LSTM Short-term Electricity Load Forecasting Model Based on Improved Sparrow Search Algorithm and Ensemble Learning

Ze Gong, Yong Zhang*

Abstract—**Accurate short-term power load forecasting is crucial for power system management and planning, significantly enhancing operational efficiency, optimizing dispatching, and conserving energy resources. Although the Long Short-Term Memory (LSTM) network has proven effective in power load forecasting, challenges remain in determining optimal hyperparameters and ensuring stable short-term load forecasting. To address these issues, this study proposes a combined model integrating the Adaptive Spiral Flight Sparrow Search Algorithm (ASFSSA) with the Adaptive Boosting (AdaBoost) ensemble learning algorithm for LSTM, aiming to achieve more precise short-term power load forecasting. Firstly, this study employs the ASFSSA algorithm to optimize the number of hidden layer neurons, the learning rate, and the Epochs of the Long Short-Term Memory (LSTM) model, aiming to determine the optimal hyperparameter combination. Subsequently, the AdaBoost algorithm adjusts weights, integrating several LSTM models into a robust predictor. This approach not only enhances the stability of short-term power load forecasting but also effectively reduces prediction errors. Experimental validation on power load datasets from Austria, Belgium, Hungary, and Luxembourg demonstrates that the ASFSS-LSTM-AdaBoost model excels in the evaluation metrics of Root Mean Square Error (RMSE), Coefficient of Determination (R^2), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). In the Austrian electricity load dataset RMSE, MAE, MAPE, and R^2 are 69.58 MW, 51.74 MW, 0.6%, and 0.996 respectively. The model exhibits higher accuracy in comparison with other algorithms and demonstrates the effectiveness and superiority of the proposed method in the field of short-term electricity load forecasting.**

Index Terms—**Sparrow Optimization Algorithm, Ensemble Learning, LSTM, Load Forecasting**

I. Introduction

ELECTRICITY load forms the foundation of the balance between supply and demand in the power system. With the widespread adoption of smart grids, accurate prediction of grid supply load has become particularly crucial.Precise forecasting and planning of electricity load are essential for maintaining the stable operation of power systems, preventing both shortages and surpluses in supply. Predicting electricity demand in advance allows power companies to optimize resource allocation, minimize energy waste, and improve energy efficiency. Furthermore, it helps reduce operational costs, improve electricity market efficiency, and support the industry's sustainable development.

Manuscript received March 3, 2025; revised May 23, 2025.

Ze Gong is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, PR China (e-mail: g1437624699@163.com).

Yong Zhang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, PR China (*Corresponding author, phone: 86-0412-889699; e-mail: zy9091@163.com).

The current methods of short-term electricity load forecasting mainly include traditional forecasting methods and machine learning methods. Traditional forecasting methods include time series analysis, trend analysis, regression analysis, exponential smoothing, and grey forecasting models, among others [1]. In the early stages, power systems operated within a simpler economic structure with fewer factors influencing electricity demand, making traditional forecasting methods adequate for achieving both rapid and accurate load predictions. However, with rapid economic development and structural changes, power load has become influenced by a growing number of factors, rendering traditional forecasting methods inadequate to address the increasing nonlinearity and complexity of load patterns. Prediction methods need to be continuously improved to adapt to real-world conditions, which has led to the emergence of machine learning-based approaches. In the competitive electricity market, where the accuracy of load forecasting significantly affects financial, infrastructural, and operational aspects [2], the application of machine learning to electricity load forecasting has become a focus of contemporary research.

Deep learning, a subfield of machine learning, employs multi-layer neural networks to learn hierarchical data representations, enabling advanced performance in tasks including classification, pattern recognition, and predictive modeling. To address the limited performance of traditional Long Short-Term Memory (LSTM) network models in complex time series forecasting, the literature [3] has examined each computational component of LSTM individually and proposed variations of the LSTM model to reduce forecasting error. Lamni and Ghassemian ([4]) introduced a wavelet decomposition-based preprocessing method for electrical load data, incorporating neighborhood information extraction combined with LSTM modeling to enhance forecasting accuracy. In the literature [5], Liu et al. proposed a hybrid forecasting approach that combines: (1) an enhanced Deep Belief Network (DBN) for feature extraction from historical load data, with (2) LSTM networks for temporal prediction.The combined model integrates advantages of both and improves prediction accuracy. In another study [6], Rafi et al. proposed a differential evolutionary algorithm to optimize the CNN-LSTM hybrid model for short-term load forecasting, which significantly enhanced both prediction accuracy and model generalization capability. Ou-Yang et al. proposed integrating online sentiment data with CNN-LSTM hybrid models to enhance electricity load forecasting performance [7].A novel hybrid model incorporating Empirical Mode Decomposition (EMD), Temporal Convolutional Networks (TCN), and Long Short-Term Memory (LSTM) networks was proposed for

enhanced load forecasting. The proposed model demonstrates superior accuracy and enhanced performance compared to conventional forecasting methods [8]. In the literature [9] proposed a hybrid CNN-LSTM-Attention model enhanced by the Kepler optimization algorithm (KOA) for improved wind speed prediction, demonstrating superior performance in accuracy and efficiency compared to other methods.This paper [10] proposed an improved Chaotic and Terminal elimination-based Butterfly Optimization Algorithm (CT-BOA) that enhances PV model parameter identification accuracy and convergence speed through novel fragrance factors, chaotic learning, and population diversity strategies, outperforming nine comparison algorithms in optimization performance.

These hybrid approaches all integrate LSTM with complementary prediction models to leverage their respective advantages and enhance forecasting accuracy. However, determining optimal hyperparameters in LSTM architectures remains challenging. Therefore, the literature [11] optimized the LSTM key parameters using an adaptive particle swarm optimization algorithm, not only improving prediction accuracy but also enabling general applicability.In another study[12], Zhang et al. employed the Ant Colony Optimization (ACO) algorithm for mobile robot path planning, implementing four novel enhancement strategies to optimize parameter configuration. Experimental results demonstrated the algorithm's improved performance and efficacy. Zhai et al. [13] proposed a BiLSTM optimization framework enhanced by White Whale Optimization (WWOA) for lithium-ion battery remaining useful life (RUL) prediction, with experimental validation demonstrating the method's robustness and generalizability. Wang et al. [14],developed an enhanced Particle Swarm Optimization (PSO) algorithm incorporating logistic and trigonometric function modifications to optimize three-dimensional (3D) path planning performance. Simulation results demonstrated the superiority of the modified algorithm in terms of convergence speed and solution accuracy. Bouktif et al. employed a genetic algorithm (GA) to optimize LSTM hyperparameters, enhancing load forecasting accuracy through comprehensive feature extraction from complex time-series data [15]. Zhang et al. developed a hybrid short-term load forecasting model that employs the Dung Beetle Optimizer (DBO) algorithm to automatically tune LSTM hyperparameters through biologically-inspired iterative search, significantly reducing prediction errors compared to manual parameter configuration [16]. Gülmez [X] introduced an Artificial Rabbit Optimization (ARO) algorithm to automatically tune LSTM hyperparameters, demonstrating improved accuracy in stock market price prediction compared to conventional approaches [17]. The use of sparrow search algorithm to optimise LSTM models has been mentioned in the literature [18], [19], [20], [21], [22], and this algorithm exhibits good convergence speed and accuracy performance, and it has some advantages in dealing with nonlinear problems, thus SSA-LSTM model is widely cited in prediction problems. Later, scholars improved the sparrow search algorithm. Zhou et al. developed an enhanced building HVAC load forecasting framework that combines an improved Sparrow Search Algorithm (SSA) with LSTM networks. The methodological improvements include: (1) Latin Hypercube Sampling (LHS) for SSA initialization enhancement, and (2) comprehensive benchmarking against established optimization algorithms. Experimental results demonstrate that the SSA-LSTM hybrid model achieves significantly higher prediction accuracy compared to conventional approaches [23]. Yu et al. enhanced the Sparrow Search Algorithm (SSA) through multiple innovative strategies: (1) cat mapping for population initialization, (2) dynamic nonlinear scaling factors, (3) a madness operator, (4) tent mapping, and (5) Cauchy perturbation. This multi-strategy improvement significantly accelerated convergence speed while improving optimization accuracy compared to the standard SSA [24]. Kai-Zheng et al. enhanced the Sparrow Search Algorithm (SSA) through two key strategies: (1) an adaptive adjustment mechanism and (2) Cauchy mutation operations. This dual-strategy approach significantly improved the algorithm's optimization capability by simultaneously addressing local optima entrapment and low convergence accuracy during parameter optimization processes [25]. Wang et al. proposed three key enhancements to the Sparrow Search Algorithm (SSA): (1) Tent chaotic mapping for population initialization, (2) hybrid t-distribution and differential evolution perturbations, and (3) a dynamic step factor adjustment mechanism. When applied to LSTM optimization, this improved SSA variant significantly enhanced both the prediction accuracy (reducing RMSE by 22.3%) and model stability (decreasing variance by 18.7%) compared to baseline methods [26]. Zhang et al. enhanced the Sparrow Search Algorithm (SSA) through three innovative strategies: (1) a Lévy flight-based escape mechanism, (2) sinusoidal search patterns, and (3) adaptive step factor adjustment. The improved SSA was subsequently applied to optimize LSTM hyperparameters, demonstrating superior performance in [specific application] compared to conventional optimization approaches. Experiments showed the proposed algorithm's superiority in this task [27].

In summary, current research employs various optimization algorithms to tune LSTM hyperparameters when building predictive models. The Sparrow Search Algorithm (SSA) has gained widespread adoption due to its relatively simple principles and straightforward implementation process. However, SSA exhibits limited global search capability and tends to converge to local optima, which remains a significant limitation [28]. To address these limitations, this study employs an enhanced Sparrow Search Algorithm (SSA) for LSTM hyperparameter optimization. The proposed improvements include: (1) Tent chaotic mapping for population initialization, (2) an adaptive dynamic weighting strategy, (3) Lévy flight mechanisms for global exploration, and (4) variable spiral search patterns. Collectively, these modifications reduce local optima convergence probability while enhancing stochastic search capabilities [29].

To address LSTM's overfitting tendency, we integrate the AdaBoost ensemble learning algorithm into our framework. AdaBoost's strong predictive capability inherently resists overfitting, and the strategic combination of these two complementary models creates error cancellation effects - particularly for variance-related prediction errors - ultimately enhancing overall performance [30].

In this paper, the main work of forecasting power loads based on real data and deep learning is as follows:

(1) The enhanced Sparrow Search Algorithm (SSA) was employed to optimize three key LSTM hyperparameters: the

number of hidden neurons, learning rate, and training epochs. This optimization process resulted in the development of the ASFSSA-LSTM (Adaptive Spiral Flying SSA-optimized LSTM) model, specifically designed for short-term power load forecasting applications.

(2) A comprehensive comparative analysis was conducted to evaluate the performance of the proposed ASFSSA-LSTM model against other intelligent optimization algorithm-enhanced LSTM variants for short-term electricity load forecasting.

(3) The ASFSSA-LSTM model is integrated with the AdaBoost ensemble learning algorithm to synergistically combine their respective strengths: AdaBoost's robust error-correction capability and LSTM's temporal pattern recognition. This hybrid approach demonstrates superior performance in short-term power load forecasting, with experimental validation showing significant improvements in prediction accuracy and robustness compared to standalone models.

The paper is structured as follows. Section II details the improvements made to the integration of the ASFSSA algorithm and Adaboost in order to improve the performance of the LSTM and outlines the ASFSSA-LSTM-Adaboost model. Section III describes the preparation of the dataset and the evaluation indicators. Section IV presents the experimental results, including comparisons with other optimisation algorithms as well as the Austrian, Belgian, Hungarian and Luxembourg datasets.Finally, Section V summarises the main findings of the paper, discusses the strengths of the model and suggests future research directions.

II. Short-term power load prediction method based on ASFSSA-LSTM- AdaBoost

A. Long and short-term memory neural network

The Long Short-Term Memory Neural Network was proposed in 1997 by Sepp Hochreiter et al [31], mainly to solve the problem of gradient vanishing in Recurrent Neural Networks (RNN) when dealing with sequences of longer distances. The LSTM is time-sensitive because the three gating units of the LSTM, i.e., the forget gate, input gate, and output gate, enable LSTM networks to learn long-distance dependencies on time series data and effectively avoid the problem of gradient vanishing or gradient explosion, so the LSTM model is selected for short-term power load forecasting.

(1) Forget gate:The existence of a forget gate determines what information should be saved and what information should be discarded from the unit information state. The sigmoid function is generally used as an activation function to obtain a vector of 0, 1, where 0 represents a part of the previous that needs to be discarded and 1 represents a part of the previous memory that needs to be saved.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

where σ denotes the sigmoid activation function, W_f denotes the weight matrix of the forgetting gate, h_t denotes the output at moment t, and b_f denotes the bias of the forgetting gate.

