energies

MDPI

Article

Model NOx, SO_2 Emissions Concentration and Thermal Efficiency of CFBB Based on a Hyper-Parameter Self-Optimized Broad Learning System

Yunpeng Ma [1], Chenheng Xu [2,*], Hua Wang [3], Ran Wang [1], Shilin Liu [1] and Xiaoying Gu [1]

[1] School of Information Engineering, Tianjin University of Commerce, Beichen, Tianjin 300134, China
[2] School of Economics, Tianjin University of Commerce, Beichen, Tianjin 300134, China
[3] School of Artificial Intelligence, Hebei University of Technology, Hongqiao, Tianjin 300132, China
* Correspondence: xuchenheng@stu.tjcu.edu.cn

Abstract: At present, establishing a multidimensional characteristic model of a boiler combustion system plays an important role in realizing its dynamic optimization and real-time control, so as to achieve the purpose of reducing environmental pollution and saving coal resources. However, the complexity of the boiler combustion process makes it difficult to model it using traditional mathematical methods. In this paper, a kind of hyper-parameter self-optimized broad learning system by a sparrow search algorithm is proposed to model the NOx, SO_2 emissions concentration and thermal efficiency of a circulation fluidized bed boiler (CFBB). A broad learning system (BLS) is a novel neural network algorithm, which shows good performance in multidimensional feature learning. However, the BLS has several hyper-parameters to be set in a wide range, so that the optimal combination between hyper-parameters is difficult to determine. This paper uses a sparrow search algorithm (SSA) to select the optimal hyper-parameters combination of the broad learning system, namely as SSA-BLS. To verify the effectiveness of SSA-BLS, ten benchmark regression datasets are applied. Experimental results show that SSA-BLS obtains good regression accuracy and model stability. Additionally, the proposed SSA-BLS is applied to model the combustion process parameters of a 330 MW circulating fluidized bed boiler. Experimental results reveal that SSA-BLS can establish the accurate prediction models for thermal efficiency, NOx emission concentration and SO_2 emission concentration, separately. Altogether, SSA-BLS is an effective modelling method.

Keywords: broad learning system; sparrow search algorithm; hyper-parameter optimization; circulating fluidized bed boiler; complex system modeling

Citation: Ma, Y.; Xu, C.; Wang, H.; Wang, R.; Liu, S.; Gu, X. Model NOx, SO_2 Emissions Concentration and Thermal Efficiency of CFBB Based on a Hyper-Parameter Self-Optimized Broad Learning System. *Energies* **2022**, *15*, 7700. https://doi.org/10.3390/en15207700

Academic Editors: Jaroslaw Krzywanski, Yunfei Gao, Marcin Sosnowski, Karolina Grabowska, Dorian Skrobek, Ghulam Moeen Uddin, Anna Kulakowska, Anna Zylka and Bachil El Fil

Received: 14 September 2022
Accepted: 14 October 2022
Published: 18 October 2022

1. Introduction

Nowadays, the heat and electricity we use are mainly generated through power plants. During the combustion process of a station boiler, large amounts of polluting gases are produced, such as NOx, SO_2 and CO_2, that cause great harm to the human living environment. Simultaneously, a large amount of coal is consumed. Coal resources are becoming increasingly scarce; the goals of saving energy and emission reduction are imminent. The realization of dynamic multi-objective optimal control of boiler combustion process under variable loads is an effective method to reduce environmental pollution and save coal resources, and is called the boiler combustion optimization problem [1,2]. In order to solve the problem, the first priority is to establish the multi-dimensional feature model of the boiler combustion system. However, the boiler combustion system has complex characteristics with nonlinearity, strong coupling and large hysteresis, making it difficult to be modeled by traditional mathematical mechanistic methods. Zhou et al. have successively applied artificial neural networks or support vector machines (SVM) [3,4] combined with swarm intelligence optimization algorithms to model boiler combustion systems [5–10]. For example, ref. [5] combined ANN with genetic algorithms (GA), ref. [7]

combined SVM and a meta-genetic algorithm (MGA), and ref. [9] combined SVM and ant colony optimization (ACO) [11]. These research results showed that good prediction results could be obtained via applying artificial neural networks and support vector machines. Li and Ma et al. applied various improved extreme learning machine (ELM) models [12–14] to establish prediction models for boiler thermal efficiency, NOx emission concentration or SO_2 emission concentration [15–21]. For example, Li et al. proposed the fast-learning networks (FNN) by connecting the input and output layers of ELM and implemented the fast modeling of boiler combustion systems; Ma et al. proposed an improved online sequential ELM and implemented the real-time modeling of boiler combustion systems. However, it was proven that in traditional ANN and ELM there exists an over-fitting problem when solving small sample data regression problems. In addition, the model computation speed of SVM is slow when solving the large sample data regression problem. This paper firstly uses a newly neural network, called broad learning system [22], to solve the modeling problem of boiler combustion systems.

