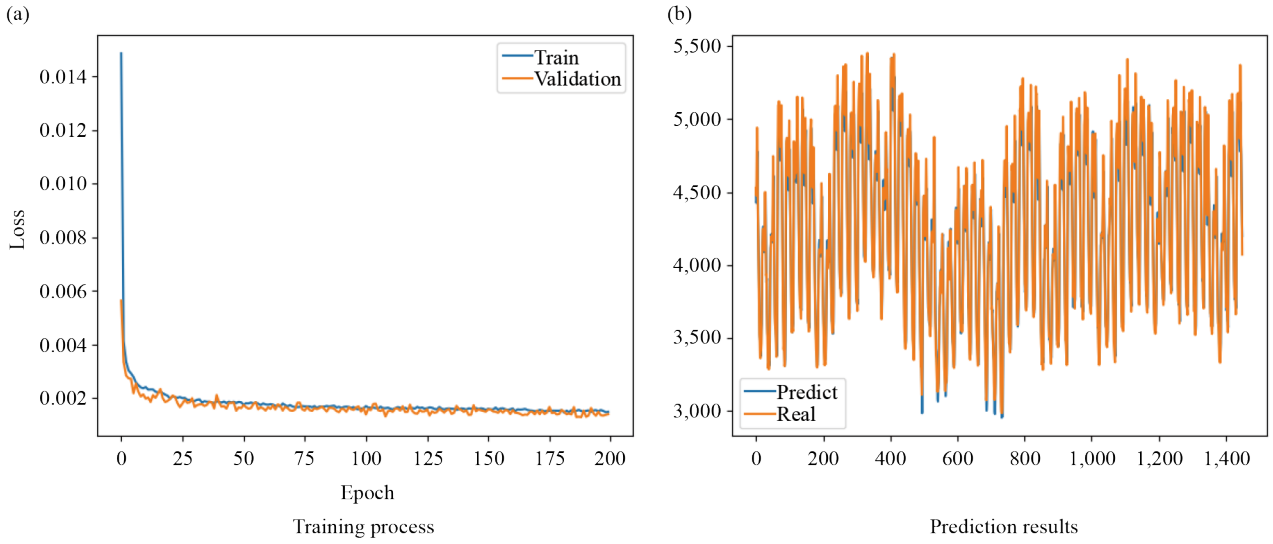


Figure 8. Training and prediction results of LSTM model

Table 5. The effect of power load prediction

RMSE	MAE	R^2	MAPE
118.0913	90.0927	0.9590	0.0209

5.2 *Prediction of wind power generation*

Similarly, for wind power generation forecasting, the neural network's hyperparameters were optimized to a batch size of 157, a timestep length of 24, and a hidden layer size of 178. The model's training process and its forecasting accuracy are respectively depicted in Figures 9a and 9b. As delineated in Table 6, the model's performance indicators reveal that the forecast accuracy approaches 99%, demonstrating the model's capability to predict future wind power generation accurately. This level of precision is crucial for enabling power generation enterprises to devise strategic plans for baseline electricity production, thereby facilitating the green transition of the energy system.