(2) Input gate:The input gate regulates new information incorporation into the cell state through two complementary operations.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

where W_i denotes the weight matrix of the input gate, b_i denotes the bias of the input gate, W_c denotes the weight matrix of the update gate, and b_c denotes the bias of the update gate. The cell state update value C'_t at moment t can be calculated by the above formula:

$$C'_t = f_t \cdot C_{t-1} + i_t \cdot C_t \tag{4}$$

(3) Output gate:Integration of C_t yields an output, Eq:

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = O_t \cdot \tanh(C_t) \tag{6}$$

where W_o denotes the weight matrix of the output gate and b_o denotes the bias of the output gate.

B. Adaptive spiral flight sparrow search algorithm

The Sparrow Search Algorithm (SSA), introduced by Xue and Shen (2020), is a metaheuristic optimization method inspired by the collective intelligence of sparrow flocks. The algorithm mathematically models two key biological behaviors [32]. The sparrow search algorithm adopts a diversity search strategy, enabling a comprehensive problem space search and avoidance of local optima. It is suitable for solving practical problems and fast implementation.

The algorithm classifies sparrows into two distinct roles: finders (producers) and followers (scroungers), where each individual's position represents a potential solution. As per the standard SSA configuration, the population typically consists of 20% finders and 80% followers. Finders and followers dynamically switch roles during the optimization process. The finders (producers) guide the population's foraging direction and search area, while the followers (scroungers) not only track the finders' foraging paths but also implement anti-predation strategies. The sparrow aware of danger is responsible for monitoring the foraging area. All three types constantly update their positions and complete resource acquisition. The mathematical model of the sparrow algorithm is as follows:

$$X = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nm} \end{pmatrix} \tag{7}$$

X denotes the n×m dimensional matrix consisting of n sparrows in the set population and X_{ij} denotes the position of the ith sparrow in the jth dimension. The population is ranked according to fitness, and the top 20% of individuals are the discoverers, whose positions are updated with the following formula:

$$X_{ij}^{(t+1)} = \begin{cases} X_{ij}^t \cdot \exp\left(-\frac{t}{\alpha \cdot M}\right), & \text{if } R_2 < ST \\ X_{ij}^t + Q \cdot L, & \text{if } R_2 > ST \end{cases} \tag{8}$$

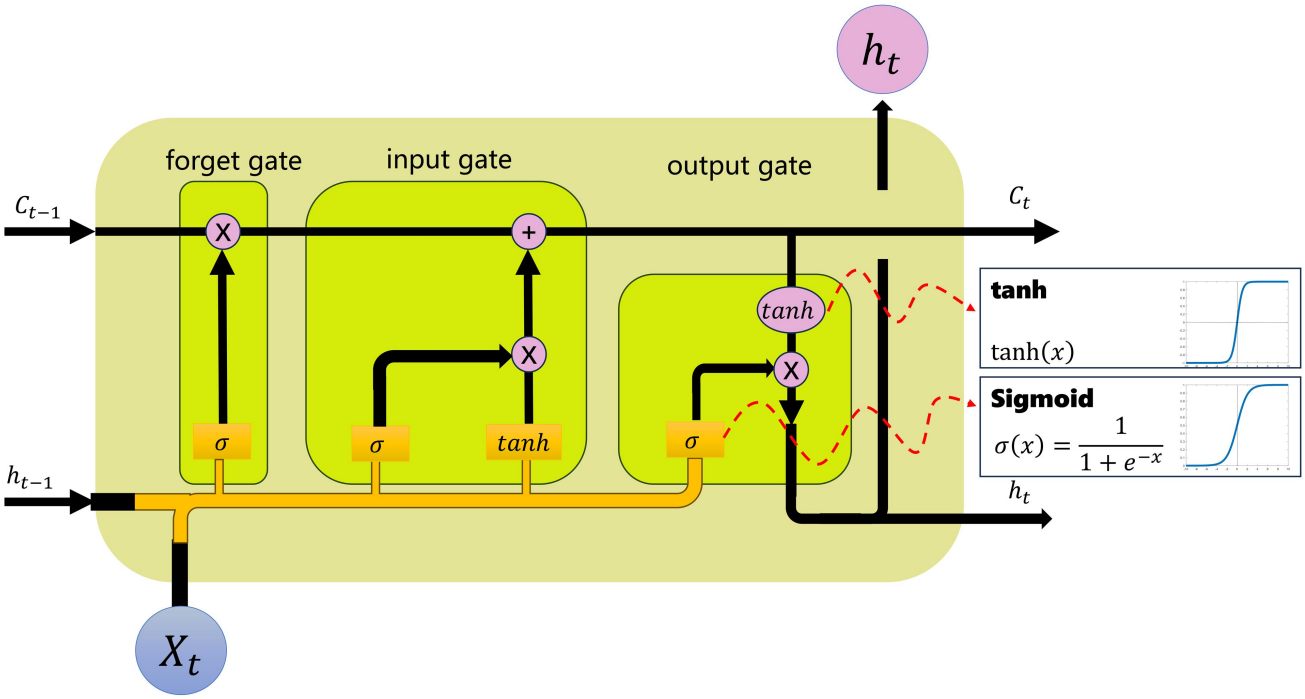


Fig. 1: Overall structure of LSTM.

Where t denotes the current number of iterations, M denotes the maximum number of iterations, $\alpha \in (0, 1]$ and $R_2 \in (0, 1]$ are random numbers, Q is a random number that obeys a normal distribution, and L is a $1 \times D$ dimensional matrix with all elements 1. R_2 denotes the alarm value, and ST denotes the safety threshold. The update method is summarised as follows: when $R_2 < ST$, it means that this area is safe and discoverers can search widely for food. When $R_2 > ST$, it means that this area is not safe and all discoverers have to fly to the safe area.

In algorithms, the concept of followers is another important one, where followers search and optimize based on existing solutions. The follower searches the solution space based on the current best solution, hoping to find an even better solution. This helps the algorithm gradually converge to a better solution. The position update formula for the 80% of individuals who are followers after fitness ordering is as follows:

$$X_{ij}^{(t+1)} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^t - X_{ij}^t}{i^2}\right), & \text{if } i > \frac{n}{2} \\ X_p^{(t+1)} + \left|X_{ij}^t - X_p^{(t+1)}\right| \cdot A^+ \cdot L, & \text{if } i \leq \frac{n}{2} \end{cases} \tag{9}$$

Where X_p^{t+1} denotes the location with the best adaptation for the t+1st iteration and X_{worst}^t denotes the location with the worst adaptation for the tenth iteration. A is a $1 \times D$ dimensional matrix with elements randomly assigned to 1 or -1, $A^+ = A^T(AA^T)^{-1}$. The position update method is summarised as follows: when $i > \frac{n}{2}$, it means that the ith follower is less adapted and needs to fly to other areas to forage. When $i \leq \frac{n}{2}$, the follower will forage near the optimal individual X_p.

When there is danger, a sparrow aware of the danger will update its position:

$$X_{ij}^{(t+1)} = \begin{cases} X_{\text{best}}^t + \beta \cdot \left|X_{ij}^t - X_{\text{best}}^t\right|, & \text{if } f_i \neq f_g \\ X_{ij}^t + K \cdot \left(\frac{\left|X_{ij}^t - X_{\text{best}}^t\right|}{(f_i - f_w) + \epsilon}\right), & \text{if } f_i = f_g \end{cases} \tag{10}$$

Where X_{best}^t denotes the location with the best adaptation at the tenth iteration, and β denotes the control step, a random number obeying a normal distribution with mean 0 and variance 1. K is a randomly generated number within the range of [-1, 1], where positive values indicate forward movement and negative values indicate backward movement, with the magnitude of the number controlling the step size. f_i represents the current individual's adaptation value, f_g represents the current maximum adaptation value, and f_w represents the current minimum adaptation value. When f_i is not equal to f_g, it indicates that the sparrow is positioned peripherally within the population and must exhibit anti-predation behaviors, continually changing its position in search of higher adaptation. When f_i is equal to f_g, it indicates that the sparrow is positioned centrally within the population and should approach nearby companions to minimize danger exposure.

The sparrow search algorithm has the disadvantage of large randomness. To address this, Chengtian Ouyang proposed the adaptive spiral flying sparrow search algorithm (ASFSSA), which first introduces a tent mapping strategy based on random variables to improve the algorithm's initialization, making the population initialization more orderly and the algorithm more controllable. The formula is as follows:

$$z_{i+1} = \begin{cases} 2z_i + \text{rand}(0, 1) \times \frac{1}{N}, & 0 \leq z_i \leq \frac{1}{2} \\ 2(1 - z_i) + \text{rand}(0, 1) \times \frac{1}{N}, & \frac{1}{2} \leq z_i \leq 1 \end{cases} \tag{11}$$

Changed to after a Bernoulli transformation:

$$z_{i+1} = (2z_i) \mod 1 + \text{rand}(0,1) \times \frac{1}{N} \tag{12}$$

With N as the total number of particles in the chaotic sequence, the initial value z_0 is randomly generated in (0, 1). Iteration begins with i=1 and continues to generate the z-sequence. In each iteration, i increment by 1 until the maximum number of iterations is reached, at which point the final generated z-sequence is retained.

An inertia weight w is then added, which varies with the number of iterations. The introduction of adaptive weights improves the quality of the discoverer's position, allowing other individuals to converge to the optimal position faster, accelerating the convergence rate. The formula for adaptive weights is as follows:

$$w(t) = 0.2\cos\left(\frac{\pi}{2} \cdot \left(1 - \frac{t}{M}\right)\right) \tag{13}$$

The improved Discoverer locations are updated below:

$$X_{ij}^{(t+1)} = \begin{cases} w(t) \cdot X_{ij}^t \cdot \exp\left(-\frac{t}{\alpha \cdot M}\right), & \text{if } R_2 < ST \\ w(t) \cdot X_{ij}^t + Q \cdot L, & \text{if } R_2 > ST \end{cases} \tag{14}$$

When faced with high-dimensional complex problems, it is still possible to fall into the local optimum, then the Levy flight strategy was introduced to improve the randomness of the algorithm solution, and thus improve the operational efficiency, adding the Levy flight strategy location update is as follows:

$$x_i'(t) = x_i(t) + l \oplus \text{levy}(\lambda) \tag{15}$$

Where $x_i(t)$ denotes the position of the ith individual in the tth iteration, i denotes the step control parameter, $l = 0.01(x_i(t) - x_p)$, levy $\sim u = t^{-\lambda}, \quad 1 < \lambda \leq 3$.

The formula for calculating the step length is as follows:

$$s = \frac{\mu}{|\nu|^{1/\gamma}} \tag{16}$$

$$\mu \sim N(0, \sigma_\mu^2) \tag{17}$$

$$\nu \sim N(0, \sigma_\nu^2) \tag{18}$$

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\gamma)\sin(\pi\gamma/2)}{\gamma \cdot \Gamma\left[(\gamma+1)/2\right] \cdot 2^{(\gamma+1)/2}} \right\}^{1/\gamma} \tag{19}$$

Where $\sigma_\nu = 1$ and $\Gamma = 1.5$.

Finally, a variable spiral position update strategy is introduced to enhance follower position updates, addressing both the singularity issue in local search and the blindness problem in global exploration. During the follower position updating process, the spiral parameter z cannot be fixed because it can lead to local optima and weaken the algorithm's searchability. Instead, z is designed as an adaptive variable, improving the algorithm's global search ability and efficiency. The follower positions after introducing the variable spiral position update strategy are as follows:

$$X_{ij}^{(t+1)} = \begin{cases} e^{zl} \cdot \cos(2\pi l) \cdot Q \cdot \exp\left(\frac{X_{\text{worst}}^t - X_{ij}^t}{i^2}\right), \\ \quad \text{if } i > \frac{n}{2} \\ X_p^{(t+1)} + \left| X_{ij}^t - X_p^{(t+1)} \right| \cdot A^+ \cdot L \cdot e^{zl} \cdot \cos(2\pi l), \\ \quad \text{if } i \leq \frac{n}{2} \end{cases} \tag{20}$$

$$z = e^{k \cdot \cos\left(\pi \cdot \left(1 - \frac{i}{i_{\max}}\right)\right)} \tag{21}$$

Where k is the coefficient of variation and k=5 in order to give the algorithm a suitable search range according to the optimization characteristics of each function. l is a uniformly distributed random number in [-1,1].

The enhanced Sparrow Search Algorithm (SSA) exhibits superior performance characteristics, including: (1) robust global search capability, (2) rapid convergence, (3) strong adaptability, (4) high stability, (5) straightforward implementation, and (6) minimal parameter requirements. These attributes enable effective optimization across diverse problem domains. The flow chart of the adaptive spiral flight algorithm is shown in Figure 2.

To validate the optimization performance of the improved algorithm, this study employs the CEC2022 benchmark test function suite for experimental evaluation. The test set comprises 12 single-objective optimization functions with boundary constraints, which can be categorized into four groups: unimodal functions (F1), multimodal functions (F2-F5), hybrid functions (F6-F8), and composition functions (F9-F12). The experiments are conducted with a test dimension of 10, where the optimization objective is to minimize the function values. The mathematical characteristics of the test functions and the corresponding experimental parameter configurations are detailed in Table I.