Broad learning system (BLS), as a new neural network algorithm proposed in 2018, has greater advantages in multi-dimensional feature learning and computing time compared to other deep learning algorithms, such as deep belief network (DBN) [23], deep boltzmann machine (DBM) [24] and convolutional neural network (CNN) [25]. Chen et al. [26] proposed several variants of BLS to solve regression problems, and experimental results showed that BLS variants had better performance than other state-of-the-art methods, such as conventional BLS, ELM and SVM. BLS has been researched and applied in many fields in recent years [27–34]. For example, Shuang and Chen [33] combined fuzzy system with BLS, proposed a new neuro-fuzzy algorithm, and applied it to solve regression and classification problems. Zhao et al. [34] extended BLS using a stream regularization framework and proposed a new algorithm for semi-supervised learning to solve complex data classification problems. However, the hyper-parameters of BLS could seriously affect its model performance. If the hyper-parameters are set too large, BLS encounters an over-fitting problem and spends more computation time. If the hyper-parameters are set too small, the generalization ability of BLS is weakened. BLS has more hyper-parameters, and every hyper-parameter needs to be set in a wide range, so the optimal combination of several hyper-parameters is difficult to determine by using traditional methods. It is of great research value to design a method to optimize the hyper-parameters of BLS to ensure its good model performance. Nacef et al. [35] leverages deep learning and network optimization techniques to solve various network configuration and scheduling problems, enabling fast self-optimization and the lifecycle management of networks, and demonstrating the great role of optimization techniques in saving runtime and reducing computational costs. In addition, swarm intelligence optimization algorithms can provide substantial benefits in reducing computational effort and improving system performance without a priori knowledge of the system parameters [36]. To address the above-mentioned problem, this paper proposes a kind of hyper-parameter self-optimized broad learning system, namely SSA-BLS. The proposed method mainly introduces the optimization mechanism of sparrow search algorithm [37] to determine the optimal hyper-parameter combination of BLS through three different behaviors of the sparrow population during foraging, i.e., sparrows as explorers provide search directions and regions for the optimal hyper-parameter combinations, sparrows as followers search through the guidance of explorers, and sparrows as vigilantes rely on anti-predation strategy to avoid hyper-parameter combinations from falling into local optima. The sparrow search algorithm (SSA) is a new swarm intelligence optimization algorithm proposed in 2020. Compared with particle swarm optimization (PSO) [38,39], gravitational search algorithm (GSA) [40] and grey wolf optimization (GWO) [41], the SSA had better computation efficiency. This paper combines SSA with BLS to automatically adjust the hyper-parameters and obtain the optimal hyper-parameter combination. In SSA-BLS, a mechanism to achieve automatic optimization of hyperparameters with the objective of minimizing the average root-mean-square-error (RMSE) of the testing set after ten-fold cross-validation [42] is proposed in order to obtain better model accuracy and model stability.

In order to verify the effectiveness of SSA-BLS, it was applied to ten benchmark regression datasets. Compared with BLS, RELM [43] and KELM [44,45], SSA-BLS can achieve the best model accuracy and model stability whose hyper-parameters are determined by using the nested cross-validation method [46].

Simultaneously, the proposed SSA-BLS was applied to establish the prediction comprehensive model of thermal efficiency, NOx emission concentration and SO_2 emission concentration. Compared with conventional BLS, the proposed SSA-BLS has better model accuracy and stronger stability. The experimental results reveal that the model accuracy can reach 10−2-10−3 by SSA-BLS.

The contributions of this paper are summarized as follows:

(1) A novel optimized BLS is proposed. SSA is firstly used to optimize the hyper-parameters of BLS, which can determine the optimal hyper-parameter combination.
(2) The proposed SSA-BLS is used to solve the regression problem of ten benchmark datasets.
(3) The proposed SSA-BLS and traditional BLS are firstly applied to establish the prediction conventional model of one circulation fluidized bed boiler combustion system.

The structure of this paper is as follows: basic knowledge and related works are given in Section 2; the proposed SSA-BLS is given in Section 3; Section 4 shows the performance evaluation of the SSA-BLS; Section 5 addresses the real-world modelling problem; the conclusion of this paper is in Section 6.

2. Basic Knowledge and Related Works

2.1. Broad Learning System

Broad learning system (BLS), as a novel artificial neural network algorithm, is capable of replacing deep architecture. It adds a dynamic stepwise update mechanism and a sparse self-coding algorithm to the random vector functional-link neural network (RVFLNN) [47–49], which greatly improves the model computing efficiency.

As opposed to RVFLN, BLS replaces the input layer with the mapping layer. The mapping layer of BLS is obtained by sparse representation and linear transformation of the input layer data. The augmentation layer is obtained by applying a nonlinear transformation to the activation function mapping layer. BLS connects the mapping layer and the enhancement layer together with the output layer to solve the connection weight of the neural networks. The structure diagram of BLS is shown in Figure 1.

