A Combined Model Based on ISSA-CNN-BiGRU-MH-Attention and Its Application in Power Load Forecasting

Yixiang Ding

School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China, 212013

3220613043.ujs@vip.163.com

Abstract. Existing prediction algorithms face challenges in terms of accuracy and training speed, which hinders high-efficiency, high-accuracy power load forecasting. This paper proposes a combined prediction model to address these issues. To overcome the Bidirectional Gated Recurrent Unit (BiGRU)'s limitations in capturing long-term dependencies and handling complex time-series data, a Convolutional Neural Network (CNN) module is introduced to extract local features and enhance the model's feature representation. Additionally, a Multi-Head Attention (MH-Attention) module is incorporated to dynamically assign weights to different time steps, improving adaptivity and focus on key features. For hyperparameter optimization, an Improved Sparrow Search Algorithm (ISSA) is proposed, which addresses traditional SSA's tendency to fall into local optima and slow convergence by incorporating an adaptive update mechanism and hybrid heuristic strategy. The model is validated using a power plant dataset from Quanzhou, with results showing excellent forecasting ability: R2=0.9955, RMSE=56.9596, and MAE=34.6080. Comparison with other models demonstrates improved performance, with R2 increasing by 0.3%-0.65%, RMSE decreasing by 6.38%-35.96%, and MAE reducing by 26.54%-49.52%. These results confirm the model's effectiveness and superiority.

Keywords: Electricity Load Short-term Forecasting, BiGRU, MH-Attention, Sparrow Search Algorithm.

1. Introduction

Accurate load forecasting is crucial for power system stability and security [1]. As demand grows, advanced forecasting algorithms are needed to optimize generation, dispatch, and resource use [2]. However, existing algorithms often face challenges in balancing prediction accuracy and computational efficiency, especially in short-term load forecasting scenarios.

Current power load forecasting algorithms primarily encompass statistical methods [3], machine learning methods [4] and deep learning methods [5]. In the realm of traditional statistics, Mohammad et al. [3] employed regression analysis for short-term load forecasting, a method that yielded superior forecasting outcomes. However, this approach is hindered by its inability to effectively handle nonlinear relationships and intricate patterns. In contrast, Sanjeev et al. [4] utilised an AutoRegressive Integrated Moving Average (ARIMA) model to forecast power loads, a model that could capture certain trends. Nevertheless, its capacity to handle sudden changes is deemed inadequate. Machine learning methods have been shown to have significant advantages over traditional statistical methods in terms of improving forecasting accuracy and handling more complex nonlinear data. Siti et al. [4] employed a support vector machine (SVM) for load forecasting, achieving more accurate results. However, the model's robustness is compromised in noisy data. Shi et al. [6] utilised random forest (RF) for short-term load forecasting, though the forecasting accuracy was not as good as that of the forecasting itself. The accuracy of load forecasting was enhanced, but the training time was lengthier, which impacted the practical application. Deep learning methods have been shown to be capable of handling complex high-dimensional data with strong expressive ability and self-learning ability in power load forecasting. Hua et al. [7] achieved high prediction accuracy by feature extraction of load data through a Convolutional Neural Network (CNN) model, but the method has difficulty in modelling long time dependencies. Hardanee et al. [8] used an Recurrent Neural Network (RNN) for load forecasting, achieving some results, but in the long time series, it is difficult to predict loads.

While certain effects were observed, the challenge of gradient disappearance in long-time series prediction was identified. Yang et al. [9] employed an Long Short-Term Memory (LSTM) model for load prediction, achieving enhanced prediction accuracy. However, the adaptability to diverse load types was found to be inadequate. Hua et al. [7] implemented a Gated Recurrent Unit (GRU) model for short-term load prediction, yielding superior results. Nevertheless, the method remains challenging when it comes to handling the training demands of large-scale data. Lai et al. [10] employed Bidirectional Gated Recurrent Unit (BiGRU) for load forecasting, yielding substantial results, however, the training process remains time-consuming. Consequently, there is an urgent need to develop more efficient prediction models to meet the demand for high accuracy, real-time, and large-scale data processing in modern power systems.

The prevailing power load forecasting model is a combination model, which enhances the forecasting accuracy by integrating sub-modules with different functions. However, the integrated model generally has a greater number of parameters, which results in a reduction in the efficiency of hyper-parameter training. In addressing this challenge, Konyrbaev et al. [11] employed Bayesian optimization to optimize the hyperparameters of the combined model, thereby enhancing the training efficiency. However, this method is constrained by its limited adaptability to high-dimensional spaces and its substantial computational demands. Aqueel et al. [12] used genetic algorithm (GA) to optimize the hyperparameters of the combined model, which improves the training speed, but the method is prone to falling into local optimal solutions, which limits the potential for further performance enhancement. In contrast, many researchers have attempted to optimize the hyperparameters using heuristic methods. Jiang et al. [13] utilised the particle swarm optimization (PSO) algorithm to enhance the hyperparameters of combinatorial models, thereby improving the training efficiency. However, the method was slow to converge and exhibited limited accuracy when confronted with complex objective functions. In contrast, Phanden et al. [14] employed the simulated annealing (SA) algorithm for hyperparameter optimization, circumventing the predicament of local optimal solutions, which curtails performance. This approach, however, is sluggish to converge and inefficient in large-scale problems, despite its capacity to evade local optima. Recent advancements in the utilisation of optimization algorithms for hyperparameter optimization have yielded encouraging outcomes. Li et al. [15] proposed an innovative approach by employing the whale optimization algorithm (WOA) to enhance the training efficiency of a combined model. However, the method encounters certain limitations when confronted with intricate constraints. Similarly, Tang et al. [16] utilised the Sparrow Search Algorithm (SSA) to optimize the hyperparameters of the combined model, thereby markedly enhancing the training efficiency. Nevertheless, the method lacks stability and is overly simplistic. However, the stability and global search capability of this method still need to be further improved. Although intelligent optimization algorithms improve the training efficiency to a certain extent, they generally suffer from slow convergence speed and the tendency to fall into local optimal solutions. Therefore, there is an urgent need to develop more efficient optimization algorithms for the hyper-parameter training process of combinatorial models to meet the demand for high accuracy and fast training for power load forecasting.