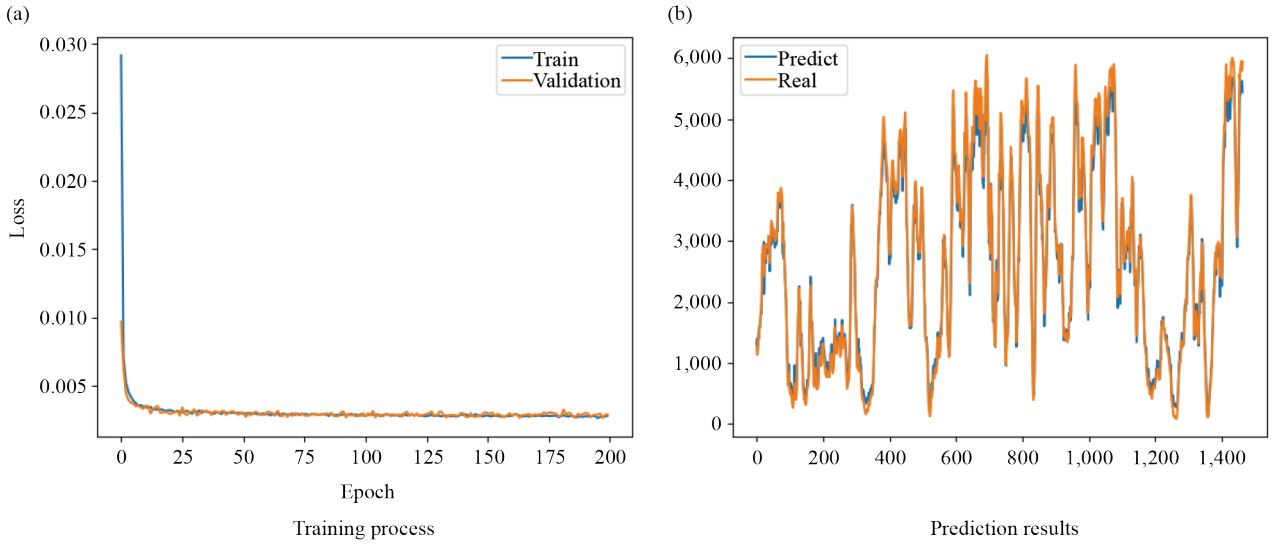


Figure 9. Training and prediction results of LSTM model

Table 6. Evaluation of the effect of wind power generation prediction

RMSE	MAE	R^2	MAPE
222.4649	170.9821	0.9814	0.114

The power load reflects the cumulative electricity requirements of the population, with the shortfall between this and wind power generation necessitating compensation from other energy sources, even in Denmark, where green energy is extensively developed. This pattern is characterized by its regularity and periodicity, primarily due to the seasonal periodicity of wind power data and the more frequent daily periodicity in power load, showing periodic gaps. Utilizing the model introduced in this document, predictions can be made for both electricity load and wind power generation, leading to a strategic plan for supplemental electricity provision. Essentially, this enables the preemptive forecasting of electricity demands and renewable generation capabilities, facilitating the formulation of baseload energy production strategies and the subsequent adjustment and planning of a diverse electricity generation infrastructure.

6. Conclusions and future research directions

6.1 *Conclusions*

Centering on power load forecasting and tackling the equilibrium challenge of green power supply and demand, an optimization algorithm is presented to enhance the neural network model, which facilitates adaptive parameter tuning to establish an improved predictive model. Through comparative analysis of several enhanced models, the most effective one was chosen for forecasting electricity load and generation data, thereby aiding in addressing complex power supply-demand arrangement challenges. Initially, the paper elaborated on the principles behind the Differential Evolution algorithm and its enhancements, leading to the development of DE, EVDE, and SAEVDE algorithms. It also outlined the construction of BPNN, RNN, and LSTM neural networks. The optimization algorithms DE, EVDE, and SAEVDE were then applied to tune the hyperparameters of these neural networks, creating models capable of adaptive parameter adjustment. These models were tasked with forecasting electricity load data, and their performance was evaluated using the actual power load data. The findings demonstrated that the SAEVDE-LSTM model outshined others across all four evaluative metrics: RMSE, MAE, R^2, and RMSE. Subsequently, this optimized model was utilized for forecasting Denmark's power load and

wind power generation, allowing for a detailed comparison and analysis of the predicted results, which in turn facilitated the identification of a gap with periodic fluctuations between the total electricity demand and wind power generation, thereby aiding the formulation of supplemental electricity supply plans and contributing to the strategic adjustment and design of a multifaceted power system.