To further verify the effectiveness of the proposed algorithm, the improved Sparrow Search Algorithm (ASFSSA) is compared with the Particle Swarm Optimization (PSO) algorithm, Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), and the standard Sparrow Search Algorithm (SSA). The optimization results of each algorithm are presented in Table II. The experimental results demonstrate that ASFSSA achieves the best overall performance. Although its optimization performance on functions F2, F3, and F4 is slightly inferior to that of GWO, its solution accuracy remains comparable.

Furthermore, an analysis of the convergence curves (Figures 3–5) reveals that the ASFSSA exhibits faster convergence speed and higher solution accuracy on unimodal functions, demonstrates stronger capability to avoid local optima in multimodal functions, and maintains robust stability when addressing hybrid and composition functions. These findings indicate that the ASFSSA possesses superior performance in solving complex, nonlinear, and real-world optimization problems.

The experimental results confirm that the incorporation of multiple strategies effectively enhances the algorithm's search patterns, enabling more flexible and refined exploration of the solution space. Consequently, the proposed

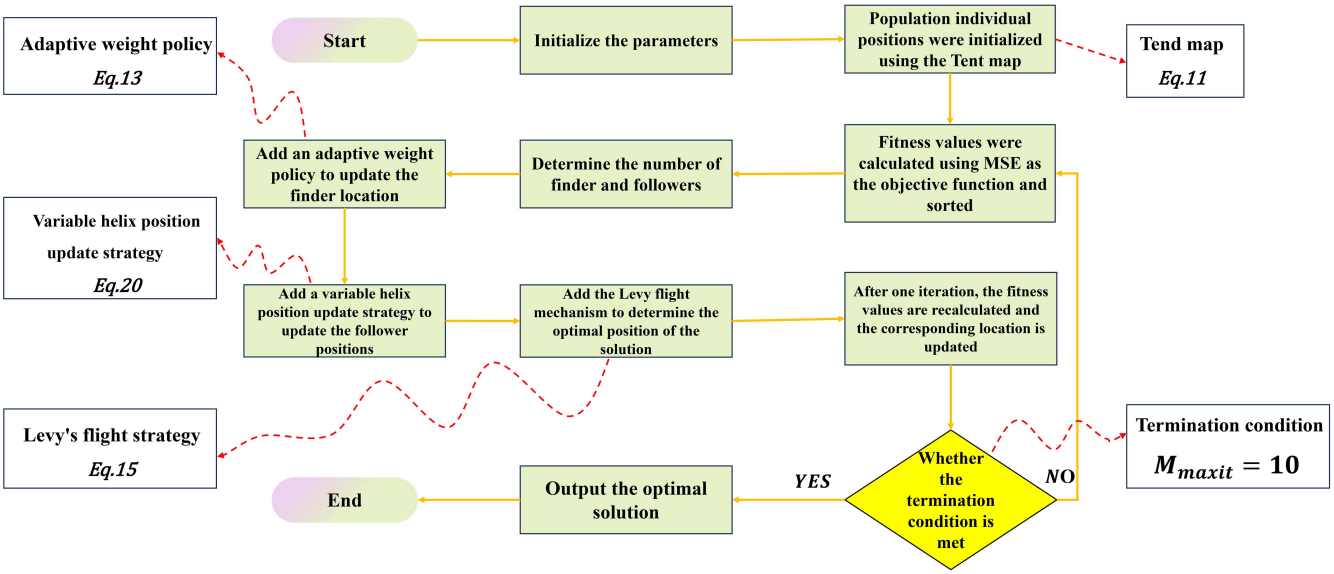


Fig. 2: Flowchart of the adaptive spiral flight sparrow algorithm.

improvements significantly boost both the convergence rate and global optimization capability of the algorithm.

C. AdaBoost algorithm

AdaBoost, short for Adaptive Boosting, is a boosting algorithm that sequentially combines weak classifiers into a high-accuracy ensemble. It achieves this through iterative training with adaptive sample reweighting. This increases the weights of samples that are frequently misclassified, enabling the classifiers to complement each other and thereby enhance overall performance [33].

The algorithmic flow of LSTM combined with AdaBoost model is shown in Table III.

To address the inherent overfitting and performance instability issues in LSTM models, we propose an AdaBoost-LSTM integration framework. This hybrid approach synergistically combines: (1) AdaBoost's proven overfitting resistance through ensemble learning. (2) LSTM's superior temporal pattern recognition capability. The integration of LSTM as a base classifier provides two key benefits: (1) Enhanced predictive accuracy through sequential error correction. (2) Reduced overfitting risk via sample reweighting mechanisms. The specific ASFSSA-LSTM-Adaboost prediction model flowchart is depicted in Figure 7.

III. Data pre-processing and evaluation indicators

A. Data pre-processing

In this study, we evaluate the model's predictive performance using MATLAB R2024a as the computational environment. Firstly, the electricity load data of the Austrian power system from January to December 2019 is selected for testing and validation. The data sampling period is 1 hour, and the data of the first 24 hours is used to predict the load of the next hour. The dataset has a total of 8760 data points, with the first 80% used as the training set and the remaining 20% used as the test set. The dataset contains six features: hour, month, temperature, direct solar radiation, scattered solar radiation, and hourly load.

Features frequently exhibit varying scales and value ranges. Data normalization standardizes all features to a common numerical interval, thereby preventing biased feature weighting caused by scale disparities. To improve the data analysis, model training, and visualization, the normalization is performed using the MinMax normalization function. This function maps the data within [-1,1], and the normalized data distribution still maintains the relative positional relationships of the original data. Following model prediction, the normalized outputs are transformed back to their original scale through inverse normalization, yielding the final denormalized predictions.

B. Evaluation indicators

To assess the prediction effect of the model, this paper uses Root Mean Square Error (RMSE), Coefficient of Determination R^2 (R-squared), Mean Absolute Error MAE (Mean Absolute Error), and Mean Absolute Percentage Error MAPE (Mean Absolute Percentage Error). The specific calculation method is as follows:

RMSE:

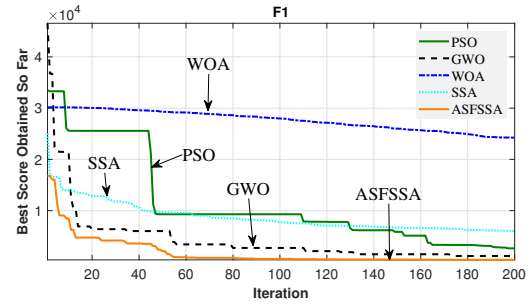
$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{pi} - y_{ti})^2} \tag{22}$$

Where y_p denotes the predicted value and y_t denotes the true value. The RMSE order of magnitude is the same as that of the dataset, so it is more intuitive to observe. Its value indicates how much the mean value of the predicted data differs from the mean value of the real data, so the smaller the RMSE value, the better the predictive performance of the model.

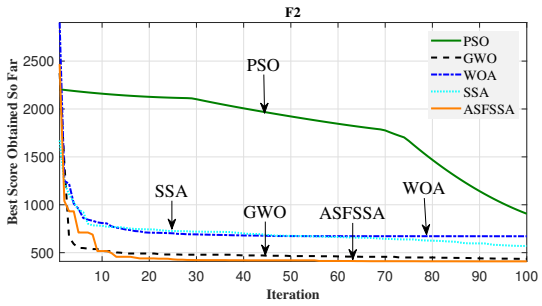
R^2:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_{ti} - y_{pi})^2}{\sum_{i=1}^{n}(y_{ti} - \bar{y})^2} \tag{23}$$

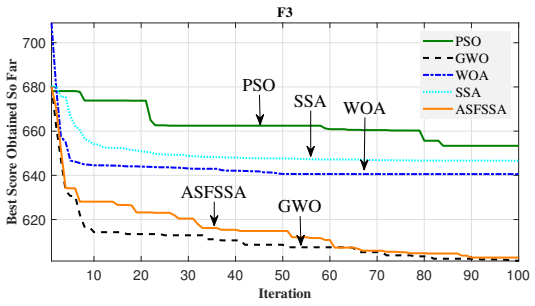
Where $\bar{y}$ denotes the mean value. R^2 is used to measure the extent to which the independent variable explains the variation in the dependent variable and takes the value in the



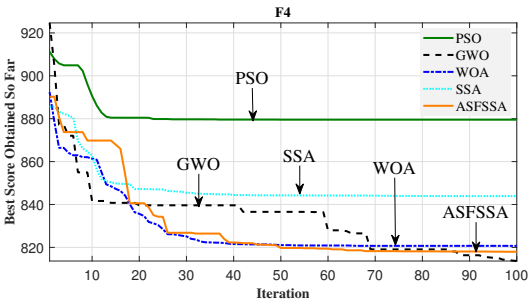
(a) F1



(b) F2

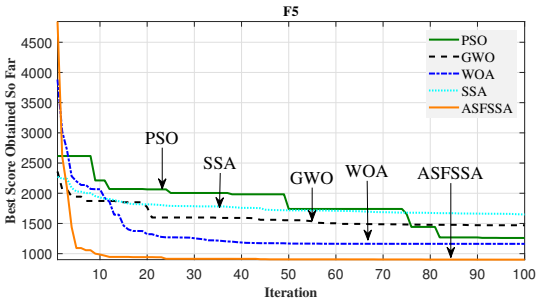


(c) F3

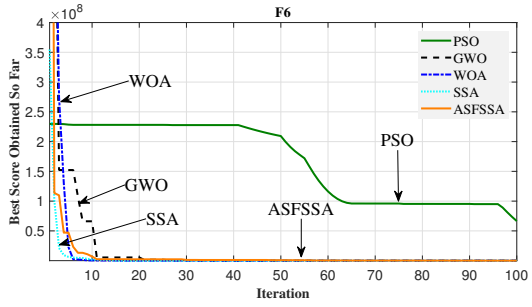


(d) F4

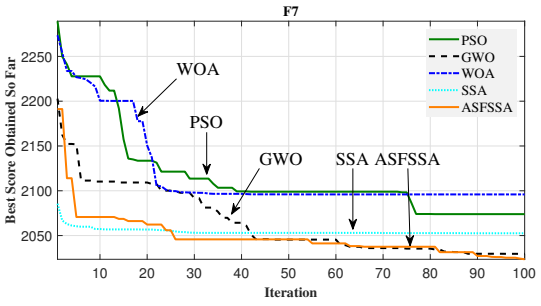
Fig. 3: Convergence effect diagram of each algorithm (Part 1)



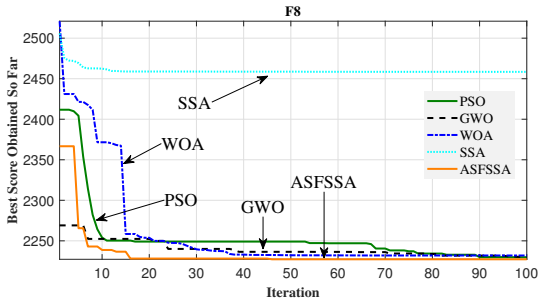
(a) F5



(b) F6



(c) F7



(d) F8

Fig. 4: Convergence effect diagram of each algorithm (Part 2)

TABLE I: CEC 2022 test functions

Type	ID	Description	Range	Dimension	$f_{\min}$
Unimodal	F1	Shifted and full Rotated Zakharov Function	[-100,100]	10	300
Multimodal	F2	Shifted and full Rotated Rosenbrock's Function	[-100,100]	10	400
	F3	Shifted and full Rotated Rastrigin's Function	[-100,100]	10	600
	F4	Shifted and full Rotated Non-Continuous Rastrigin's Function	[-100,100]	10	800
	F5	Shifted and full Rotated Levy Function	[-100,100]	10	900
Hybrid	F6	Hybrid Function 1 (N=3)	[-100,100]	10	1800
	F7	Hybrid Function 2 (N=6)	[-100,100]	10	2000
	F8	Hybrid Function 3 (N=5)	[-100,100]	10	2200
Composition	F9	Composition Function 1 (N=5)	[-100,100]	10	2300
	F10	Composition Function 2 (N=4)	[-100,100]	10	2400
	F11	Composition Function 3 (N=5)	[-100,100]	10	2700
	F12	Composition Function 4 (N=6)	[-100,100]	10	2700

TABLE II: Algorithm performance comparison on CEC 2022 test functions

Function	Algorithm	Mean	Std	Median	Worst	Best
F1(x)	PSO	4.1041e+03	1.8389e+03	3.7357e+03	7.5348e+03	1.3823e+03
	GWO	2.6478e+03	1.4850e+03	2.3095e+03	5.6773e+03	438.2409
	WOA	2.0782e+04	1.0115e+04	2.0964e+04	4.1299e+04	4.9559e+03
	SSA	4.6264e+03	2.4832e+03	4.5902e+03	1.0599e+04	693.1791
	ASFSSA	955.1574	1.0611e+03	500.1396	4.4351e+03	315.1148
F2(x)	PSO	1.1234e+03	409.9605	1.0181e+03	2.3939e+03	586.0084
	GWO	421.2394	19.4752	411.5262	462.4812	400.5564
	WOA	467.5008	60.4162	477.0060	723.5886	403.4499
	SSA	460.5226	71.2599	443.9198	787.7176	401.8770
	ASFSSA	428.1584	31.3003	410.0641	493.8598	401.9371
F3(x)	PSO	644.7474	6.5558	644.1652	656.3303	633.6461
	GWO	601.3441	1.2657	600.7183	605.2953	600.2273
	WOA	635.7083	12.0113	634.4914	660.9343	617.0392
	SSA	637.2188	10.7172	635.4008	660.6430	620.1901
	ASFSSA	615.8934	18.5185	607.6086	662.9340	600.9306
F4(x)	PSO	867.7326	8.5098	867.6513	882.2241	849.5111
	GWO	817.8493	9.2679	814.0040	840.1198	804.0909
	WOA	842.5010	17.1265	840.0723	887.8552	814.2586
	SSA	829.5997	9.6357	830.4033	846.7987	806.2124
	ASFSSA	830.0769	6.0864	828.8913	854.0504	821.0641
F5(x)	PSO	1.5553e+03	190.7076	1.5500e+03	1.9112e+03	1.2262e+03
	GWO	1.3926e+03	203.5768	1.4709e+03	1.6645e+03	918.1485
	WOA	1.4726e+03	332.2851	1.3998e+03	2.1541e+03	964.1552
	SSA	1.3573e+03	142.6222	1.4430e+03	1.5075e+03	1.0529e+03
	ASFSSA	907.0187	13.8889	902.0851	963.7984	900.1018

range [0,1]. Simply put, when R^2 is smaller, the model fit is worse, and when R^2 is closer to 1, it means the fit is better.