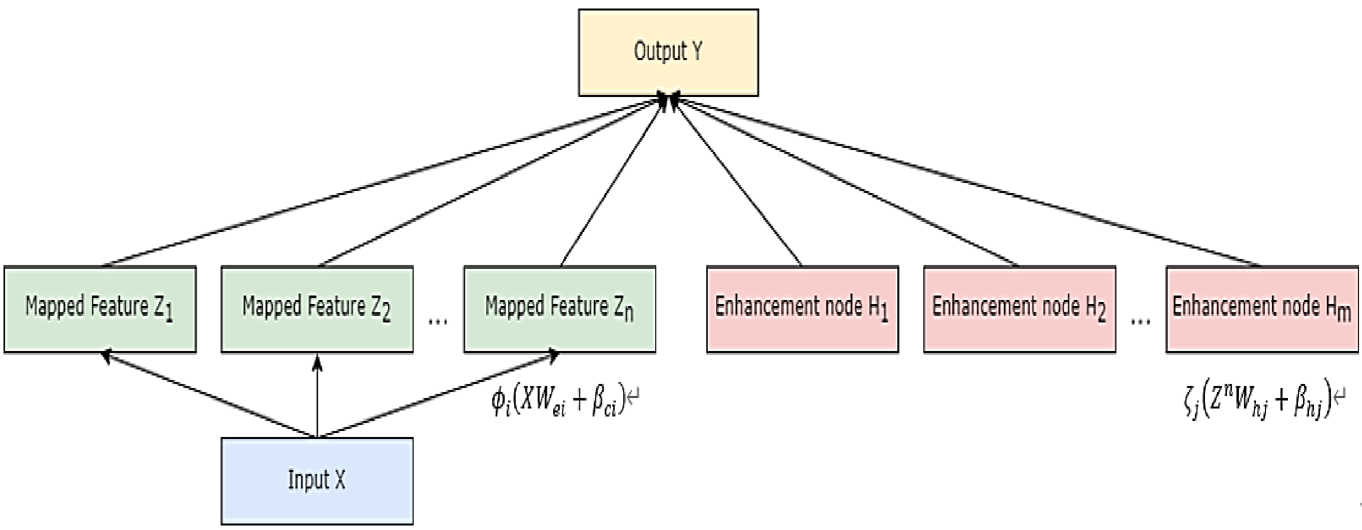

Figure 1. The structure diagram of BLS.

Where $X \in R^{N\times M}$ is the input data with sample size N and dimension M, and $Y \in R^{N\times 1}$ is the output data with sample size N and dimension 1.

Assuming that the network structure has n feature mappings and each feature mapping has k feature nodes, the expression of the ith feature mapping Z_i is shown in Equation (1).

$$Z_i = \begin{bmatrix} \phi_i(x_1 W_{i1} + \beta_{i1}) & \cdots & \phi_i(x_N W_{i1} + \beta_{i1}) \\ \vdots & \ddots & \vdots \\ \phi_i(x_1 W_{ik} + \beta_{ik}) & \cdots & \phi_i(x_N W_{ik} + \beta_{ik}) \end{bmatrix} = \phi_i(XW_{ei} + \beta_{ci}), i = 1, 2, \cdots, n \quad (1)$$

where ϕ_i denotes the feature mapping function, $W_{ik} \in R^{M\times K}$ is the connection weight of the ith group of feature mappings to all input data and $\beta_{c_i} \in R^{1\times K}$ is the bias of the ith group of feature mappings, then the expression of all feature mappings Z^n in the feature node layer is shown in Equation (2).

$$Z^n = \begin{bmatrix} \phi_1(XW_{e1} + \beta_{e1}) \\ \phi_2(XW_{e2} + \beta_{e2}) \\ \vdots \\ \phi_n(XW_{en} + \beta_{en}) \end{bmatrix}^T = [Z_1 Z_2 \cdots Z_n], j = 1, 2, \ldots, m \quad (2)$$

Similarly, the expression for the jth group of enhanced nodes H_j is shown in Equation (3).

$$H_j = \begin{bmatrix} \zeta_j(Z_1 W_{j1} + \beta_{j1}) & \cdots & \zeta_j(Z_n W_{j1} + \beta_{j1}) \\ \vdots & \ddots & \vdots \\ \zeta_j\left(Z_1 W_{jl} + \beta_{jl}\right) & \cdots & \zeta_j\left(Z_n W_{jl} + \beta_{jl}\right) \end{bmatrix} = \zeta_j\left(Z^n W_{hj} + \beta_{hj}\right), j = 1, 2, \cdots, m \quad (3)$$

where ζ_j denotes the activation function, l is the number of the jth group of augmented nodes and $W_{hj} \in R^{kn\times l}$ and $\beta_{hj} \in R^{1\times l}$ are the connection weights and biases randomly generated by the system, then the expression of all the feature augmentation H^m in the augmented node layer is shown in Equation (4).