This paper proposes a combined model using an Improved Sparrow Search Algorithm (ISSA)-CNN-BiGRU-MH-Attention to address accuracy and timeliness challenges [16]. The model integrates CNN for local feature extraction, BiGRU for long-term dependencies, and MH-Attention for key feature focus. An enhanced SSA optimizes the hyperparameter process with an adaptive update mechanism and hybrid heuristic strategy. The model offers high-precision predictions, better training efficiency, and flexibility for large-scale real-time forecasting in power systems.

2. Combined prediction model

This paper proposes a combined forecasting model based on CNN-BiGRU-MH-Attention to meet the demands for high accuracy and timeliness in power load forecasting. First, to address the limitations of the BiGRU algorithm in capturing long-term dependencies and handling complex time-

series data, a CNN module is introduced to enhance the model's ability to handle short-term dependencies. The next challenge is the inability of the CNN-BiGRU module to effectively focus on the most important features in the input data. To overcome this, the MH-Attention module is added to dynamically assign varying weights to input data at different time steps, thereby improving the model's adaptability and ability to focus on key features. The proposed combined model effectively captures the time-series characteristics and complex nonlinear relationships of power loads, improving the accuracy and stability of short-term load forecasting. It is well-suited for power system scheduling and load management applications, as demonstrated in Figure 1.

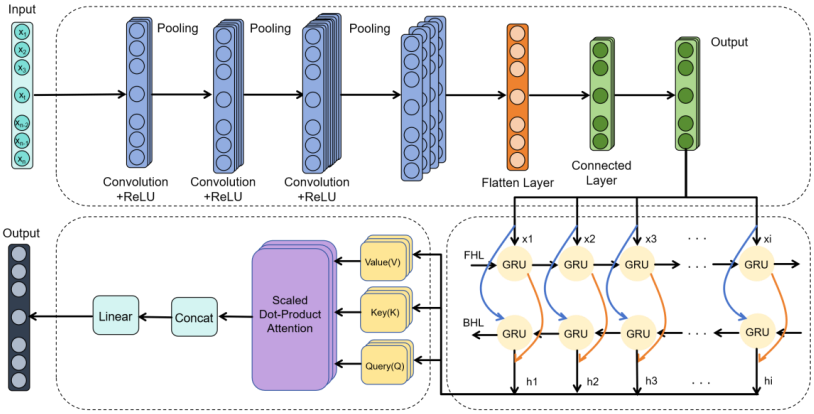

Figure 1. Structure of CNN-BiGRU-MH-Attention combined prediction model

2.1. CNN

The CNN [7] module captures the temporal change pattern of the data by extracting local features in the input data. Through convolutional operations, it can automatically learn the local correlations between different time steps, enabling the model to better recognise short-term dependencies and improve the response to power load fluctuations. The features learned by each convolutional kernel are shared throughout the input data, thereby reducing the number of parameters and enhancing the training efficiency of the model.

The input data is represented by a sequence $X=[x_1, x_2,\ldots, x_t]$, where x_i denotes the power load data at time step i. The fundamental formulation of the convolution operation is as follows:

$$y_i = \sum_{k=1}^{K} w_k \cdot x_{i+k-1} + b \tag{1}$$

Where y_i is the output feature after convolution, w_k is the parameter of the convolution kernel, b is the bias term, K is the length of the convolution kernel, and x_{i+k-1} is the local window data in the input sequence. The 1D convolution operation involves the gradual movement of the convolution kernel, w_k, over the input data, thereby extracting local features in a stepwise manner. The reiteration of this process, through multiple convolution operations, leads to the generation of varying levels of feature representations. Consequently, these higher levels of feature representation serve to enhance the feature representation of the model.

2.2. BiGRU

BiGRU [10] enhances RNN technology by effectively addressing long-term dependency challenges in time-series data. It processes input data through both forward and reverse GRU networks. The outputs from both directions are then merged to form a comprehensive representation of timing features. The BiGRU model consists of two GRU units: one for forward sequences and the other for reverse sequences, each with update and reset gates to regulate the flow of information directionally. The reset gate r_t for the forward GRU is calculated as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{2}$$

Where σ is the sigmoid activation function, W_r and U_r are the weight matrices, x_t is the input of the current moment, and h_{t-1} is the output of the previous moment. The update gate Z_t controls how much information is retained at the current moment, its formula is:

$$Z_t = \sigma（W_z x_t + U_z h_{t-1}）\tag{3}$$

Candidate Hidden State $\hat{h}_t$ Calculation formula:

$$\hat{h}_t = \tanh(W_h x_t + U_h(r_t \cdot h_{t-1}))\tag{4}$$

Where tanh is the hyperbolic tangent function, W_h and U_h are the weight matrices, and $r_t \cdot h_{t-1}$ denotes the gating value of the hidden state at the previous moment. The final hidden state h_t is calculated by the formula:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \hat{h}_t\tag{5}$$

The computation of the reverse GRU is similar to the forward GRU, except that the time order is reversed and the processing is completely symmetric with the forward GRU. BiGRU merges the forward and reverse hidden states to get the final bi-directional hidden state:

$$h_t^{bi} = [h_t; h_t^{back}]\tag{6}$$

Where $[\cdot,\cdot]$ indicates that the forward and reverse hidden states are stitched together to form a richer representation.

2.3. MH-Attention

The Multi-Headed Attention mechanism [17] addresses computational inefficiency and inadequate long-term dependency capture in traditional sequence models. It allows the model to focus on important information at different locations, enhancing its ability to learn complex patterns. Multiple attention heads compute in parallel to capture different features. Each query vector calculates a dot product with all key vectors to obtain a correlation score, which is processed by Softmax to get attention weights. The value vectors are weighted and summed based on these weights. The results from multiple heads are concatenated to produce the final output. Assuming an input sequence $X=[x_1, x_2,..., x_n]$, compute the query, key and value vectors for each element, compute the attention weights, and perform data splicing to obtain the multi-head attention output. Specific steps include: input data is linearly transformed with the weight matrix with the formula:

$$Q = XW_Q\tag{7}$$

$$K = XW_K\tag{8}$$

$$V = XW_V\tag{9}$$

Where W_Q, W_K and W_V are the learned weight matrices. Calculating Attention Weights: for each query, the relevance is measured by calculating the dot product of the query and the keys, scaled, and then the normalized weights are obtained by Softmax. The formula for this is:

$$Attention(Q,K,V) = soft\max(\frac{QK^T}{\sqrt{d_k}})V\tag{10}$$

Where $\frac{QK^T}{\sqrt{d_k}}$ is the dot product of the query and key, d_k is the dimension of the key, and the Softmax operation ensures that all weights sum to 1. Multi-head Attention: enables the model to learn different subspace features by computing multiple attention heads in parallel. Each attention head has an independent query, key, and value, and computes the respective outputs using the above formulas.