6.2 *Future research directions*

This study proposed a hybrid prediction model that integrates the SAEVDE algorithm with LSTM neural networks to optimize hyperparameters and verified its effectiveness in forecasting electricity load and renewable energy generation. Future research can further broaden the applicability and theoretical scope of this framework along several promising directions: Subsequent studies may integrate the proposed high-accuracy forecasting model with methodologies derived from adaptive control theory. By leveraging precise load and renewable generation forecasts to reduce uncertainties in energy management systems, such integration could enhance overall system stability, improve renewable energy utilization efficiency, and contribute to the secure and reliable operation of modern power systems. This synergy would bridge predictive modeling with real-time control, advancing the development of adaptive and resilient smart grids. Furthermore, future work will also aim to extend the model's application to a broader range of renewable energy resources by collecting datasets across different regions and energy types (e.g., solar, hydro, geothermal). By tailoring model architectures and optimization strategies to the unique generation characteristics of each energy source, these studies can improve forecasting precision and robustness. Such efforts will ultimately facilitate higher renewable energy penetration and more efficient integration of diverse resources into power systems. Moreover, another avenue for future research involves the development of regionalized forecasting models. Incorporating real-time power load data from various regions and conducting feature analyses based on local socioeconomic characteristics and consumption behaviors can yield region-specific predictive frameworks. These localized models are expected to deliver more accurate and reliable forecasts, thereby providing enhanced decision support for dynamic energy scheduling, demand-side management, and operational planning in smart grid environments. Overall, future research will focus on improving the generalizability, interpretability, and real-world adaptability of the proposed framework. Emphasizing both methodological innovation and domain-oriented application, these efforts are expected to contribute to the advancement of intelligent energy forecasting systems and promote sustainable, data-driven development within modern power grids.

Funding

This research was funded by the Fundamental Research Funds for the Central Universities, South-Central MinZu University (Grant Number: CSQ25011).

Data availability statement

Other researchers can gain the original data from the corresponding author upon a reasonable request.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Jahan IS, Snasel V, Misak S. Intelligent systems for power load forecasting: a study review. *Energies*. 2020; 13(22): 6105. Available from: https://doi.org/10.3390/en13226105.

[2] Yang A, Li W, Yang X. Short-term electricity load forecasting based on feature selection and least squares support vector machines. *Knowledge-Based Systems*. 2019; 163: 159–173. Available from: https://doi.org/10.1016/j.knosys.2018.08.027.

[3] Abdi H, Beigvand SD. Long-term load forecasting based on gravitational search algorithm. *Journal of Intelligent and Fuzzy Systems*. 2016; 30: 3633–3643. Available from: https://doi.org/10.3233/IFS-162108.

[4] Kalhori MRN, Emami IT, Fallahi F, Tabarzadi M. A data-driven knowledge-based system with reasoning under uncertain evidence for regional long-term hourly load forecasting. *Applied Energy*. 2022; 314: 118975. Available from: https://doi.org/10.1016/j.apenergy.2022.118975.

[5] Aisyah S, Simaremare AA, Adytia D, Aditya IA, Alamsyah A. Exploratory weather data analysis for electricity load forecasting using SVM and GRNN, case study in Bali, Indonesia. *Energies*. 2022; 15(10): 3566. Available from: https://doi.org/10.3390/en15103566.

[6] Singh RP, Gao PX, Lizotte DJ. On hourly home peak load prediction. In: *Proceedings of the 2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*. Tainan, Taiwan: IEEE; 2012. p.163–168. Available from: https://doi.org/10.1109/SmartGridComm.2012.6485977.

[7] Fan C, Xiao F, Wang S. Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques. *Applied Energy*. 2014; 127: 1–10. Available from: https://doi.org/10.1016/j.apenergy.2014.04.016.

[8] Mahmud K, Ravishankar J, Hossain MJ, Dong ZY. The impact of prediction errors in the domestic peak power demand management. *IEEE Transactions on Industrial Informatics*. 2020; 16(7): 4567–4579. Available from: https://doi.org/10.1109/TII.2019.2946292.

[9] Wu Q, Ren H, Shi S, Fang C, Wan S, Li Q. Analysis and prediction of industrial energy consumption behavior based on big data and artificial intelligence. *Energy Reports*. 2023; 9: 395–402. Available from: https://doi.org/10.1016/j.egyr.2023.01.007.