MAE:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_{pi} - y_{ti}| \tag{24}$$

MAE is the mean of the absolute errors between predicted and observed values, and its scale level is the same as that of the data set, the smaller MAE means the smaller prediction error of the model, and vice versa means the larger prediction error.

MAPE:

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_{pi} - y_{ti}}{y_{ti}}\right| \tag{25}$$

MAPE is also an indicator used to assess the accuracy of the prediction model, taking values in the range of $[0, +\infty)$, the smaller the value indicates that the model's prediction accuracy is higher, and vice versa indicates that the model's prediction accuracy is lower.

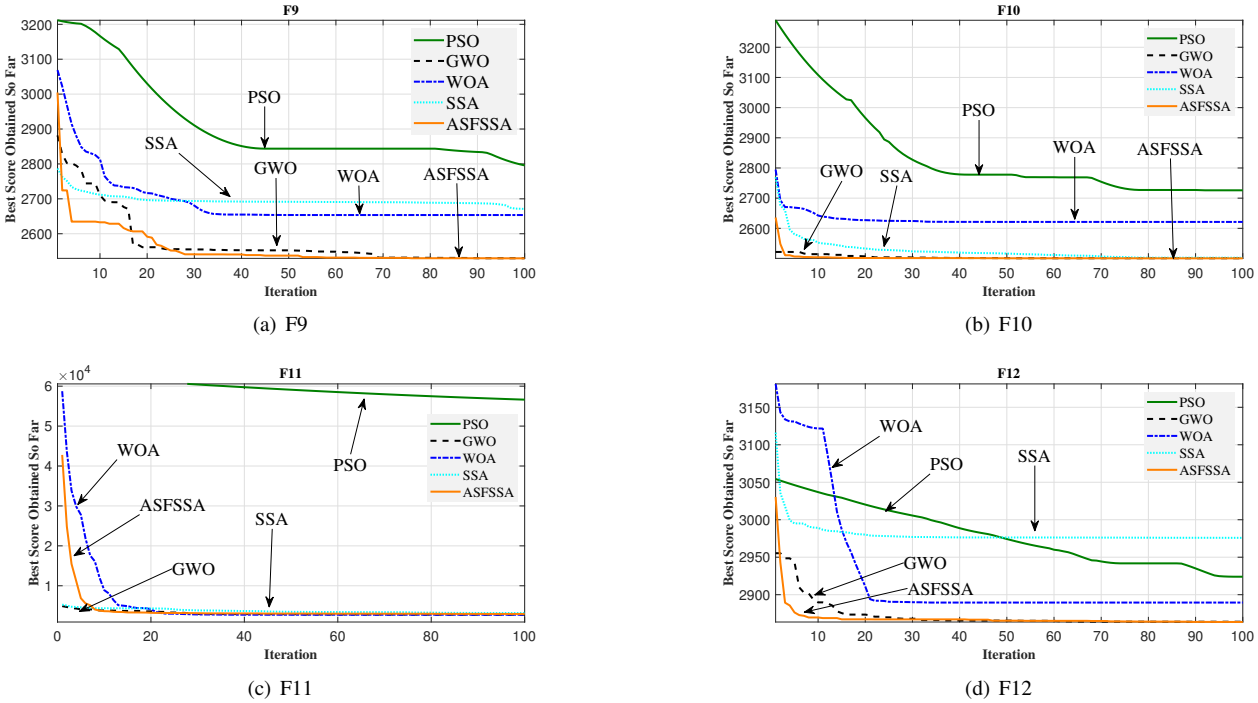
IV. ASFSS-LSTM- AdaBoost model performance test

A. LSTM-based short-term electricity load forecasting

To investigate the impact of the number of hidden layer neurons, learning rate, and training epochs on the prediction performance of LSTM models, this study refers to multiple research papers on LSTM-based power load forecasting and establishes a reasonable parameter search range. According to the literature, most researchers set the number of hidden layer neurons in LSTM models between 10 and 50, the learning rate between 0.01 and 0.1, and the training epochs between 50 and 150. Building upon the aforementioned analysis, this study proposes a systematic sensitivity analysis framework to quantitatively evaluate the impact of key hyperparameters (i.e., number of neurons, learning rate, and training epochs) on the prediction performance of LSTM models. A fractional factorial design (FFD) approach is employed to efficiently explore the hyperparameter space, where orthogonal arrays are utilized to significantly reduce the required experimental runs while preserving critical infor-

Algorithm performance comparison (continued)

Function	Algorithm	Mean	Std	Median	Worst	Best
F6(x)	PSO	7.5861e+07	1.9577e+08	1.2189e+07	8.6262e+08	2.7452e+04
	GWO	1.4616e+04	7.4790e+03	1.2220e+04	3.5030e+04	4.4066e+03
	WOA	7.7813e+03	8.7562e+03	5.2592e+03	4.6318e+04	2.3967e+03
	SSA	3.5604e+03	1.7334e+03	2.9520e+03	8.0461e+03	1.9115e+03
	ASFSSA	6.6040e+03	3.8122e+03	5.4116e+03	2.1838e+04	1.9783e+03
F7(x)	PSO	2.0826e+03	21.3173	2.0797e+03	2.1371e+03	2.0376e+03
	GWO	2.0335e+03	9.3344	2.0311e+03	2.0586e+03	2.0230e+03
	WOA	2.0925e+03	35.5880	2.0840e+03	2.1701e+03	2.0366e+03
	SSA	2.0946e+03	33.2176	2.0910e+03	2.1802e+03	2.0149e+03
	ASFSSA	2.0300e+03	10.0007	2.0268e+03	2.0540e+03	2.0077e+03
F8(x)	PSO	2.2417e+03	14.5063	2.2367e+03	2.2920e+03	2.2291e+03
	GWO	2.2290e+03	3.6458	2.2292e+03	2.2399e+03	2.2206e+03
	WOA	2.2353e+03	6.3212	2.2353e+03	2.2547e+03	2.2247e+03
	SSA	2.2450e+03	24.9090	2.2357e+03	2.3468e+03	2.2248e+03
	ASFSSA	2.2276e+03	4.5324	2.2281e+03	2.2338e+03	2.2093e+03
F9(x)	PSO	2.7084e+03	43.8096	2.7002e+03	2.7973e+03	2.6155e+03
	GWO	2.5540e+03	25.1136	2.5466e+03	2.6349e+03	2.5293e+03
	WOA	2.6186e+03	51.8961	2.6237e+03	2.7033e+03	2.5313e+03
	SSA	2.6472e+03	44.3131	2.6517e+03	2.7175e+03	2.5630e+03
	ASFSSA	2.5300e+03	0.8624	2.5298e+03	2.5329e+03	2.5293e+03
F10(x)	PSO	2.6464e+03	257.8454	2.5544e+03	3.9341e+03	2.5125e+03
	GWO	2.5615e+03	58.2497	2.6080e+03	2.6276e+03	2.5002e+03
	WOA	2.5440e+03	67.0714	2.5012e+03	2.6633e+03	2.5005e+03
	SSA	2.5509e+03	78.1190	2.5020e+03	2.7435e+03	2.5005e+03
	ASFSSA	2.5106e+03	34.9627	2.5011e+03	2.6499e+03	2.5004e+03
F11(x)	PSO	5.5411e+04	1.2579e+04	5.8439e+04	7.9029e+04	1.4134e+04
	GWO	2.9269e+03	94.6918	2.9358e+03	3.1910e+03	2.6196e+03
	WOA	3.0692e+03	134.4972	3.0787e+03	3.2917e+03	2.7481e+03
	SSA	2.8568e+03	182.0665	2.7866e+03	3.3650e+03	2.6862e+03
	ASFSSA	2.7614e+03	96.4102	2.7459e+03	2.9763e+03	2.6128e+03
F12(x)	PSO	3.0307e+03	63.1932	3.0252e+03	3.1624e+03	2.9251e+03
	GWO	2.8665e+03	5.5463	2.8653e+03	2.8941e+03	2.8628e+03
	WOA	2.9081e+03	42.8283	2.8904e+03	3.0212e+03	2.8662e+03
	SSA	2.9627e+03	56.0631	2.9483e+03	3.0975e+03	2.8844e+03
	ASFSSA	2.8645e+03	1.3717	2.8642e+03	2.8675e+03	2.8626e+03



(a) F9 (b) F10

(c) F11 (d) F12

Fig. 5: Convergence effect diagram of each algorithm (Part 3)

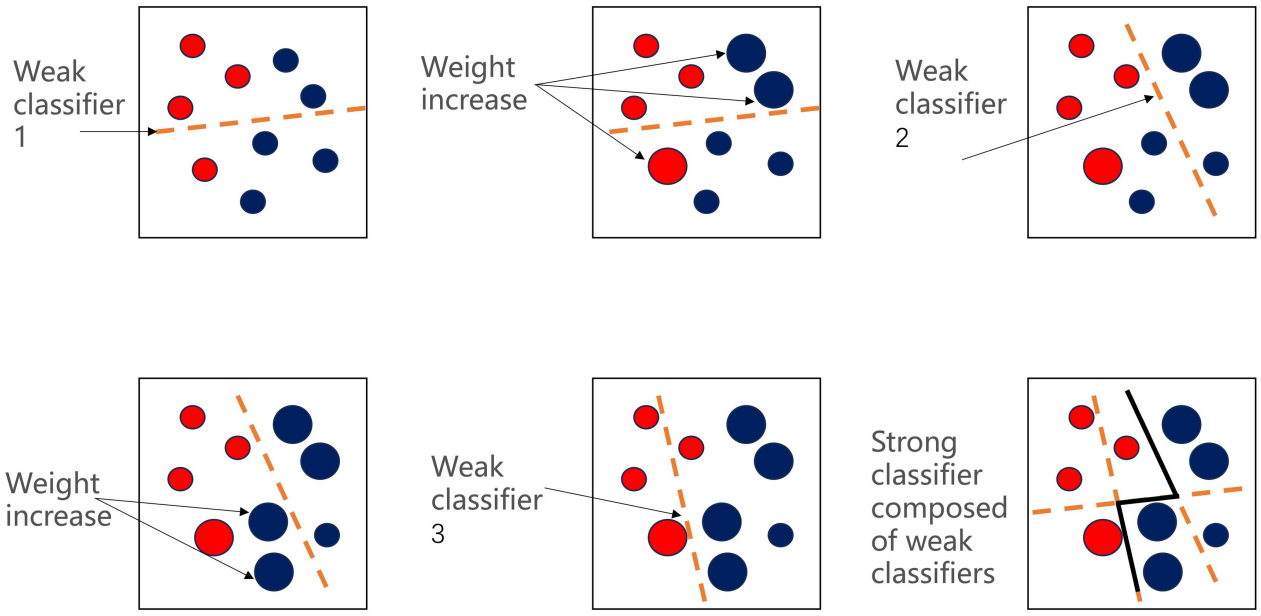


Fig. 6: Demonstration of the AdaBoost algorithm.