These outputs are then stitched together and linearly transformed to obtain the final multi-head attention output. The formula for this is:

$$\text{MultiHead}(Q,K,V) = \text{Con}cat(head_1, head_2, ..., head_h)Wo \tag{11}$$

Where $head_i = Attention(QW_{Q_i}, KW_{K_i}, VW_{V_i})$, Wo are the output weight matrices obtained from learning and h is the number of attention heads.

The CNN-BiGRU-MH-Attention model proposed in this paper is a combined model, which has more number of parameters compared to BiGRU to better capture complex temporal features and nonlinear relationships, but it also reduces the parameter training efficiency of the model. Currently, the mainstream hyperparameter optimization algorithms are dominated by heuristic algorithms, but these algorithms are slow to converge and inefficient when dealing with complex objective functions or dealing with large-scale problems.

3. Improved SSA algorithm

This paper introduces an enhanced SSA algorithm for optimizing the hyperparameter search process of the CNN-BiGRU-MH-Attention combined model. Traditional SSA algorithms often struggle with high-dimensional and complex problems, tending to converge to local optima. Additionally, the search process is rigid in terms of step size and direction, which slows down convergence. To address these issues, we propose an adaptive update mechanism and a hybrid heuristic strategy. The adaptive update mechanism dynamically adjusts step size and search direction based on the current fitness value of individual sparrows, preventing overly large or small step sizes. This improves the global search capability and accelerates convergence. The hybrid heuristic strategy combines the global search ability of PSO with the local search advantages of SSA, enhancing the algorithm's performance in high-dimensional spaces. The enhanced SSA algorithm significantly improves convergence speed, avoids local optima, and increases search accuracy.

3.1. Traditional SSA algorithm

The SSA [16] follows these steps: Initialization: A population of sparrows is randomly generated, with each individual representing a potential solution. Adaptation Evaluation: Each sparrow's adaptability is evaluated, reflecting the quality of the solution, with better-adapted sparrows considered better solutions. Behavioral Update: Sparrows decide between global exploration or local exploitation based on their fitness. Better-adapted individuals choose local exploitation, adjusting the search direction, while less adapted ones perform global exploration.

Observer Update: Observers search for potential high-quality solutions around sparrows and follow promising individuals, mainly identifying local optima. In the traditional SSA algorithm, the position and velocity are updated by the following equations: position update: the current position x_i of each sparrow is updated by the formula:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{12}$$

Where x_i^t denotes the current position of the i-th sparrow in generation t. v_i^{t+1} denotes the speed of the i-th sparrow in generation t+1. Speed Update: Sparrow's speed update formula is adjusted according to its exploratory and exploitative behaviors, and the common update strategies are as follows:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (x_{best}^t - x_i^t) + c_2 \cdot r_2 \cdot (x_{global}^t - x_i^t) \tag{13}$$

Where w is the inertia weight, c_1 and c_2 are the acceleration constants, r_1 and r_2 are the random numbers, x_{best}^t is the individual optimal solution, x_{global}^t is the global optimal solution. Behavioral switching: Individual sparrows decide whether to explore (global search) or exploit (local search) based on their fitness values. If the individual's current fitness is poor, it will perform more

extensive exploration behavior, if it is better fit, the individual will perform local exploitation to fine-tune the search of the nearby solution space.

SSA has some limitations: it often lacks flexibility, leading to confinement in specific search regions, particularly with fixed step sizes and directions, resulting in inefficiency and lower solution quality. Additionally, SSA relies on a single local search strategy, limiting its global search ability. This inefficiency causes slow convergence or local optima in high-dimensional problems, highlighting the need for improvements in the algorithm.

3.2. Improvement strategies

(1) Adaptive updating mechanism

The adaptive update mechanism [18] adjusts the step length and search direction based on each sparrow's fitness value, improving search flexibility and efficiency. In this mechanism, step size and direction are dynamically adjusted. Sparrows with higher fitness may expand the search space, while those with lower fitness focus on more promising areas by reducing step size. Fitness Calculation: Fitness is evaluated using the objective function, indicating the sparrow's position in the solution space. Dynamic Step Size Adjustment: Step size is increased for higher fitness sparrows and decreased for lower fitness ones, ensuring efficiency. Search Direction Adjustment: The search direction is modified based on fitness, optimizing the search strategy and improving overall search efficiency and accuracy.

Let the position of the individual sparrow be x_t, its fitness value be $f(x_t)$, and the step size be Δx, then the update formula for the step size under the adaptive update mechanism is:

$$\Delta x_t = \eta \cdot \frac{f(x_{best}) - f(x_t)}{f(x_{best})} \tag{14}$$

Where Δx_t is the adjusted step length of the current sparrow individual, η is the constant controlling the change of step length, $f(x_{best})$ is the fitness value of the global optimal solution, $f(x_t)$ is the fitness value of the current sparrow individual. The step size Δx_t is determined by the fitness difference between the current sparrow individual and the global optimal individual, and the individual with larger fitness difference will get a larger step size in order to expand the search space. The search direction update formula is:

$$x_t^{new} = x_t + \Delta x_t \cdot sign(f(x_t) - f(x_{best})) \tag{15}$$

Where x_t^{new} is the updated sparrow position, $sign(f(x_t) - f(x_{best}))$ denotes the search direction.

With these two formulas, the sparrow's step size and direction can be dynamically adjusted according to the current fitness value, which makes the search process more intelligent and avoids the fixed search strategy in traditional SSA, thus improving the search efficiency and the quality of the final solution.