[10] Liu J, Zhang Z, Fan X, Zhang Y, Wang J, Zhou K, et al. Power system load forecasting using mobility optimization and multi-task learning in COVID-19. *Applied Energy*. 2022; 310: 118303. Available from: https://doi.org/10.1016/j.apenergy.2021.118303.

[11] Arias MB, Kim M, Bae S. Prediction of electric vehicle charging-power demand in realistic urban traffic networks. *Applied Energy*. 2017; 195: 738–753. Available from: https://doi.org/10.1016/j.apenergy.2017.02.021.

[12] Moon H, Park SY, Jeong C, Lee J. Forecasting electricity demand of electric vehicles by analyzing consumers' charging patterns. *Transportation Research Part D: Transport and Environment*. 2018; 62: 64–79. Available from: https://doi.org/10.1016/j.trd.2018.02.009.

[13] Cesaro Z, Ives M, Nayak-Luke R, Mason M, Bañares-Alcántara R. Ammonia to power: Forecasting the levelized cost of electricity from green ammonia in large-scale power plants. *Applied Energy*. 2021; 282: 116009. Available from: https://doi.org/10.1016/j.apenergy.2020.116009.

[14] Zhou P, Jin RY, Fan LW. Reliability and economic evaluation of power system with renewables: A review. *Renewable and Sustainable Energy Reviews*. 2016; 58: 537–547. Available from: https://doi.org/10.1016/j.rser.2015.12.344.

[15] Cordelli DM, Garavelli L, Savasta S, Guerra A, Pellicciari A, Giordano L, et al. Epilepsy in Mowat-Wilson syndrome: Delineation of the electroclinical phenotype. *American Journal of Medical Genetics Part A*. 2013; 161(2): 273–284. Available from: https://doi.org/10.1002/ajmg.a.35717.

[16] Serrano-Guerrero X, Briceño-León M, Clairand JM, Escrivá-Escrivá G. A new interval prediction methodology for short-term electric load forecasting based on pattern recognition. *Applied Energy*. 2021; 297: 117173. Available from: https://doi.org/10.1016/j.apenergy.2021.117173.

[17] Bracale A, Caramia P, De Falco P, Hong T. Multivariate quantile regression for short-term probabilistic load forecasting. *IEEE Transactions on Power Systems*. 2020; 35(1): 628–638. Available from: https://doi.org/10.1109/TPWRS.2019.2924224.

[18] Liu J, Li C. The short-term power load forecasting based on sperm whale algorithm and wavelet least square support vector machine with DWT-IR for feature selection. *Sustainability*. 2017; 9(7): 1188. Available from: https://doi.org/10.3390/su9071188.

[19] Gao J, Wang K, Kang X, Li H, Chen S. Ultra-short-term electricity load forecasting based on improved random forest algorithm. *AIP Advances*. 2023; 13: 065208. Available from: https://doi.org/10.1063/5.0153550.

[20] Chen J, Zhang D, Nanehkaran Y. Research of power load prediction based on boost clustering. *Soft Computing*. 2021; 25: 6401–6413. Available from: https://doi.org/10.1007/s00500-021-05632-5.