$$H^m = \begin{bmatrix} \zeta_1(Z^n W_{h1} + \beta_{h1}) \\ \zeta_2(Z^n W_{h2} + \beta_{h2}) \\ \vdots \\ \zeta_m(Z^n W_{hm} + \beta_{hm}) \end{bmatrix}^T = [H_1 H_2 \cdots H_m] \quad (4)$$

Then the expression of the final network output $\hat{Y}$ is shown in Equation (5).

$$\begin{aligned} \hat{Y} &= [Z_1, \ldots, Z_n \mid \zeta(Z^n W_{h1} + \beta_{h1}), \ldots \ldots, \zeta(Z^n W_{hm} + \beta_{hm})]W^m \\ &= [Z_1, \ldots, Z_n \mid H_1, \ldots \ldots, H_m)]W^m \\ &= [Z^n \mid H^m]W^m \end{aligned} \quad (5)$$

Then the expression for the final connection weight W^m is shown in Equation (6).

$$W^m = [Z^n \mid H^m]^+ Y \quad (6)$$

2.2. Sparrow Search Algorithm

Sparrow search algorithm (SSA) [34] is a novel swarm intelligence optimization algorithm based on the foraging and anti-predatory behaviors of sparrows. Its bionic principle is as follows: sparrows as explorers provide the search direction and region for the population, sparrows as followers search through the guidance of explorers, and sparrows as vigilantes rely on anti-predation strategies to avoid the population from falling into a local optimal solution.

The location update rules for the three types of sparrows are as follows:

(1) For sparrows as explorers, when the warning value is less than the safety value, it indicates that the sparrow has not found a predator and can perform a wide range of jumping searches, and when its warning value is greater than or equal to the safety value, it indicates that the sparrow has found a predator and immediately moves to other places for searching.
(2) For the sparrow as a follower, when its fitness value is less than or equal to half of the sparrows, it indicates that the sparrow did not obtain food and needs to move to other places to search, and when its fitness value is greater than half of the sparrows, it indicates that the sparrow can obtain food and will conduct a random search at the current location.
(3) For the sparrow as vigilant, when its fitness value is not equal to the current best fitness value, it indicates that the sparrow is at the edge of the population and is highly vulnerable to predators, and when its fitness value is equal to the current best fitness value, it indicates that the sparrow is in the middle of the population and needs to move closer to other sparrows to reduce the risk of being predated.

Suppose there are S sparrows in a D-dimensional search space, then the position of the ith sparrow in the D-dimensional search space is $X_i = [x_{ia}, \cdots, x_{id}, \cdots, x_{iD}], i = 1, 2, \cdots, S$, where x_{id} is the position of the ith sparrow in the d-dimension.

Sparrows as explorers generally account for 10–20% of the population, and their position is updated by the expression shown in Equation (7).

$$x_{id}^{i+1} = \begin{cases} x_{id}^{t} \cdot exp\left(\frac{-i}{\alpha T}\right) & R_2 < S_T \\ x_{id}^{t} + QL & R_2 \geqslant S_T \end{cases} \tag{7}$$

where t is the current number of iterations; T is the maximum number of iterations; α is a uniform random number between $(0,\ 1]$; Q is a random number obeying the standard normal distribution; L is a matrix of size $1 \times d$ and all elements are 1; $R_2 \in\ [0,\ 1]$ is the warning value; $S_T \in\ [0.5,\ 1]$ is the safety value.

The other sparrows in the population act as followers, and the expression for their position update is shown in Equation (8).

$$x_{id}^{t+1} = \begin{cases} Q \cdot exp\left(\frac{xw_d^t - x_{id}^t}{i^2}\right) & i > \frac{n}{2} \\ xb_d^{t+1} + \left|x_{id}^t - xb_d^{t+1}\right| A^{+} \cdot L & i \leqslant \frac{n}{2} \end{cases} \tag{8}$$

where A is a $1 \times D$-dimensional matrix; xw_d^t is the worst position of the sparrow in the dth dimension when the population undergoes the tth iteration; xb_d^{t+1} is the optimal position of the sparrow in the dth dimension when the population undergoes the $t+1$th iteration.

The sparrows as vigilantes are some sparrows randomly selected from explorers and followers, generally accounting for 10–20% of the population size, and their position update expressions are shown in Equation (9).

$$x_{id}^{t+1} = \begin{cases} xb_d^t + \beta\left(x_{id}^t - xb_d^t\right) & f_i \neq f_g \\ x_{id}^t + K\left(\frac{x_{id}^t - xw_d^t}{|f_i - f_w| + e}\right) & f_i = f_g \end{cases} \tag{9}$$

where β and K are step control parameters, β is a random number obeying standard normal distribution, and K is a random number between $[-1,\ 1]$; e is a very small constant to avoid the case that the denominator is 0; f_i is the fitness value of the ith sparrow; f_g and f_w are the best fitness value and the worst fitness value in the current sparrow population.