(2) Hybrid heuristics

The hybrid heuristic strategy [19] integrates the SSA and PSO algorithms. The PSO algorithm explores the solution space broadly, leveraging the global search ability of the particles, while the SSA algorithm focuses on fine-tuning within the local region through local search. This combination enables the optimization process to search globally without losing accuracy and to refine solutions locally without easily getting stuck in local optima. The hybrid heuristic strategy operates as follows: Global search (PSO part): The PSO algorithm is used to explore the entire solution space, identifying potential superior solutions through its global search strategy. Local search (SSA part): The SSA algorithm then refines these solutions by conducting a more detailed search within the local region to improve their accuracy.

An alternative mechanism proposes alternating between the two algorithms: the SSA refines the solutions after each PSO global search, and PSO performs another global search after each SSA

optimization. This alternating mechanism ensures that the global search avoids falling into local optima, while the local search continuously enhances the accuracy of the solutions, ultimately leading to a better global optimal solution. The particle swarm optimization algorithm searches for the optimal solution by the position and velocity of the particles. The formulas are as follows: velocity update formula:

$$v_i^{new} = w \cdot v_i + c_1 \cdot r_1 \cdot (p_i - x_i) + c_2 \cdot r_2 \cdot (g - x_i) \tag{16}$$

Where v_i^{new} is the update speed of the i-th particle, w is the inertia weight, c_1 and c_2 are the learning factors, r_1 and r_2 are the random numbers, p_i is the individual best position of the i-th particle, g is the global best position, x_i is the particle current position. The position update formula:

$$x_i^{new} = x_i \cdot v_i^{new} \tag{17}$$

Where x_i^{new} is the position update of the particle. In the hybrid heuristic strategy, the sparrow performs local search tuning through its fitness value and position:

$$\Delta x_t = \eta \cdot \frac{f(x_{best}) - f(x_t)}{f(x_{best})} \tag{18}$$

Where Δx_t is the step size adjustment of the current sparrow individual, $f(x_{best})$ is the fitness value of the global optimal solution, $f(x_t)$ is the fitness value of the current sparrow individual. The position update formula is:

$$x_t^{new} = x_t + \Delta x_t \cdot sign(f(x_t) - f(x_{best})) \tag{19}$$

Where $sign(f(x_t) - f(x_{best}))$ determines the direction of the update. The hybrid heuristic strategy enhances the optimization by alternating the update strategies of PSO and SSA. Suppose that at a certain stage, after the PSO algorithm performs a global search, the sparrow search algorithm makes local adjustments based on this formula:

$$x_t^{new} = \alpha \cdot x_t^{PSO} + (1 - \alpha) \cdot x_t^{SSA} \tag{20}$$

Where x_t^{PSO} is the position obtained by particle swarm optimization. x_t^{SSA} is the position adjusted by sparrow search, α is the weight coefficient, which is used to balance the effects of PSO and SSA.

3.3. Algorithm optimization process

Figure 2 illustrates the flowchart for optimizing the CNN-BiGRU-MH-Attention model using the ISSA optimization algorithm.

As shown in Figure 2, the model's basic parameters are first entered, influencing the optimization process. The data is divided into training, validation, and test sets for model training, hyperparameter tuning, and performance evaluation, respectively. In the ISSA optimization, the sparrow positions represent model parameters and are randomly initialized to start the search. The adaptation value, based on model performance, is calculated, with lower values indicating better models. The adaptive update mechanism adjusts step size based on fitness: it increases step size for higher fitness and decreases it for lower fitness to focus on promising regions. The position of sparrows is updated using the formula. In the hybrid heuristic strategy (PSO+SSA), a mutation operation generates new solutions, followed by a crossover to merge mutated vectors with the target solution, and a selection operation to choose the optimal solution as the new sparrow position. After each iteration, the parameters are updated and passed to the CNN-BiGRU-MH-Attention model for training. The process terminates when the maximum iterations are reached or adaptation no longer improves, with the optimal parameters output for the final prediction task.

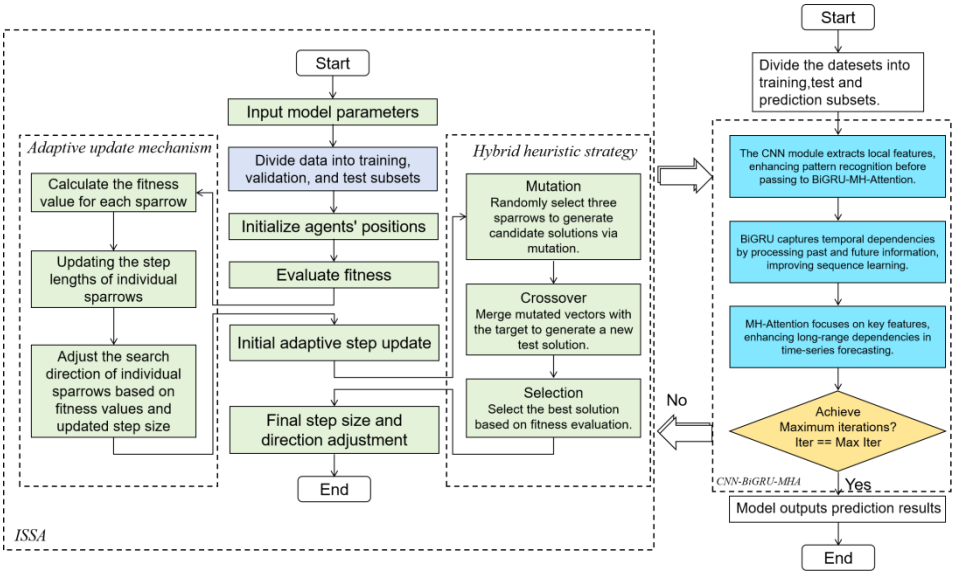

Figure 2. Flowchart of optimization algorithm

4. Experimentation and Analysis

4.1. Simulation environment

(1) Data set

In this paper, the power load dataset of a power plant in Quanzhou, a southern region, from 2016-1-1 0:00:00 to 2016-1-31 23:45:00 is selected as a sample, with a total of 2976 data samples and a sampling frequency of 15 minutes. The training set, test set as well as validation set's are 70%, 20% and 10% of the total sample size respectively. The input variables are data related to electricity consumption of power load.

(2) Simulation environment

The computer environment used in this paper is shown in Table 1.