[21] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*. 1997; 9(8): 1735–1780. Available from: https://doi.org/10.1162/neco.1997.9.8.1735.
[22] Kolokythas KV, Argiriou AA. Optimizing the input vectors of applied artificial neural network models for wind power production forecasting. *Wind Engineering*. 2022; 46: 712–723. Available from: https://doi.org/10.1177/0309524X211046382.
[23] Mo J, Wang R, Cao M, Yang K, Yang X, Zhang T. A hybrid temporal convolutional network and prophet model for power load forecasting. *Complex and Intelligent Systems*. 2023; 9: 4249–4261. Available from: https://doi.org/10.1007/s40747-022-00952-x.
[24] Zuo C, Wang J, Liu M, Deng S, Wang Q. An ensemble framework for short-term load forecasting based on TimesNet and TCN. *Energies*. 2023; 16: 5330. Available from: https://doi.org/10.3390/en16145330.
[25] Shen Y, Ma Y, Deng S, Huang C, Kuo P. An ensemble model based on deep learning and data preprocessing for short-term electrical load forecasting. *Sustainability*. 2021; 13(4): 1694. Available from: https://doi.org/10.3390/su13041694.
[26] Geng G, He Y, Zhang J, Qin T, Yang B. Short-term power load forecasting based on PSO-optimized VMD-TCN-attention mechanism. *Energies*. 2023; 16(12): 4616. Available from: https://doi.org/10.3390/en16124616.
[27] Liu M, Li Y, Hu J, Wu X, Deng S, Li H. A new hybrid model based on SCINet and LSTM for short-term power load forecasting. *Energies*. 2024; 17(1): 95. Available from: https://doi.org/10.3390/en17010095.
[28] Ageng D, Huang C, Cheng R. A short-term household load forecasting framework using LSTM and data preparation. *IEEE Access*. 2021; 9: 167911–167919. Available from: https://doi.org/10.1109/ACCESS.2021.3133702.
[29] Wang Y, Zhang N, Chen X. A short-term residential load forecasting model based on LSTM recurrent neural network considering weather features. *Energies*. 2021; 14(10): 2737. Available from: https://doi.org/10.3390/en14102737.
[30] Son N. Comparison of the deep learning performance for short-term power load forecasting. *Sustainability*. 2021; 13(22): 12493. Available from: https://doi.org/10.3390/su132212493.
[31] Qi Y, Luo H, Luo Y, Liao R, Ye L. Adaptive clustering long short-term memory network for short-term power load forecasting. *Energies*. 2023; 16(7): 6230. Available from: https://doi.org/10.3390/en16176230.
[32] Alonso A, Nogales F, Ruiz C. A single scalable LSTM model for short-term forecasting of massive electricity time series. *Energies*. 2020; 13(20): 5328. Available from: https://doi.org/10.3390/en13205328.
[33] Zhang S, Chen R, Cao J, Tan J. A CNN and LSTM-based multi-task learning architecture for short and medium-term electricity load forecasting. *Electric Power Systems Research*. 2023; 222: 109507. Available from: https://doi.org/10.1016/j.epsr.2023.109507.
[34] Saeed F, Paul A, Seo H. A hybrid channel-communication-enabled CNN-LSTM model for electricity load forecasting. *Energies*. 2022; 15(6): 2263. Available from: https://doi.org/10.3390/en15062263.
[35] Rubasinghe O, Zhang X, Chau T, Chow Y, Fernando T, Iu H. A novel sequence to sequence data modelling based CNN-LSTM algorithm for three years ahead monthly peak load forecasting. *IEEE Transactions on Power Systems*. 2024; 39(1): 1932–1947. Available from: https://doi.org/10.1109/TPWRS.2023.3271325.
[36] Wan A, Chang Q, AL-Bukhaiti K, He J. Short-term power load forecasting for combined heat and power using CNN-LSTM enhanced by attention mechanism. *Energy*. 2023; 282: 128274. Available from: https://doi.org/10.1016/j.energy.2023.128274.
[37] Qu J, Qian Z, Pei Y. Day-ahead hourly photovoltaic power forecasting using attention-based CNN-LSTM neural network embedded with multiple relevant and target variables prediction pattern. *Energy*. 2021; 232: 120996. Available from: https://doi.org/10.1016/j.energy.2021.120996.
[38] Shakir M, Biletskiy Y. Forecasting and optimisation for microgrid in home energy management systems. *IET Generation, Transmission and Distribution*. 2020; 14(17): 3458–3468. Available from: https://doi.org/10.1049/iet-gtd.2019.1285.
[39] Yu Y, Zhang M. Control chart recognition based on the parallel model of CNN and LSTM with GA optimization. *Expert Systems with Applications*. 2021; 185: 115689. Available from: https://doi.org/10.1016/j.eswa.2021.115689.
[40] Banik R, Das P, Ray S, Biswas A. Prediction of electrical energy consumption based on machine learning technique. *Electrical Engineering*. 2021; 103: 909–920. Available from: https://doi.org/10.1007/s00202-020-01126-z.
[41] Khan PW, Byun YC, Lee SJ, Kang DH, Kang JY, Park HS. Machine learning-based approach to predict energy consumption of renewable and nonrenewable power sources. *Energies*. 2020; 13(18): 4870. Available from: https://doi.org/10.3390/en13184870.