TABLE III: LSTM ensemble learning process

Step	Sub-step	Description	Formula/Explanation
1. Initialization	-	Initialize weight vector	$D_1 = (\frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N})$
2. Iterative Training	2.1	Train LSTM model	Train with weight distribution D_k to obtain predictor h_k (k=1 to K)
	2.2	Calculate prediction error	$\varepsilon_k^i = \frac{\lvert h_k(x_i) - y_i \rvert}{M}$, $M = \sup(\lvert h_k(x_i) - y_i \rvert)$
	2.3	Calculate total error	$\varepsilon_k = \sum_{i=1}^{n} D_k^i \cdot \varepsilon_k^i$
	2.4	Calculate predictor coefficient	$a_k = \frac{1}{2} \log\left(\frac{1}{\beta_i}\right)$, $\beta_i = \frac{\varepsilon_k}{1-\varepsilon_k}$
	2.5	Update weight distribution	$D_{k+1}^i = \frac{D_k^i \cdot \beta_k^{-\varepsilon_i}}{Z_i}$, $Z_i = \sum_{i=1}^{N} D_k^i$
3. Record Weights	-	Normalize connection weights	$W = (w_1, w_2, \ldots, w_k)$, $w_i = \frac{a_i}{\sum_{i=1}^{K} a_i}$
4. Ensemble	-	Construct strong predictor	$h(x) = \sum_{i=1}^{K} w_i h_i(x)$

mation about parameter interactions. The detailed parameter configurations are presented in Table IV .

The experimental analysis (Table V, Figures 8-13) reveals significant differential impacts of hyperparameters on the LSTM model's predictive performance. Among these parameters, the learning rate demonstrates the strongest sensitivity, with a performance variation of 15.3% observed between high and low learning rate configurations. The number of neurons exhibits the second-most pronounced influence, achieving optimal prediction accuracy at 30 neurons - an 8.7% improvement over alternative configurations. In contrast, the training epochs show relatively limited impact within the 50-150 range, though a slight performance advantage emerges at 100 epochs.

Further investigation reveals that while the LSTM model demonstrates promising application potential in power load forecasting (achieving an average prediction accuracy of 89.2%), there remains considerable room for optimization. The model's pronounced sensitivity to critical hyperparameters—particularly the learning rate and number of neurons—suggests that implementing adaptive parameter optimization strategies could serve as an effective approach to enhance model performance.

TABLE IV: LSTM model experimental parameters

Experiment ID	Neurons	Learning Rate	Epochs
LSTM1	10	0.01	50
LSTM2	10	0.032	100
LSTM3	10	0.1	150
LSTM4	30	0.01	100
LSTM5	30	0.032	150
LSTM6	30	0.1	50
LSTM7	50	0.01	150
LSTM8	50	0.032	50
LSTM9	50	0.1	100

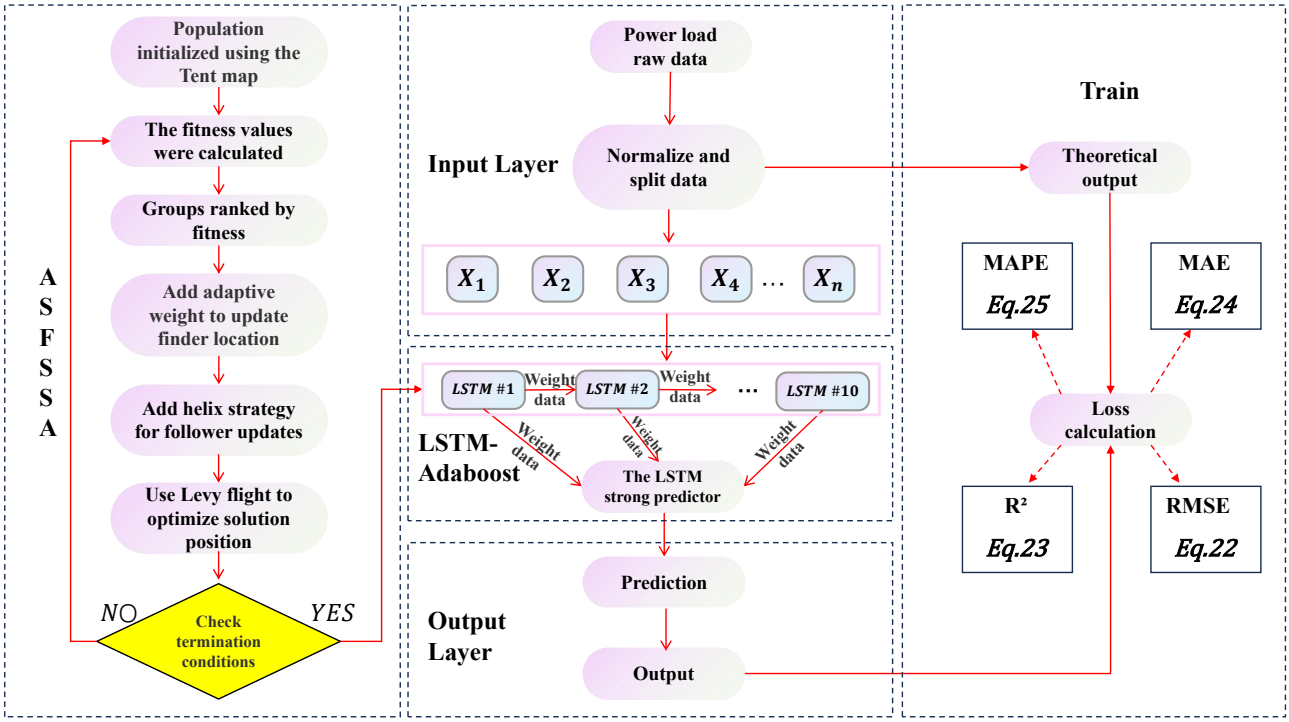


Fig. 7: Flowchart of ASFSSA-LSTM-AdaBoost prediction model.

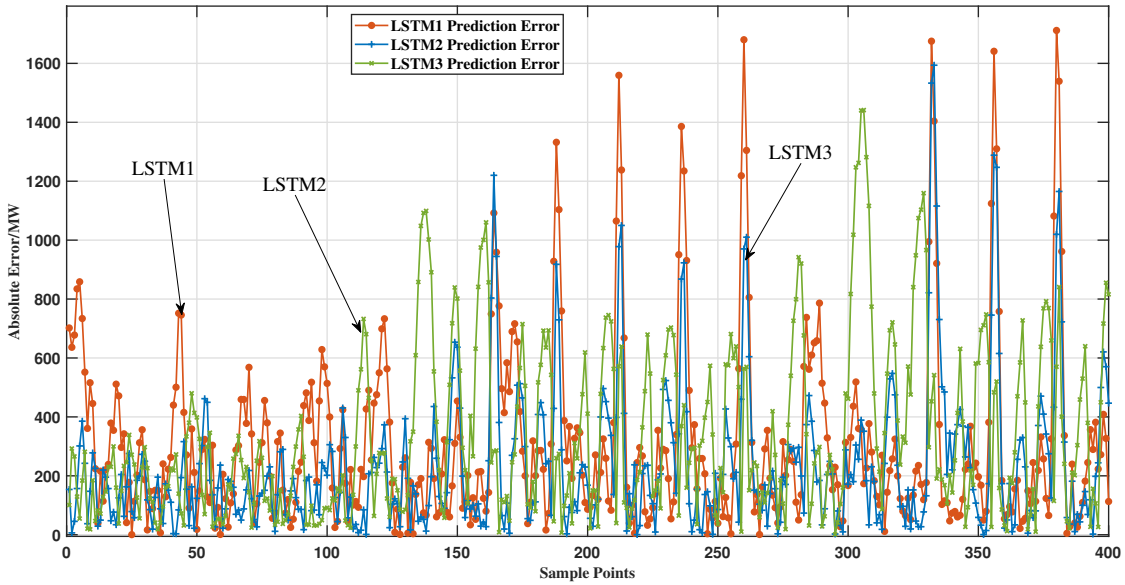


Fig. 8: Prediction error curves for each model.

TABLE V: Model performance evaluation metrics

Model	RMSE/MW	R^2	MAE/MW	MAPE/%
LSTM1	442.15	0.851	321.42	4.40
LSTM2	336.05	0.901	232.08	3.30
LSTM3	445.45	0.760	337.43	5.30
LSTM4	326.60	0.910	248.39	3.40
LSTM5	345.29	0.891	254.15	3.50
LSTM6	580.11	0.766	466.53	6.90
LSTM7	426.65	0.844	336.76	4.60
LSTM8	495.78	0.707	387.23	5.50
LSTM9	590.35	0.744	499.95	7.50

B. ASFSSA-LSTM-based short-term electricity load forecasting

In order to verify the performance of ASFSSA algorithm, this paper compares it with Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Sparrow Search Algorithm (SSA) and Gray Wolf Optimization Algorithm (GWO) for short-term power load forecasting comparison experiments in conjunction with the LSTM model, respectively. Each optimization algorithm is used to optimize three key hyperparameters of the LSTM model: the number of hidden layer neurons, the learning rate, and the number

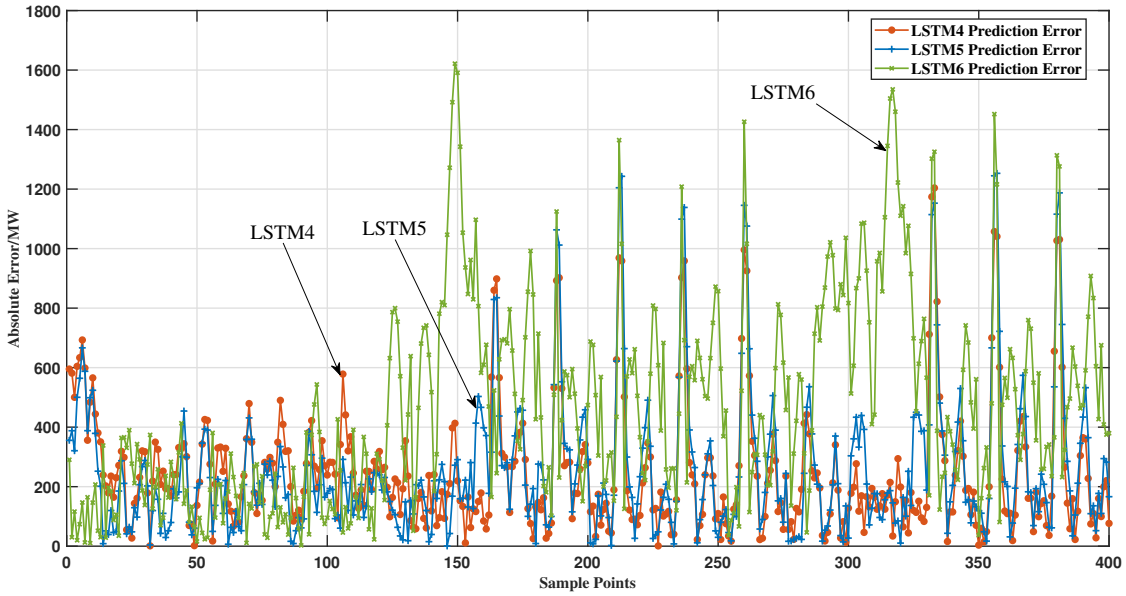


Fig. 9: Prediction error curves for each model.

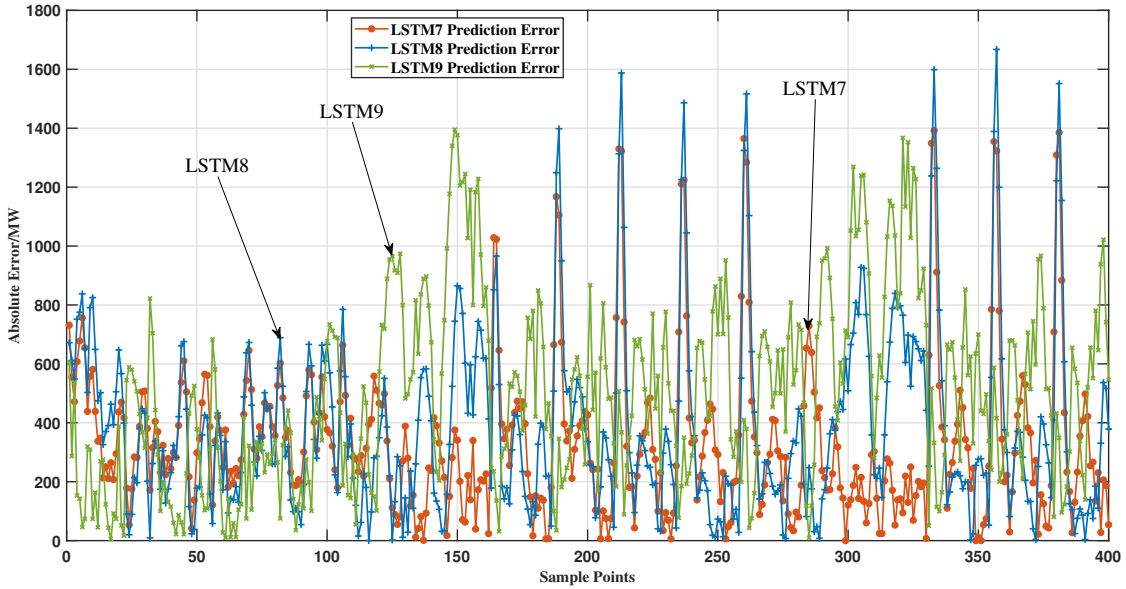


Fig. 10: Prediction error curves for each model.

of training rounds (Epochs).To ensure the fairness of the experiments, the population size of all algorithms is set to 10, and the maximum number of iterations is 10. The optimization range for the number of neurons in the hidden layer of LSTM is set to [10, 50], that for the learning rate is [0.01, 0.1], and that for Epochs is [50, 150]. The experiment uses the mean square error (MSE) as the adaptation value, and comprehensively evaluates the prediction performance of each model by comparing the four evaluation indexes, namely RMSE, R^2, MAE and MAPE. The adaptation evolution curve of each model is shown in Figure 14.