Table 1. Experimental environment

Parameter Name	Parameter
CPU	Intel(R) Core(TM) i9-14900HX 2.20 GHz
Video card	NVIDIA GeForce RTX 4070 Laptop GPU 8GB
Random access memory (RAM)	32GB
Code language	Python 3.8
Hardware	Pycharm 2024.2.1

(3) Parameterization

The model parameters used in this paper are shown in Tables 2 and 3.

Table 2. Parameters related to the combined model

Model Parameter Name	Parameter value
Time_step	12
Filters	128
Kernel_size	3
Batch_size	16
Epochs	100
Learning_rate	0.001

Table 3. ISSA related parameters

Model Parameter Name	Parameter value
Population_size	50
Max_iterations	100
Alpha	0.5
Beta	0.7
Inertia_weight	0.9
Cognitive_weight	2.0
Social_weight	2.0

(4) Assessment indicators

In order to assess the accuracy of the model, Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and coefficient of determination: R^2 (R-Square) are chosen as the assessment indexes of the model in short-term power load forecasting, RMSE and MAE are the core indexes for assessing the model forecasting accuracy, the closer its value is to 0, the higher the forecasting accuracy of the surface model and the smaller the error. R^2 reflects the goodness of the model's fitting, the closer its value is to 1, the better the model's fitting effect is, and the specific calculation formula is as follows:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{21}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|(y_i - \hat{y}_i)\right| \tag{22}$$

$$R^2 = 1 - \frac{\sum_{i}^{n}(\hat{y}_i - y_i)}{\sum_{i}^{n}(\overline{y_i} - y_i)} \tag{23}$$

Where n is the total number of test samples, y_i denotes the true value of the i-th sample point, and $\hat{y}_i$ denotes the predicted value of the i-th sample point. $\overline{y_i}$ denotes the mean of the i-th sample point.

4.2. Model validation

In order to verify the validity and applicability of the proposed model, we have compiled the trend of the Loss function during the training of the model and recorded it in Figure 3.

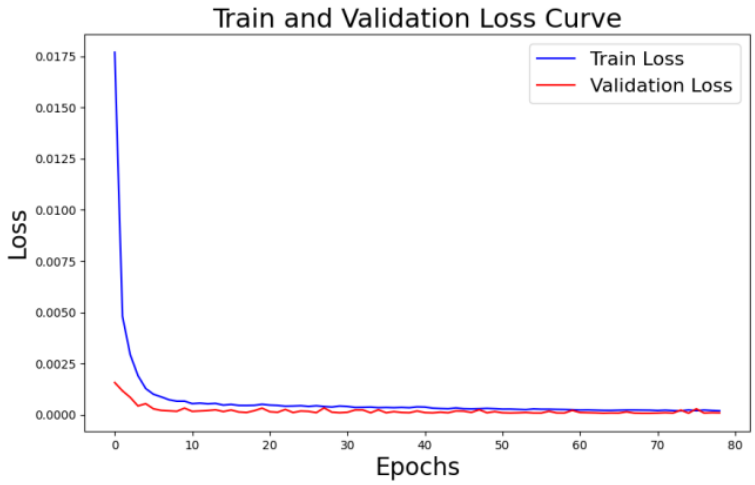

Figure 3. Algorithm Loss Diagram

As shown in Figure 3, the Loss curve shows a decreasing trend with the continuous increase of epoch. The curve shows a large decline in the early stage, when the epoch reaches about 50 the curve decline slows down, when the epoch reaches about 100 the curve tends to stabilize, the value tends to be close to 0, indicating that the model convergence is good, there is no obvious overfitting or underfitting phenomenon, the training process is effective, the model finds the optimal parameter selection, the model in the training process can effectively reduce the error and achieve a better performance. In addition, in order to represent the deviation of the model's predicted and true values, we organized the trends of the predicted and true values and recorded them in Figure 4.

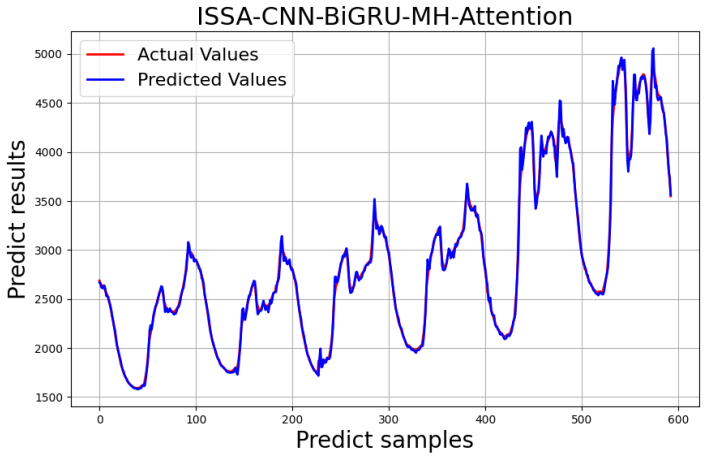

Figure 4. Comparison of real values and predicted values

As shown in Figure 4, the red curve represents the true value, and the blue curve represents the predicted value of the ISSA-CNN-BiGRU-MH-Attention combination model, and the predicted value of the combination model is very close to the true value, which indicates that the model has a high degree of fitting, and that the model is able to accurately capture the features of the data with strong accuracy and generalization performance, and it has a good load prediction capability.

Table 4. Indicators for model evaluation

Model	RMSE	MAE	R^2
Training	56.9596	34.6080	0.9955
Test	56.9584	34.6072	0.9934

The combined model evaluation metrics are shown in Table 4. The RMSE metric of the model reaches 0.9950 on the training set and 0.9934 on the test set, the MAE metric reaches 56.9596 on the training set and 56.9584 on the test set, and the R^2 metric reaches 34.60780 on the training set and 34.6072 on the test set, which can be reflected that the model has a good metrics evaluation result, and the model performance is good. The performance of the model is good. In summary, the Loss curve of the model converges and tends to 0, and the true value and the predicted value are very close to each other, and at the same time, the model has good evaluation indexes on both the training set and the test set, thus verifying the validity and applicability of the model.

4.3. Ablation experiments

To evaluate the contribution effect of individual modules in the proposed model, this paper will compare the performance of different combination modules (BiGRU, CNN-BiGRU, CNN-BiGRU-MH-Attention) on the same dataset with RMSE, MAE and R^2 as the evaluation metrics. The experimental results are shown in Table 5.