[42] Lukong TK, Nganyu Tanyu D, Nkongtchou Y, Tatietse TT, Schulz D. A spatial long-term load forecast using a multiple delineated machine learning approach. *Energies*. 2025; 18(10): 2484. Available from: https://doi.org/10.3390/en18102484.

[43] Ge Q, Guo C, Jiang H, Lu Z, Yao G, Zhang J, et al. Industrial power load forecasting method based on reinforcement learning and PSO-LSSVM. *IEEE Transactions on Cybernetics*. 2022; 52(2): 1112–1124. Available from: https://doi.org/10.1109/TCYB.2020.2983871.

[44] Song Y, Park J, Suh MS, Kim C. Prediction of full-load electrical power output of combined cycle power plant using a super learner ensemble. *Applied Sciences*. 2024; 14(24): 11638. Available from: https://doi.org/10.3390/app142411638.

[45] Goswami K, Kandali AB. Machine learning algorithms for predicting electrical load demand: an evaluation and comparison. *Sādhanā*. 2024; 49: 40. Available from: https://doi.org/10.1007/s12046-023-02354-2.

[46] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing*. 2019; 23: 715–734. Available from: https://doi.org/10.1007/s00500-018-3102-4.

[47] Wang D, Tan D, Liu L. Particle swarm optimization algorithm: an overview. *Soft Computing*. 2018; 22: 387–408. Available from: https://doi.org/10.1007/s00500-016-2474-6.

[48] Liu Y, Li H, Zhu J, Lin Y, Lei W. Multi-objective optimal scheduling of household appliances for demand side management using a hybrid heuristic algorithm. *Energy*. 2023; 262: 125460. Available from: https://doi.org/10.1016/j.energy.2022.125460.

[49] Chen XD, Yang HY, Wun JS, Wu CH, Wang CH, Li LL. Power load forecasting in energy system based on improved extreme learning machine. *Energy Exploration and Exploitation*. 2020; 38: 1194–1211. Available from: https://doi.org/10.1177/0144598720903797.

[50] Wang H, Zhang Y, Mu L. Short-term electrical load forecasting using an enhanced extreme learning machine based on the improved dwarf mongoose optimization algorithm. *Symmetry*. 2024; 16(5): 628. Available from: https://doi.org/10.3390/sym16050628.

[51] Xiong G, Zhang J, Shi D, Zhu L, Yuan X. Optimal identification of solid oxide fuel cell parameters using a competitive hybrid differential evolution and Jaya algorithm. *International Journal of Hydrogen Energy*. 2021; 46(9): 6720–6733. Available from: https://doi.org/10.1016/j.ijhydene.2020.11.119.

[52] Xiong G, Zhang J, Yuan X, Shi D, He Y, Yao G. Parameter extraction of solar photovoltaic models by means of a hybrid differential evolution with whale optimization algorithm. *Solar Energy*. 2018; 176: 742-761. Available from: https://doi.org/10.1016/j.solener.2018.10.050.

[53] Sun W, Liang Y. Research of least squares support vector regression based on differential evolution algorithm in short-term load forecasting model. *Journal of Renewable and Sustainable Energy*. 2014; 6(5): 053137. Available from: https://doi.org/10.1063/1.4900552.

[54] Zhang F, Deb C, Lee SE, Yang J, Shah KW. Time series forecasting for building energy consumption using weighted support vector regression with differential evolution optimization technique. *Energy and Buildings*. 2016; 126: 94–103. Available from: https://doi.org/10.1016/j.enbuild.2016.05.028.

[55] Elman JL. Finding structure in time. *Cognitive Science*. 1990; 14(2): 179–211. Available from: https://doi.org/10.1016/0364-0213(90)90002-E.