3. The Proposed SSA-BLS

In the BLS model, the randomly generated weights and biases as well as its five hyper-parameters (convergence coefficient s, regularization coefficient c, the number of feature nodes N_f, the number of feature mapping groups N_m and the number of enhancement nodes N_e) all have an impact on its performance. Among them, the most influential on its model accuracy and model stability are the hyper-parameters N_f, N_m and N_e. However, these three hyper-parameters have a wide range of values, so it is difficult to determine the best combination of hyper-parameters by traditional methods. An optimized broad learning system by sparrow search algorithm with self-adjusting hyper-parameters, i.e., SSA-BLS, is proposed in this paper to enhance the model performance and generalization capability.

The pseudo-code of the proposed SSA-BLS algorithm is shown in Algorithm 1.

Algorithm 1. The pseudo-code of SSA-BLS

Input:
MaxIter: the maximum iterations
dim: the number of hyper-parameters to be optimized
pop: the number of hyper-parameter combination populations
lb&ub: hyper-parameter combination search range
X: the initial population of hyper-parameter combinations
Output:
the optimal hyper-parameter combination X_{best}
best fitness value f_{best}
Iterative Curve IC

Establish an objective function $f(x)$, i.e., the AVG of RMSE obtained by 10-fold cross-validation;
Generate *pop* hyper-parameter combinations as initial population;
Calculate the fitness values by BLS;
while $t < MaxIter$ **do**
    Randomly select hyper-parameter combinations as explorers, followers and vigilantes;
    for each $i = explorer$ **do**
        Using Equation (7) to update locations;
    end
    for each $i = follower$ **do**
        Using Equation (8) to update locations;
    end
    for each $i = vigilante$ **do**
        Using Equation (9) to update locations;
    end
    Calculate the fitness values by BLS;
    Compare with previous f_{best};
    if *the current values better than* f_{best} **then**
        Update the f_{best} and X_{best};
    end
    Save current f_{best} to IC;
    $t = t + 1$
end

The determination steps of three hyper-parameters are summarized as follows:

(1) Generate a certain number of hyper-parameter combinations randomly as the initial population for optimization.
(2) Calculate the fitness value of the hyper-parameter combinations in the initial population, which is the average of the root mean square error (RMSE) obtained by 10-fold cross-validation of the testing set.

The expression for calculating the root mean square error (RMSE) is shown in Equation (10).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{T}(\hat{y}_i - y_i)^2} \tag{10}$$

where T is the number of samples, y_i denotes the actual value and $\hat{y}_i$ denotes the predicted value.

The schematic diagram of the fitness value calculating process is shown in Figure 2.