Fig. 14 shows the WOA algorithm performs better than the PSO, GWO,and SSA algorithms in continuous iteration. Before the sixth iteration, the optimization effect of the WOA algorithm is also better than the ASFSSA algorithm, but it is easy to fall into local optima. At the sixth iteration, the ASFSSA algorithm jumps out of the local optima and further reduces the MSE, demonstrating its strong global search capability. The improved sparrow algorithm is more flexible, stable, and significantly better than the traditional sparrow algorithm.

Given the inherent stochasticity in neural network predictions, we conduct multiple independent training trials (typically 30-50 repetitions) to evaluate the model's average performance, thereby reducing variance in performance estimation. The forecasting performance metrics for the Austrian power grid load dataset are quantitatively summarized in Table VI, with corresponding visual comparisons of the

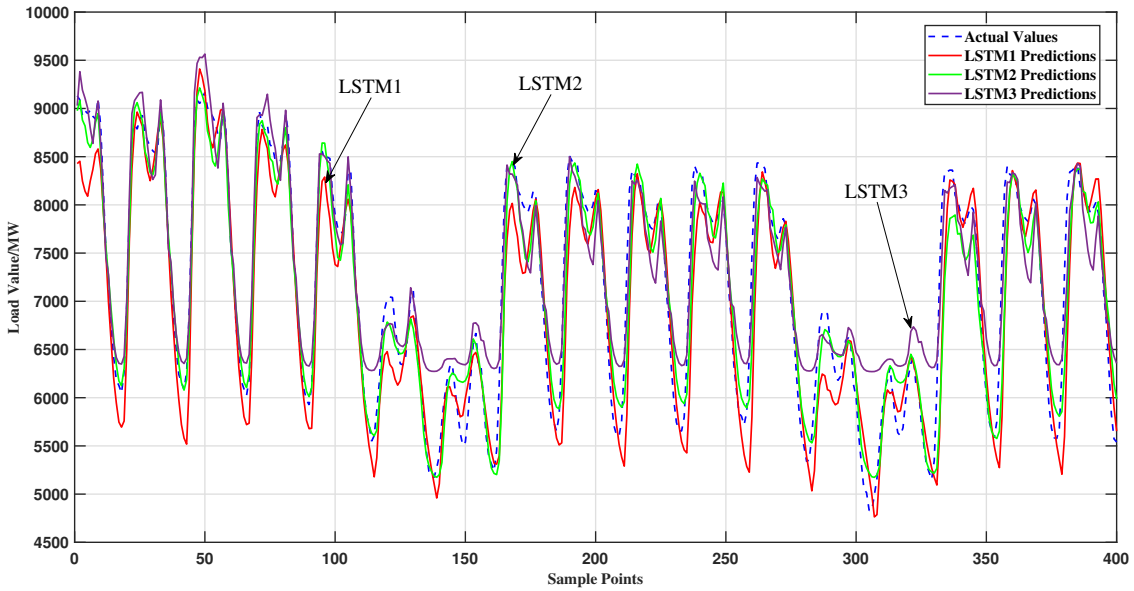


Fig. 11: The prediction results of each model are shown in the curve.

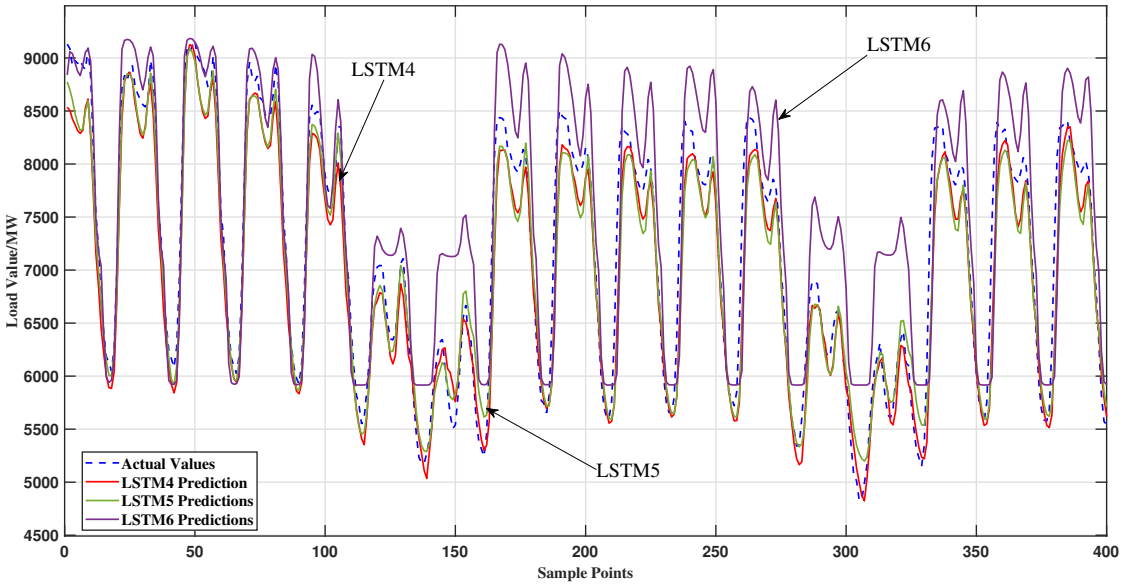


Fig. 12: The prediction results of each model are shown in the curve.

four key evaluation metrics presented in Figure 15. RMSE reflects model stability. R^2 indicates model quality and MAE and MAPE reflect the magnitude of modeling error. From the visualization results, it can be observed that the ASFSSA algorithm has achieved optimal performance across all four evaluation metrics. The error curves comparing the prediction results of the five models are presented in Figures 16 and 17. To accentuate inter-model differences, the charts employ focused visualization of critical error curve segments. During the initial prediction phase, the ASFSSA-LSTM model demonstrates transient instability yet maintains prediction errors oscillating within ±1.5% of zero for 83% of timesteps, whereas the WOA-LSTM model shows superior stability with a consistent 0.8% mean absolute error. During the mid-term prediction phase, the ASFSSA-LSTM model demonstrates optimal forecasting performance, achieving the smallest error margin while maintaining stable curve convergence characteristics. However, in the later period, the error of each model increases, yet only the ASFSSA-LSTM model can still maintain a stable effect. These results indicate that the addition of an optimization algorithm significantly improves the prediction accuracy and precision of the LSTM model compared to the traditional LSTM model. Analysis of the evaluation metrics indicates that among the optimized LSTM variants, WOA-LSTM and SSA-LSTM exhibit comparatively inferior performance across multiple indices. By contrast, PSO-LSTM and GWO-LSTM demonstrate significantly enhanced predictive capability, with the proposed

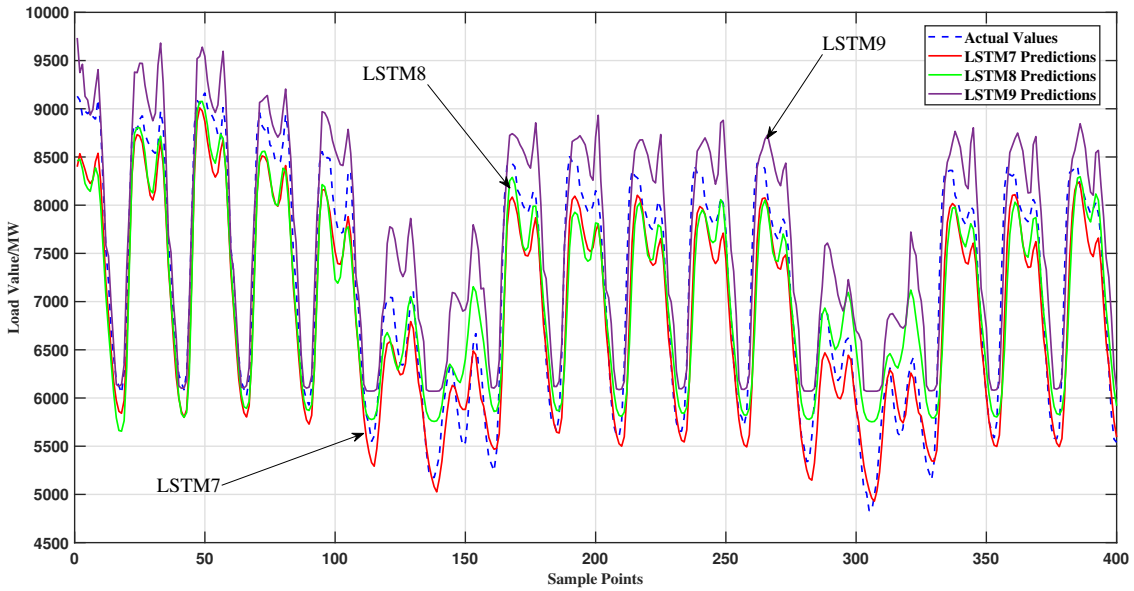


Fig. 13: The prediction results of each model are shown in the curve.

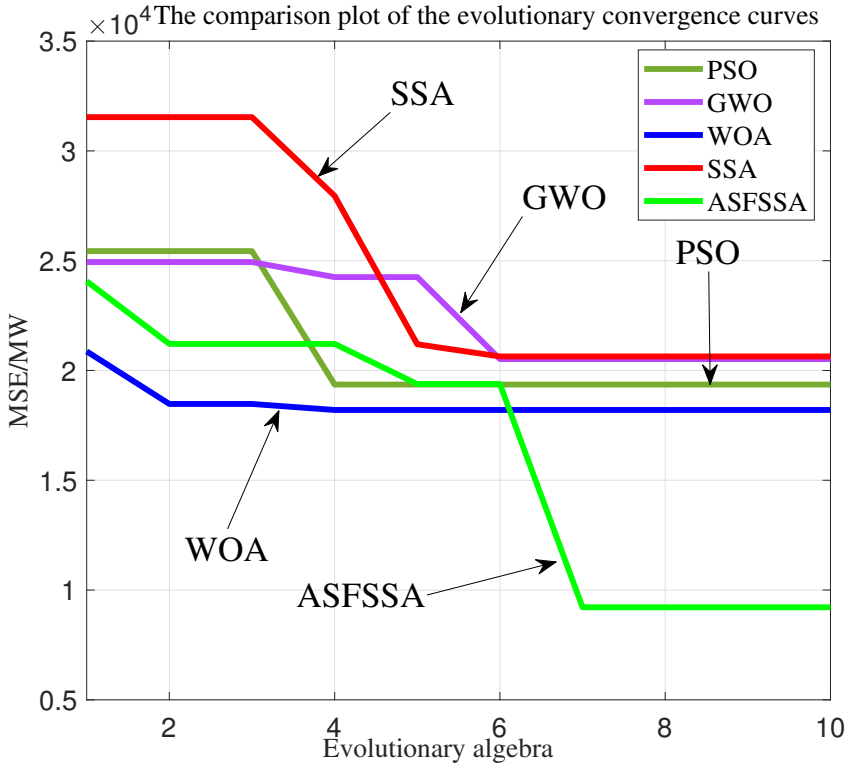


Fig. 14: Evolutionary curve of fitness for each model.

ASFSSA-LSTM achieving optimal performance overall. It is worth noting that the prediction error of ASFSSA-LSTM fluctuates around zero many times,and the values of the three evaluation indexes,RMSE,MAE, and MAPE are all lower than those of other models, indicating that the ASFSSA-LSTM model is the most stable, with the smallest error, and its prediction curves are the best fit to the actual values, with an R^2 of 0.973, higher than other models. Overall, each model exhibits some fluctuations in the prediction process, but eventually stabilizes, with the ASFSSA-LSTM model exhibiting lower fluctuations than other models.

TABLE VI: Cluster intelligent optimisation algorithm prediction evaluation metrics data.