Table 5. Comparison of Indicators across Models

Model	RMSE	MAE	R^2
BiGRU	117.9121	82.7217	0.9807
CNN-BiGRU	99.0951	70.6370	0.9864
CNN-BiGRU-MH-Attention	83.1903	53.4445	0.9904

As shown in Table 5, the BiGRU model's test set results have an R^2 of 0.9807, RMSE of 117.91, and MAE of 82.72, indicating high fit but relatively large error. The combined CNN-BiGRU model improves response to power load fluctuations, reducing RMSE by 15.96%, MAE by 14.61%, and increasing R^2 by 0.58%. Introducing the multi-attention mechanism, the CNN-BiGRU-MH-Attention model further improves performance: RMSE decreases by 16.05% to 83.19, MAE drops by 24.34% to 53.44, and R^2 increases by 0.41% to 0.9904. These results show that optimizing the model structure reduces prediction error and improves fit. The inclusion of CNN and MH-Attention effectively enhances model performance. Deviations between predicted and true values for different models are shown in Figure 5.

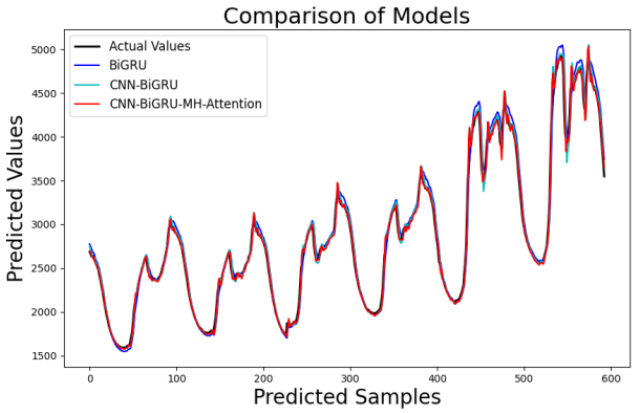

Figure 5. Comparison of different combination models

As shown in Figure 5, as the strategies are continuously introduced, the prediction curves fit more and more with the curves of the true values, and the final prediction trend fits almost perfectly with the true value trend, thus proving the effectiveness of each introduced strategy.

4.4. Comparative experiments

In order to verify the superiority of the combined CNN-BiGRU-MH-Attention model for short-term electricity load forecasting, this paper will compare the performance of differentforecasting models on the same dataset, and the comparison models include: the CNN-BiLSTM-SH-Attention, the CNN-LSTM-MH-Attention, the TCN-GRU-SENET, TCN-BiGRU-SH-Attention, and CNN-BiGRU-MH-Attention. and RMSE, MAE, and R^2 are used as evaluation metrics. The experimental results are shown in Table 6.

Table 6. Comparison of evaluation metrics of different strategies on the model

Model	RMSE	MAE	R^2
CNN-BiLSTM-SH-Attention	133.7149	95.9567	0.9753
CNN-LSTM-MH-Attention	105.6720	73.6905	0.9846
TCN-GRU-SENET	108.0688	77.8750	0.9838
TCN-BiGRU-SH-Attention	160.9901	128.6814	0.9640
CNN-BiGRU-MH-Attention	**83.1903**	**53.4445**	**0.9904**

As shown in Table 6, compared with other models, the R^2 of the CNN-BiGRU-MH-Attention combination model proposed in this paper improves by 0.6%~2.8% to 0.9904, the RMSE decreases by 21%~48% to 83.1903, and the MAE decreases by 27%~58% to 53.4445. the model's superiority was verified. In addition, in order to reflect the deviation between the predicted and true values of the different models, we recorded the trend of the predicted trend of the different models with respect to the true value, which is recorded in Figure 6.

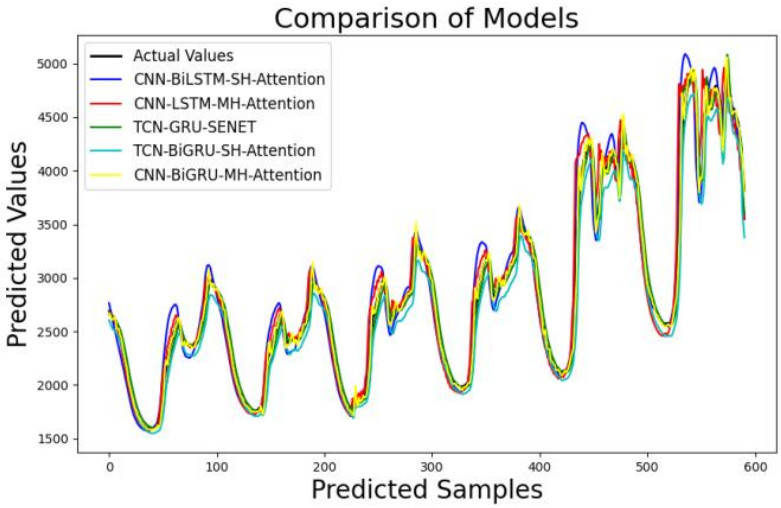

Figure 6. Comparison of the predictions of different strategies on the model

As shown in Figure 6, the black curve represents the true value, the dark blue curve represents the CNN-BiLSTM-SH-Attention model prediction trend, the red curve represents the CNN-LSTM-MH-Attention model prediction trend, the green curve represents the TCN-GRU-SENET model prediction trend, the light blue curve represents the TCN-BiGRU- SH-Attention model prediction trend, and the yellow curve represents the CNN-BiGRU-MH-Attention model prediction trend. The dark blue curve has the largest deviation, the rest of the models perform similarly to the true value, and the yellow curve almost completely overlaps withthe true value, thus proving the superiority of the models.

4.5. Algorithm Validation

In order to evaluate the effectiveness and applicability of the optimization function in the combined ISSA-CNN-BiGRU-MH-Attention model, we compiled the trends of the Loss function changes during the training of different models (CNN-BiGRU-MH-Attention, SSA-CNN-BiGRU-MH-Attention, PSSA-CNN-BiGRU-MH-Attention, ISSA-CNN-BiGRU-MH-Attention) Loss function trends during the training process and recorded in Figure 7.