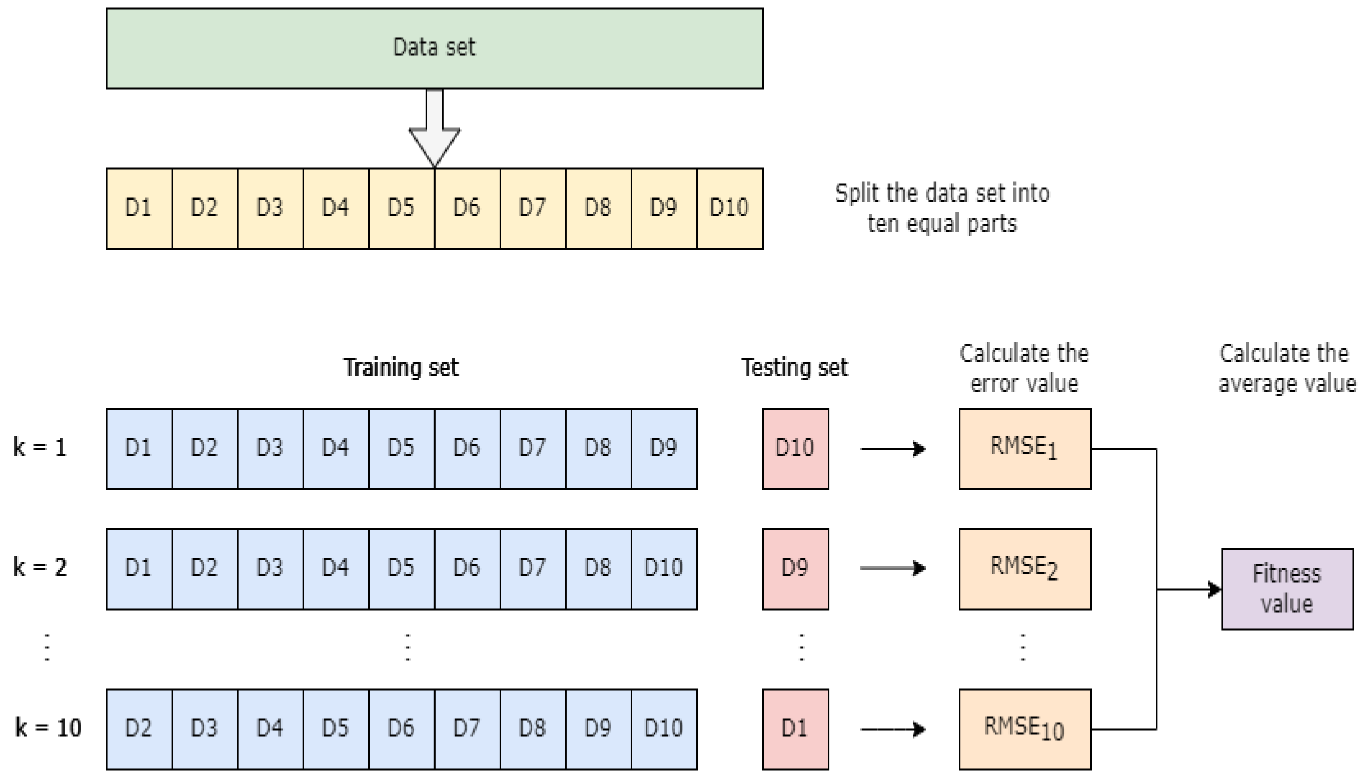

Figure 2. The schematic diagram of the fitness value calculating process, k is the number of cross-validations.

(3) A certain number of hyper-parameter combinations are randomly selected from the initial population as optimized explorers, and the positions are updated according to Equation (7).
(4) The other hyper-parameter combinations in the initial population act as optimized followers, and the positions are updated according to Equation (8).
(5) A certain number of hyper-parameter combinations are randomly selected from the optimized explorers and followers as the optimized vigilantes, and the positions are updated according to Equation (9).
(6) Repeat steps (3)–(5) until the maximum number of iterations is reached, and output the individual with the highest fitness value, i.e., the hyper-parameter combination that makes the smallest average of RMSE obtained from a 10-fold cross-validation of the testing set.

According to the above explanations, the flowchart of the SSA-BLS is shown in Figure 3.