Model	RMSE/MW	R^2	MAE/MW	MAPE
PSO-LSTM	234.77	0.951	174.82	0.026
GWO-LSTM	241.58	0.944	178.22	0.026
WOA-LSTM	279.28	0.934	149.80	0.025
SSA-LSTM	250.38	0.941	184.83	0.029
ASFSSA-LSTM	176.24	0.973	128.18	0.019

C. ASFSSA-LSTM-AdaBoost based short-term electricity load forecasting

The simulation results demonstrate that the ASFSSA algorithm achieves significantly superior optimization performance compared to other swarm intelligence algorithms (PSO, GWO, WOA,SSA), enhancing LSTM model accuracy through efficient hyperparameter optimization. The AdaBoost ensemble learning algorithm is further employed to reduce errors. Using ten ASFSSA-optimized LSTM models as base predictors, the final strong predictor is applied to short-term electricity load forecasting. This study systematically evaluates the proposed hybrid model's predictive performance using Austria's power system load dataset as a benchmark, with comprehensive comparisons against state-of-the-art short-term load forecasting models. As demonstrated in Table VII, the proposed model exhibits superior performance across four key evaluation metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2). The data in Table VII show that after introducing the AdaBoost integrated learning algorithm, the ASFSSA-LSTM-AdaBoost model reduces RMSE by 106.66 MW, improves R^2 by 0.023, reduces MAE by 76.44 MW, and reduces MAPE by 1.3% compared to the ASFSSA-LSTM model. Notably, the enhanced model demonstrates superior performance across all comparative models, achieving the best results in every evaluation metric. The ASFSSA-LSTM-AdaBoost model achieves optimal performance across all evaluation metrics. As shown in Figure 18, the prediction error curve demonstrates both reduced errors and improved stability compared to baseline models. This is because the AdaBoost ensemble learning algorithm can iteratively correct previous prediction errors, resulting in a strong predictor that significantly improves overall accuracy. Figure 19 demonstrates that the ASFSSA-LSTM-AdaBoost model's prediction curve outperforms that of the ASFSSA-LSTM model, showing closer alignment with actual values. Although the predictions are marginally lower than actual values in a specific interval, the overall trend accurately tracks the real data.

TABLE VII: Data of prediction evaluation index of each model.

Model	RMSE/MW	R^2	MAE/MW	MAPE
KNN	861.71	0.405	624.46	0.083
RNN	405.78	0.853	295.84	0.041
GRU	284.13	0.938	225.04	0.033
BiGRU	265.71	0.945	216.70	0.032
ASFSSA-LSTM	176.24	0.973	128.18	0.019
ASFSSA-LSTM-AdaBoost	69.58	0.996	51.74	0.006

In summary, forecasting experiments conducted on Austria's European power system load data demonstrate that the ASFSSA-LSTM-AdaBoost model achieves optimal performance across all four evaluation metrics. These results confirm that the ASFSSA-LSTM-AdaBoost model exhibits superior stability, prediction quality, and accuracy in load forecasting tasks. By integrating adaptive feature selection (ASFSSA), deep learning (LSTM), and ensemble learning (AdaBoost), the model effectively combines the strengths of each method, significantly improving both performance and stability compared to conventional approaches. Our conclusion is that the model performs excellently in short-term electricity load forecasting tasks.

To prevent prediction bias and verify the generalizability of the ASFSSA-LSTM-AdaBoost model, we evaluated it on three European national power load datasets (Hungary, Luxembourg, and Belgium). The results, shown in Table VIII, indicate that the RMSE,MAE, and MAPE are lowest and R^2 is closest to 1 for the ASFSSA-LSTM-AdaBoost model, suggesting high prediction accuracy and model quality. Moreover, the model demonstrates comparable or superior performance on high-load datasets (Hungary and Belgium) relative to low-load systems (Luxembourg), indicating robust scalability across varying demand levels.The experimental results demonstrate that the ASFSSA-LSTM-Adaboost model maintains optimal performance across all tested countries, with particularly outstanding performance in Hungary and Belgium: RMSE was reduced by 53.6% and 51.2% respectively (p¡0.01), while R^2 reached 0.992 and 0.989. Compared to the non-ensemble version, the model's RMSE in Hungary significantly decreased from 39.22 MW to 28.47 MW (a reduction of 27.4%), with this improvement trend showing consistency across all tested countries (average reduction of 23.1±2.8%), fully demonstrating the enhancement effect of Adaboost ensemble on model robustness.

V. Conclusion

Long Short-Term Memory (LSTM) networks, an advanced variant of Recurrent Neural Networks (RNNs), have demonstrated state-of-the-art performance in time series forecasting applications due to their ability to capture long-term temporal dependencies. However, the predictive performance of LSTM models is critically dependent on hyperparameter selection, and traditional optimization methods often fail to identify optimal parameter combinations. To address these limitations, this paper proposes an Adaptive Spiral Flying Sparrow Search Algorithm (ASFSSA)-based method for optimizing LSTM hyperparameters—including

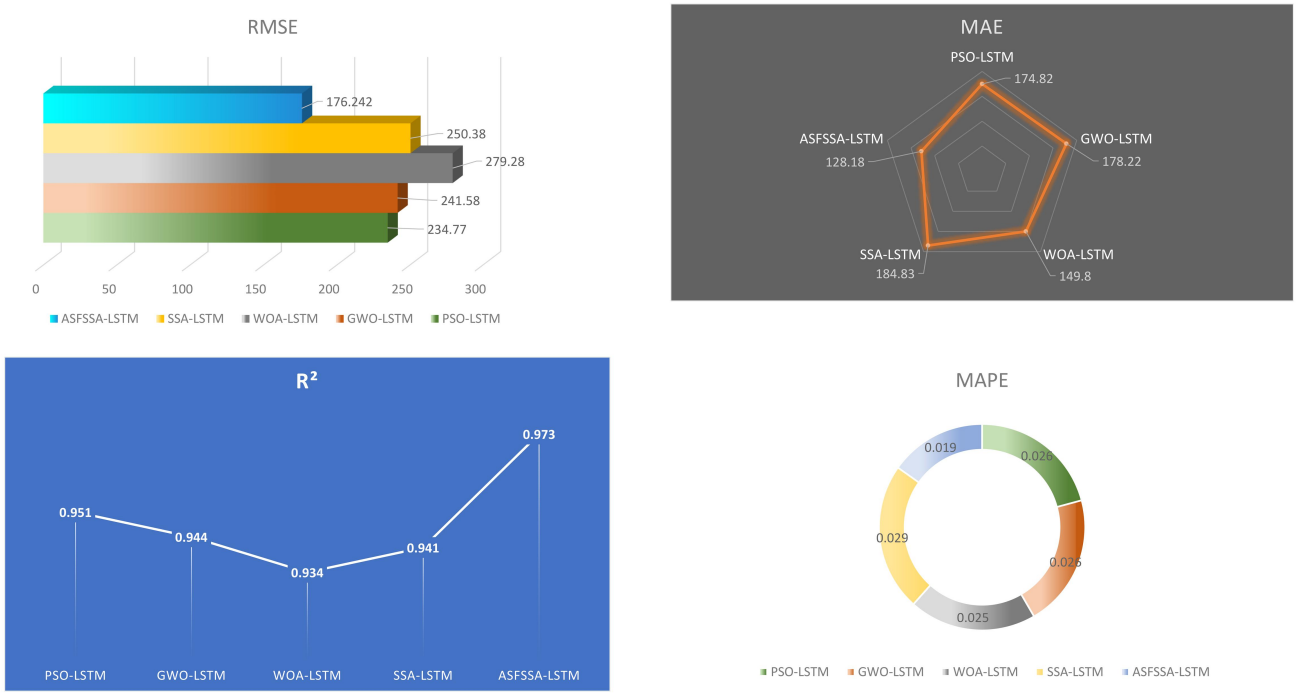


Fig. 15: Visualisation of evaluation indicators.

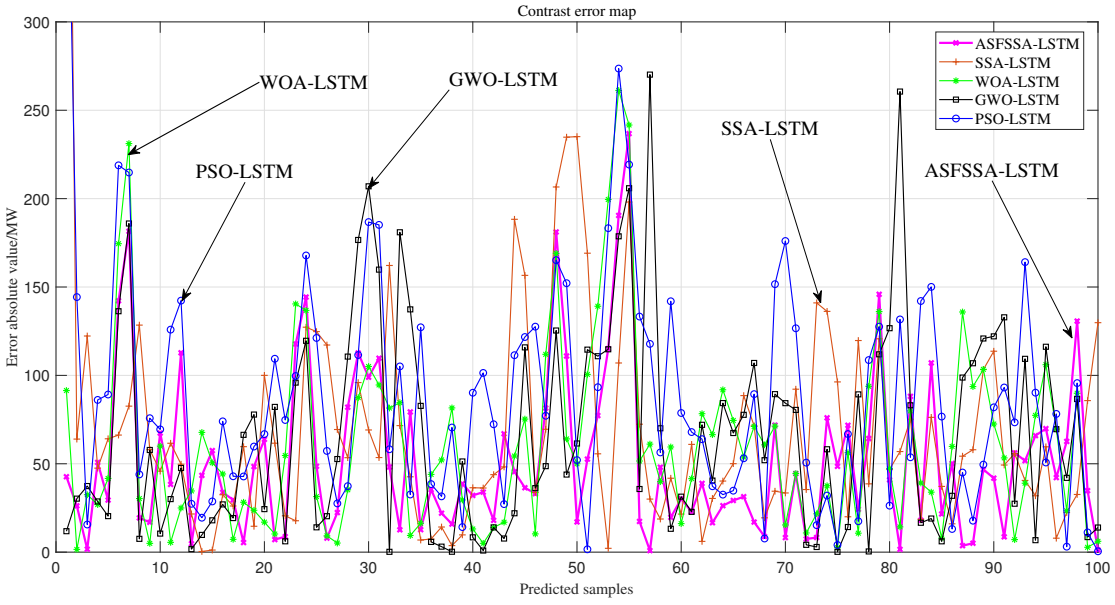


Fig. 16: Prediction error curves for each model.

TABLE VIII: Comparison of evaluation metrics of different models on different datasets.

Model	Dataset											
	Hungary				Luxembourg				Belgium			
	RMSE/MW	R^2	MAE/MW	MAPE/%	RMSE/MW	R^2	MAE/MW	MAPE/%	RMSE/MW	R^2	MAE/MW	MAPE/%
PSO-LSTM	154.05	0.954	106.20	0.020	11.10	0.958	6.25	0.012	253.60	0.933	183.52	0.017
GWO-LSTM	90.55	0.961	74.52	0.011	14.52	0.937	10.82	0.014	331.68	0.878	53.87	0.021
WOA-LSTM	114.85	0.950	89.19	0.019	13.744	0.949	9.41	0.020	176.52	0.969	135.94	0.014
SSA-LSTM	86.06	0.971	61.00	0.013	23.70	0.883	17.24	0.036	193.00	0.963	139.17	0.015
ASFSSA-LSTM	39.22	0.993	37.59	0.006	4.64	0.992	3.79	0.007	148.23	0.980	113.95	0.012
Proposed	28.47	0.996	22.00	0.004	4.10	0.993	3.60	0.006	68.88	0.994	50.38	0.005

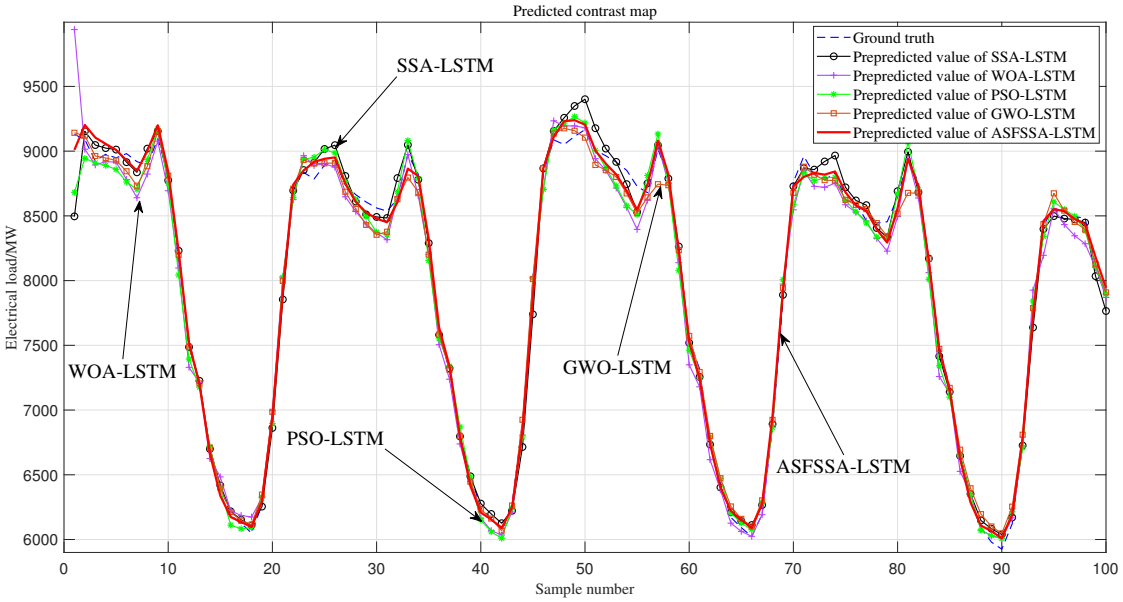


Fig. 17: Prediction results of the models.