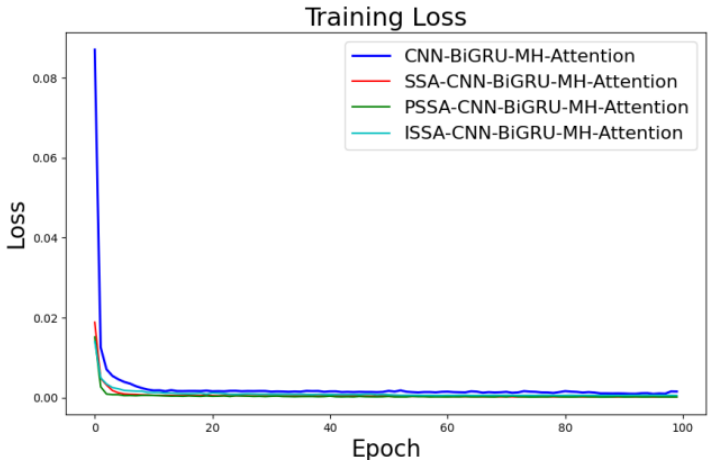

Figure 7. Loss Comparison Chart

As shown in Figure 7, where the dark blue curve indicates the training loss change trend of CNN-BiGRU-MH-Attention, the red curve indicates the training loss change trend of SSA-CNN-BiGRU-MH-Attention, the green curve indicates the training loss variation trend, and the light blue curve indicates the training loss variation trend of ISSA-CNN-BiGRU-MH-Attention. Among them, the ISSA-CNN-BiGRU-MH-Attention model has the lowest loss curve and the fastest convergence, thus verifying the effectiveness of the algorithm. Meanwhile, we record the evaluation metrics of different models in Table 7.

Table 7. Comparison of Indicators across Models

Model	RMSE	MAE	R^2
CNN-BiGRU-MH-Attention	83.1903	53.4445	0.9904
SSA-CNN-BiGRU-MH-Attention	69.8733	46.3787	0.9932
PSSA-CNN-BiGRU-MH-Attention	59.4776	36.8836	0.9951
ISSA-CNN-BiGRU-MH-Attention	**56.9596**	**34.6078**	**0.9950**

As shown in Table 7, the CNN-BiGRU-MH-Attention model has an R^2 of 53.44, RMSE of 0.99, and MAE of 83.19, indicating a good fit but large error. The SSA-CNN-BiGRU-MH-Attention model improves responsiveness, reducing MAE by 16.02% to 69.87, but RMSE increases slightly by 0.29% to 0.99, and R^2 drops by 13.22% to 46.38. The PSSA-CNN-BiGRU-MH-Attention model, with an adaptive update mechanism, further reduces MAE by 14.88% to 59.48, though RMSE increases slightly by 0.19% to 0.99, and R^2 decreases by 20.46% to 36.88. The ISSA-CNN-BiGRU-MH-Attention model, combining PSO, reduces MAE by 4.23% to 56.96, while RMSE increases slightly by 0.04% to 0.99, and R^2 decreases by 6.17% to 34.61. The results show consistent MAE reduction, indicating improved accuracy, despite slight fluctuations in RMSE and R^2. The adaptive and hybrid strategies enhance model performance, confirming the combined model's advantage in reducing load forecasting errors. Predicted values compared with real values are shown in Figure 8.

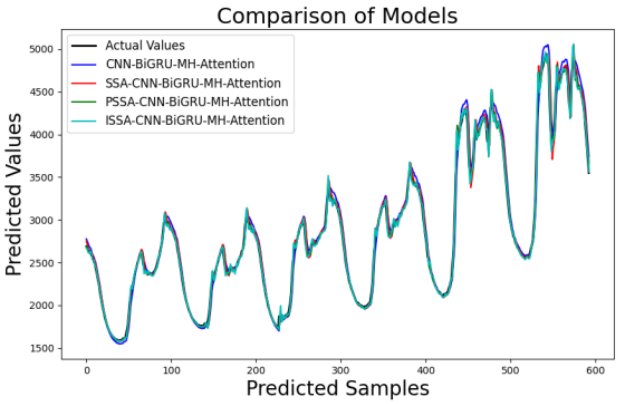

Figure 8. Comparison of predicted values of different models

As shown in Figure 8, comparison with other models, the predicted value change trend of ISSA-CNN-BiGRU-MH-Attention model is closest to the real value, thus verifying the effectiveness of the algorithm. To evaluate the effectiveness of optimization algorithms, this paper will compare the effectiveness of the optimization algorithms in different combination models on the same dataset with RMSE, MAE and R^2 as evaluation metrics. The experimental results are shown in Table 8.

Table 8. Comparison of Indicators across Models

Model	RMSE	MAE	R^2
IWOA-CNN-BILSTM-SH-Attention	84.4377	55.4939	0.9901
IGWO-CNN-LSTM-MH-Attention	63.0672	47.1101	0.9907
ISSA-TCN-GRU-SENET	60.8422	49.4437	0.9925
IPSO-TCN-BIGRU-SH-Attention	88.9362	68.5208	0.9891
ISSA-CNN-BiGRU-MH-Attention	**56.9596**	**34.6078**	**0.9950**

As shown in Table 8, compared with other models, the combined ISSA-CNN-BiGRU-MH-Attention model proposed in this paper improves the R^2 by 0.30% to 0.65% to reach 0.995, the RMSE reduces by 6.38% to 35.96% to reach 56.9596, and the MAE reduces by 26.54% to 49.52% to 34.6078, and the superiority of the model was verified. In order to more intuitively reflect the model prediction performance, we organize the predicted values of different models to compare with the real values, and the results are shown in Figure 9.

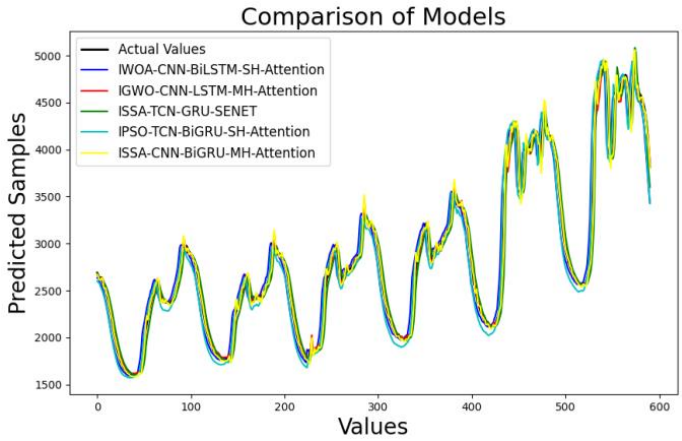

Figure 9. Comparison of predicted values of different models

As shown in Figure 9, comparison with other models, the predicted value change trend of ISSA-CNN-BiGRU-MH-Attention model is closest to the real value, thus verifying the effectiveness of the algorithm.