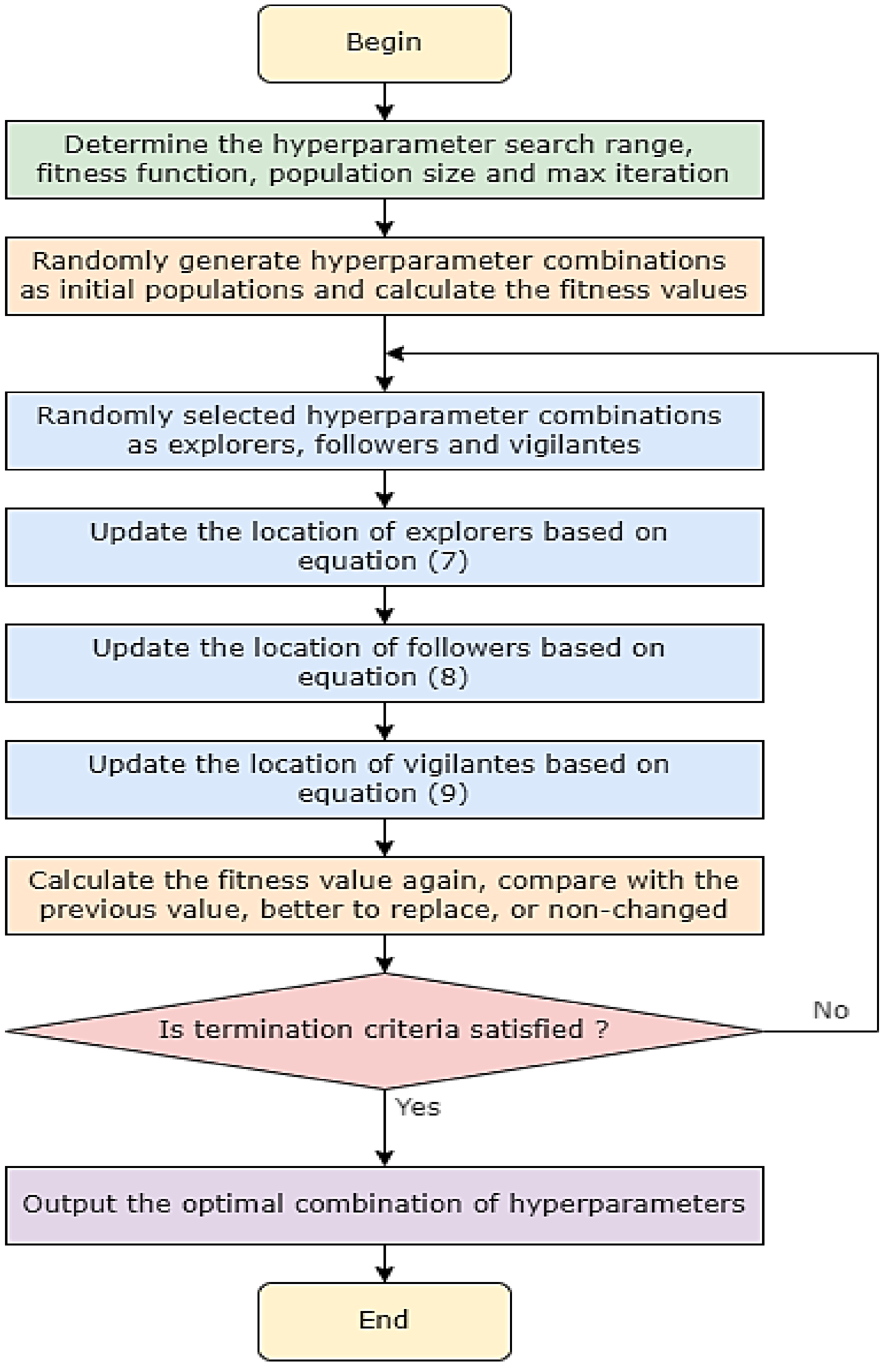

Figure 3. The flowchart of SSA-BLS.

4. Simulation

In order to evaluate the performance of the proposed SSA-BLS, it was applied to the ten benchmark regression datasets listed in Table 1, where the dataset Gasoline octane is from the web: https://www.heywhale.com/home (accessed on 5 January 2022), Fuel consumption is from the web: https://www.datafountain.cn (accessed on 10 January 2022) and the other datasets are from the web: http://www.liaad.up.pt/~ltorgo/Regression/DtaSets.html (accessed on 9 December 2021). All evaluations for RELM, KELM, BLS and SA-ELM were carried out in MacOS Mojave 10.14.6 and Python 3.9.9, running on a laptop with AMD Intel Iris Plus Graphics 645 1536MB, Processor 1.4GHz Intel Core i5 and RAM 8GB 2133MHz.

Table 1. Description of regression data sets.

Datasets	Attributes	Instances	Training Samples	Testing Samples
Gasoline octane	14	324	227	97
Fuel consumption	13	1067	747	320
Auto MPG	7	392	274	118
Abalone	8	4177	2923	1254
Bank domains	8	8192	3149	1350
Boston housing	13	506	354	152
Delta elevators	6	9517	6661	2856
Forest fires	12	517	361	156
Machine CPU	6	209	146	63
Servo	4	167	116	51

The parameters in SSA-BLS were set as follows: the initial population size and the maximum number of iterations for hyper-parameter optimization were set to 20 and 100. The compression factor in the mapping layer was set to 0.8 and the regularization factor in the enhancement layer was set to 2. The optimization range of hyper-parameter combination is shown in Table 2.

Table 2. The optimization range of hyper-parameter combination.

Hyper-Parameters	Meaning	Optimal Scope
N_f	Number of feature nodes	$[1, 20]$
N_m	Number of feature mapping groups	$[1, 40]$
N_e	Number of enhanced nodes	$[1, 500]$

Model accuracy and model stability were assessed by the average (AVG) and standard deviation (Sd) of the RMSE obtained from the ten-fold cross-validation. The averages (AVG) of the MAPE obtained from the ten-fold cross-validation were also used to evaluate the model accuracy. A smaller average value indicates higher model accuracy, and a smaller standard deviation indicates better model stability, and vice versa.

SSA-BLS was applied to the ten benchmark regression datasets in Table 1 and compared with BLS, RELM and KELM; the simulation results are shown in Tables 3–5. The hyper-parameters of the compared algorithm are determined by using the nested cross-validation method [42]. And the bolds in the table indicate the best experimental results of the four algorithms on each dataset.

Table 3. The RMSE of the four algorithms on the training set.