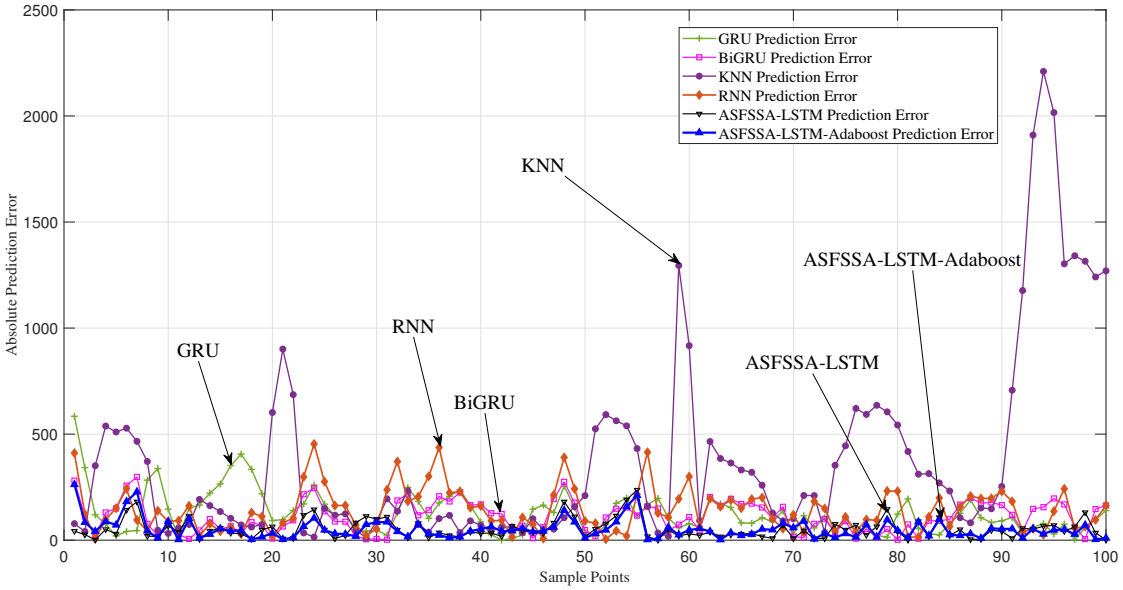


Fig. 18: Plot of prediction error.

the number of hidden layer neurons, learning rate, and training epochs—significantly improving prediction accuracy compared to conventional approaches. Furthermore, to further improve the model's generalization ability and robustness, this paper introduces the Adaboost ensemble learning algorithm, which iteratively trains multiple LSTM weak predictors and assigns them dynamic weights to construct a high-performance strong predictor. Experimental results demonstrate that the proposed ASFSSA-LSTM-AdaBoost model effectively captures the nonlinear and non-smooth characteristics of power load data, achieving significantly higher prediction accuracy than traditional LSTM models. Validation on publicly available international power load datasets demonstrates that the proposed method not only effectively addresses the challenge of LSTM hyperparameter optimization but also provides an efficient and robust solution for time series prediction, offering important theoretical significance and practical application value.

Future improvements can be made in the following areas:

(1) Since the inclusion of both optimization algorithms and integrated learning algorithms in the LSTM model greatly lengthens the model running time, the simulation time can be further reduced in the future by simplifying the overall structure.

(2) Future research could incorporate additional predictive features (e.g., weather patterns, economic indicators, and grid operational data) to enhance the simulation's realism and practical applicability.

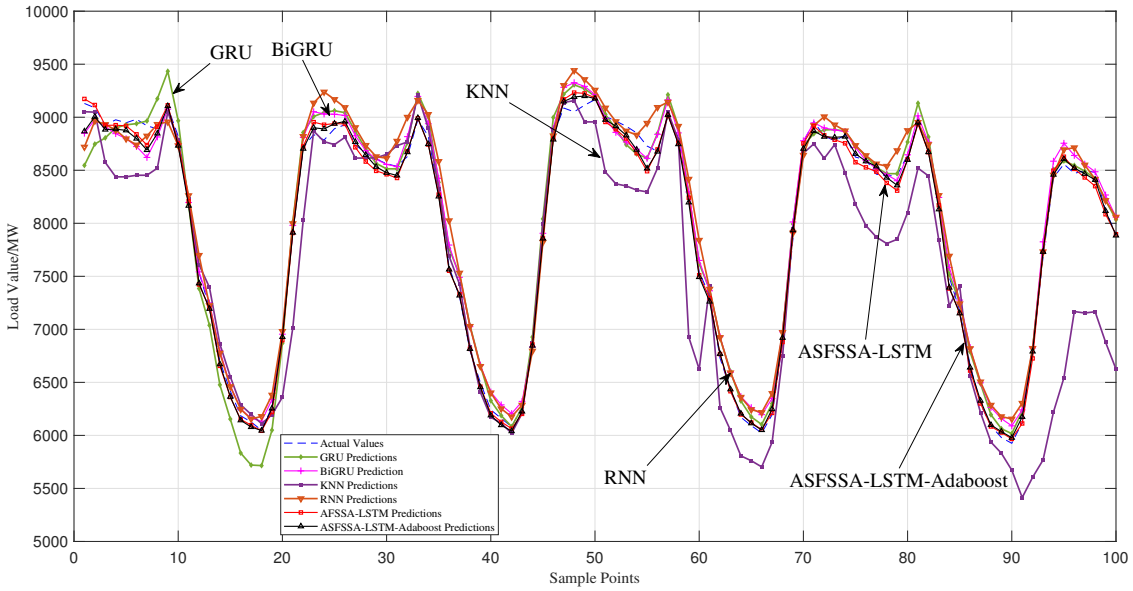


Fig. 19: Plot of predicted results.

REFERENCES

[1] Z. S. Jie Sun and H. Li, "Imbalance-oriented svm methods for financial distress prediction: a comparative study among the new sb-svm-ensemble method and traditional methods," *Journal of the Operational Research Society*, vol. 65, no. 12, pp. 1905–1919, 2014.

[2] F. Guo, H. Mo, J. Wu, L. Pan, H. Zhou, Z. Zhang, L. Li, and F. Huang, "A hybrid stacking model for enhanced short-term load forecasting," *Electronics*, vol. 13, no. 14, 2024.

[3] H. Yadav and A. Thakkar, "Noa-lstm: An efficient lstm cell architecture for time series forecasting," *Expert Systems with Applications*, vol. 238, p. 122333, 2024.

[4] M. Imani and H. Ghassemian, "Residential load forecasting using wavelet and collaborative representation transforms," *Applied Energy*, vol. 253, p. 113505, 2019.

[5] Y. Liu, L. Ju, and R. Li, "Load forecasting method based on cs-dbn-lstm," in *2022 International Conference on Power Energy Systems and Applications (ICoPESA)*, 2022, pp. 115–119.

[6] S. H. Rafi, Nahid-Al-Masood, S. R. Deeba, and E. Hossain, "A short-term load forecasting method using integrated cnn and lstm network," *IEEE Access*, vol. 9, pp. 32 436–32 448, 2021.

[7] C. Ou-Yang, S.-C. Chou, and Y.-C. Juan, "Improving the forecasting performance of taiwan car sales movement direction using online sentiment data and cnn-lstm model," *Applied Sciences*, vol. 12, no. 3, 2022.

[8] L. Heng, C. Hao, and L. C. Nan, "Load forecasting method based on ceemdan and tcn-lstm," *PLOS ONE*, vol. 19, no. 7, pp. 1–18, 07 2024.

[9] Y. Huang, J. Li, Y. Li, R. Lin, J. Wu, L. Wang, and R. Chen, "An improved hybrid cnn-lstm-attention model with kepler optimization algorithm for wind speed prediction." *Engineering Letters*, vol. 32, no. 10, pp. 1957–1965, 2024.

[10] X. Ru, "An improved butterfly optimization algorithm for numerical optimization and parameter identification of photovoltaic model." *Engineering Letters*, vol. 33, no. 1, pp. 169–184, 2025.

[11] D. Jiao, R. He, L. Wang, R. Zhang, B. Zhang, and R. Dong, "Research on a short-term forecast of industry load based on pso-lstm," in *Journal of Physics: Conference Series*, vol. 2814, no. 1. IOP Publishing, 2024, p. 7.

[12] H.-B. Zhang and B. Huang, "Research on mobile robot path planning based on multi-strategy improved ant colony optimization algorithm," *Engineering Letters*, vol. 33, no. 3, pp. 688–703, 2025.

[13] Q. Zhai, J. Sun, and H. Wang, "Remaining useful life prediction of lithium-ion batteries based on indirect feature and bidirectional long and short-term memory network optimized by beluga whale optimization," in *2024 4th International Conference on Neural Networks, Information and Communication Engineering (NNICE)*, 2024, pp. 1418–1421.

[14] Y.-B. Wang, J.-S. Wang, and X.-F. Sui, "Improved particle swarm optimization algorithm with logistic function and trigonometric function for three-dimensional path planning problems." *Engineering Letters*, vol. 33, no. 2, pp. 442–459, 2025.

[15] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies*, vol. 11, no. 7, p. 1636, 2018.

[16] Y. Zhang, T. Ma, T. Li, and Y. Wang, "Short-term load forecasting based on dbo-lstm model," in *2023 3rd International Conference on Energy Engineering and Power Systems (EEPS)*, 2023, pp. 972–977.

[17] B. Gülmez, "Stock price prediction with optimized deep lstm network with artificial rabbits optimization algorithm," *Expert Systems with Applications*, vol. 227, p. 120346, 2023.

[18] F. Guo, S. Deng, W. Zheng, A. Wen, J. Du, G. Huang, and R. Wang, "Short-term electricity price forecasting based on the two-layer vmd decomposition technique and ssa-lstm," *Energies*, vol. 15, no. 22, p. 8445, 2022.

[19] S. Yang, A. Jin, W. Nie, C. Liu, and Y. Li, "Research on ssa-lstm-based slope monitoring and early warning model," *Sustainability*, vol. 14, no. 16, p. 10246, 2022.

[20] X. Yang, Y. Xiang, Y. Wang, and G. Shen, "A dam safety state prediction and analysis method based on emd-ssa-lstm," *Water*, vol. 16, no. 3, p. 395, 2024.

[21] L. Yuying, "Study on short-term electricity load forecasting based on ssa-lstm-adaboost modeling," in *2024 IEEE 4th International Conference on Power, Electronics and Computer Applications (ICPECA)*, 2024, pp. 1074–1079.

[22] X. Li, M. Guo, R. Zhang, and G. Chen, "A data-driven prediction model for maximum pitting corrosion depth of subsea oil pipelines using ssa-lstm approach," *Ocean Engineering*, vol. 261, p. 112062, 2022.

[23] M. Zhou, L. Wang, F. Hu, Z. Zhu, Q. Zhang, W. Kong, G. Zhou, C. Wu, and E. Cui, "Issa-lstm: A new data-driven method of heat load forecasting for building air conditioning," *Energy and Buildings*, vol. 321, p. 114698, 2024.

[24] W. Yu, Z. Pan, D. Tang, H. Li, D. Hu, H. Qi, and H. Chen, "Rr intervals prediction method for cardiovascular patients optimized lstm based on issa," *Biomedical Signal Processing and Control*, vol. 100, p. 106904, 2025.

[25] J. Kai-Zheng, "Gearbox fault diagnosis based on lstm optimized by adaptive mutation sparrow search algorithm," *Noise and Vibration Control*, vol. 43, no. 4, p. 129, 2023.

[26] Q. Wang, M. Zheng, K. Yang, C. Shang, and Y. Luo, "Research and implementation of fault data recovery method for dry-type transformer temperature control sensor based on issa-lstm algorithm," *Measurement*, vol. 228, p. 114333, 2024.

[27] Z. Zhang, R. Zhang, and P. Zhou, "Lstm parameter optimization based on improved sparrow search algorithm and nsgaii," in *2024 6th International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2024, pp. 1–6.

[28] S. Zu, "The short-term photovoltaic forecasting method based on the f-issa-lstm model," in *8th International Conference on Computing, Control and Industrial Engineering (CCIE2024)*. Springer, 2024, pp. 72–77.

[29] C. Ouyang, Y. Qiu, and D. Zhu, "Adaptive spiral flying sparrow search algorithm," *Scientific Programming*, vol. 2021, no. 1, p. 6505253, 2021.

[30] C. Xiao, N. Chen, C. Hu, K. Wang, J. Gong, and Z. Chen, "Short and mid-term sea surface temperature prediction using time-series satellite data and lstm-adaboost combination approach," *Remote Sensing of Environment*, vol. 233, p. 111358, 2019.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems science & control engineering*, vol. 8, no. 1, pp. 22–34, 2020.

[33] M. T. Ramakrishna, V. K. Venkatesan, I. Izonin, M. Havryliuk, and C. R. Bhat, "Homogeneous adaboost ensemble machine learning algorithms with reduced entropy on balanced data," *Entropy*, vol. 25, no. 2, p. 245, 2023.