5. Conclusion

This paper proposes a combined forecasting model based on ISSA-CNN-BiGRU-MH-Attention to achieve high accuracy and timeliness in power load forecasting. First, to address BiGRU's limitations in capturing long-term dependencies and handling complex time-series data, we introduce a CNN module to extract local features and enhance the model's feature representation. Next, we incorporate the MH-Attention module, which uses multi-head self-attention to dynamically assign different weights to the input data at various time steps, boosting the model's adaptivity. Finally, we propose an improved SSA algorithm for hyperparameter training, enhancing model efficiency. The proposed model and algorithm are applied to the dataset of a power plant in Quanzhou, a southern region, for validation. The experimental results show that the proposed model has a good load forecasting capability, with R^2 reaching 0.9950, RMSE reaching 56.9596, and MAE reaching 34.6078, and the validity of the model is verified. Compared with other combined prediction models, R^2 improved by 0.3%~0.65%, RMSE decreased by 6.38%~35.96%, and MAE decreased by 26.54%~49.52%, and the superiority of the model was verified. The ISSA-CNN-BiGRU-MH-Attention combination model proposed in this paper combines the advantages of optimization algorithm, convolutional feature extraction and attention mechanism, and fully exploits the temporal features and key patterns in the power load data.

Despite the model's excellent performance in terms of accuracy, we also note that there is still room for improvement in its computational efficiency. In future research, in-depth improvements can be made to the optimization algorithm and more efficient hybrid optimization algorithms can be explored to enhance the computational efficiency and global search capability in the hyper-parameter optimization process. Meanwhile, focusing on the fusion and application of multimodal data will further improve the robustness and adaptability of the model to meet the needs of more complex application scenarios.

References

[1] Tan X Y, Ao G, Qian G C, et al. Research on power load forecasting using deep neural network and wavelet transform [J]. International Journal of Information Technologies and Systems Approach (IJITSA), 2023, 16 (2): 1-13.

[2] Wang S B, Luo W H, Yin S X, et al. Interpretable state estimation in power systems based on the kolmogorov-arnold networks [J]. Electronics, 2025, 14 (2): 320-339.

[3] Mohammad T A, Shohani A M A W. Short-Term prediction of the solar photovoltaic power output using nonlinear autoregressive exogenous inputs and artificial neural network techniques under different weather conditions [J]. Energies, 2024, 17 (23): 6153-6168.

[4] Sanjeev, Rohit K, Ajay S, et al. Development of seasonal ARIMA model to predict wholesale price of rice in Delhi market [J]. Current Journal of Applied Science and Technology, 2022, 155-161.

[5] Dabeeruddin S, Haitham R A, Ali G, et al. Household-level energy forecasting in smart buildings using a novel hybrid deep learning model [J]. IEEE ACCESS, 2021 (9): 33498-33511.

[6] Shi C Y, Jiang H F, Zhao F Z, et al. Blood metal levels predict digestive tract cancer risk using machine learning in a U.S. cohort [J]. Scientific Reports, 2025, 15 (1): 1285-1298.

[7] Hua Q, Fan Z, Mu W, et al. A short-term power load forecasting method using CNN-GRU with an attention mechanism [J]. Energies, 2024, 18 (1): 106-122.

[8] Hardanee A F O, Demirel H. Hydropower station status prediction using RNN and LSTM algorithms for fault detection [J]. Energies, 2024, 17 (22): 5599-5621.

[9] Yang G, Du S, Duan Q, et al. Short-term price forecasting method in electricity spot markets based on Attention-LSTM-mTCN [J]. Journal of Electrical Engineering & Technology, 2022, 17 (2): 1009-1018.

[10] Lai Y B, Wang Q F, Chen G, et al. VMD-BiGRU for short-term power load forecasting with energy valley optimizer enhancement [J]. Journal of Physics: Conference Series, 2024, 2868 (1): 012004-012012.

[11] Konyrbaev N, Nikitenko Y, Shtanko V, et al. Evaluation and optimization of the naive bayes algorithm for intrusion detection systems using the USB- IDS-1 dataset [J]. Eastern-European Journal of Enterprise Technologies, 2024, 6 (2): 74-82.

[12] Aqueel A, Kumar A Y, Achhaibar S. Enhancing waste cooking oil biodiesel yield and characteristics through machine learning, response surface methodology, and genetic algorithms for optimal utilization in CI engines [J]. International Journal of Green Energy, 2024, 21 (6): 1345-1365.

[13] Jiang Q K, Wang H Y. Risk assessment and hybrid algorithm transportation path optimization model for road transport of dangerous goods [J]. IATSS Research, 2025, 49 (1): 72-80.

[14] Mohammad B S, Mohsen T. Response estimation of reinforced concrete shear walls using artificial neural network and simulated annealing algorithm [J]. Structures, 2021, 34: 1155-1168.

[15] Li Y, Qian M, Dai D J, et al. Flow control of flow boiling experimental system by Whale Optimization Algorithm (WOA) improved single neuron PID [J]. Actuators, 2024, 14 (1): 5-22.

[16] Tang F. Short-term wind power prediction based on improved sparrow search algorithm optimized long short-term memory with peephole connections [J]. Wind Engineering, 2025, 49 (1): 71-90.

[17] Xu L L, Li F Y, Chang S. A fiber recognition framework based on multi-head attention mechanism [J]. Textile Research Journal, 2024, 94 (23-24): 2629-2640.

[18] Cai H, Lu K Z, Wu Q R, et al. Adaptive classification algorithm for concept drift data stream [J]. Journal of Computer Research and Development, 2022, 59 (03): 633-646.

[19] Ai S Y, Song Z M, Shen X. Branching heuristic strategy based on variable mixing features [J]. Computer Systems & Applications, 2020, 29 (03): 200-205.