Datasets	RELM		KELM		BLS		SSA-BLS	
	RMSE		RMSE		RMSE		RMSE	
	AVG	SD	AVG	SD	AVG	SD	AVG	SD
Gasoline octane	4.47×10^{-2}	9.02×10^{-3}	3.83×10^{-2}	$\mathbf{7.99 \times 10^{-3}}$	3.86×10^{-2}	8.16×10^{-3}	$\mathbf{3.78 \times 10^{-2}}$	8.16×10^{-3}
Fuel consumption	2.47×10^{-2}	8.38×10^{-4}	1.79×10^{-2}	$\mathbf{1.49 \times 10^{-4}}$	4.09×10^{-2}	1.76×10^{-3}	$\mathbf{9.93 \times 10^{-3}}$	2.37×10^{-3}
Auto MPG	1.42×10^{-4}	5.04×10^{-5}	$\mathbf{1.00 \times 10^{-5}}$	$\mathbf{1.34 \times 10^{-7}}$	5.75×10^{-5}	3.49×10^{-5}	1.04×10^{-5}	3.50×10^{-5}
Abalone	1.38×10^{-5}	8.43×10^{-6}	2.70×10^{-6}	$\mathbf{4.41 \times 10^{-8}}$	4.06×10^{-3}	5.37×10^{-3}	$\mathbf{4.82 \times 10^{-8}}$	7.11×10^{-8}
Bank domains	3.01×10^{-4}	3.47×10^{-5}	4.13×10^{-6}	2.57×10^{-8}	5.57×10^{-5}	3.32×10^{-5}	$\mathbf{7.59 \times 10^{-9}}$	$\mathbf{7.82 \times 10^{-9}}$
Boston housing	7.23×10^{-4}	1.72×10^{-4}	6.83×10^{-6}	6.83×10^{-8}	1.86×10^{-5}	1.07×10^{-5}	$\mathbf{9.89 \times 10^{-8}}$	$\mathbf{6.25 \times 10^{-8}}$
Delta elevators	5.28×10^{-7}	1.02×10^{-7}	2.32×10^{-8}	$\mathbf{3.41 \times 10^{-10}}$	2.08×10^{-7}	1.61×10^{-7}	$\mathbf{3.85 \times 10^{-9}}$	3.42×10^{-9}
Forest fires	3.32×10^{-4}	6.63×10^{-5}	5.97×10^{-5}	6.15×10^{-7}	2.24×10^{-4}	1.61×10^{-4}	$\mathbf{2.93 \times 10^{-7}}$	$\mathbf{1.44 \times 10^{-7}}$
Machine CPU	1.33×10^{-4}	1.52×10^{-4}	3.42×10^{-5}	3.56×10^{-6}	8.86×10^{-7}	5.15×10^{-7}	$\mathbf{2.57 \times 10^{-8}}$	$\mathbf{1.28 \times 10^{-8}}$
Servo	1.44×10^{-4}	5.06×10^{-5}	3.70×10^{-5}	3.96×10^{-7}	6.42×10^{-4}	5.02×10^{-4}	$\mathbf{3.83 \times 10^{-7}}$	$\mathbf{2.87 \times 10^{-7}}$

Table 4. The RMSE of the four algorithms on the testing set.

Datasets	RELM		KELM		BLS		SSA-BLS	
	RMSE		RMSE		RMSE		RMSE	
	AVG	SD	AVG	SD	AVG	SD	AVG	SD
Gasoline octane	7.16×10^{-2}	$\mathbf{9.68 \times 10^{-3}}$	2.70×10^{-2}	3.09×10^{-2}	3.04×10^{-2}	3.20×10^{-2}	$\mathbf{2.65 \times 10^{-2}}$	3.07×10^{-2}
Fuel consumption	2.64×10^{-2}	$\mathbf{1.26 \times 10^{-3}}$	2.09×10^{-2}	3.43×10^{-3}	4.85×10^{-2}	2.79×10^{-3}	$\mathbf{1.39 \times 10^{-2}}$	2.84×10^{-3}
Auto MPG	2.06×10^{-4}	5.65×10^{-5}	$\mathbf{1.09 \times 10^{-5}}$	$\mathbf{2.18 \times 10^{-6}}$	5.82×10^{-5}	3.58×10^{-5}	1.19×10^{-5}	3.06×10^{-5}
Abalone	1.71×10^{-5}	7.95×10^{-6}	2.86×10^{-6}	5.16×10^{-6}	4.18×10^{-3}	5.76×10^{-3}	$\mathbf{8.34 \times 10^{-8}}$	$\mathbf{1.80 \times 10^{-7}}$
Bank domains	3.34×10^{-4}	3.02×10^{-5}	4.27×10^{-6}	2.69×10^{-7}	1.94×10^{-4}	4.13×10^{-4}	$\mathbf{7.79 \times 10^{-9}}$	$\mathbf{8.32 \times 10^{-9}}$
Boston housing	2.11×10^{-3}	7.65×10^{-4}	7.40×10^{-6}	1.83×10^{-6}	1.85×10^{-5}	1.08×10^{-5}	$\mathbf{9.69 \times 10^{-8}}$	$\mathbf{6.07 \times 10^{-8}}$
Delta elevators	6.18×10^{-7}	1.26×10^{-7}	3.04×10^{-8}	6.40×10^{-9}	2.08×10^{-7}	1.61×10^{-7}	$\mathbf{3.85 \times 10^{-9}}$	$\mathbf{3.41 \times 10^{-9}}$
Forest fires	1.99×10^{-3}	1.21×10^{-3}	7.24×10^{-5}	1.45×10^{-5}	2.2×10^{-4}	1.58×10^{-4}	$\mathbf{2.94 \times 10^{-7}}$	$\mathbf{1.43 \times 10^{-7}}$
Machine CPU	5.43×10^{-4}	2.41×10^{-4}	7.36×10^{-5}	9.17×10^{-5}	8.13×10^{-7}	5.16×10^{-7}	$\mathbf{2.28 \times 10^{-8}}$	$\mathbf{1.61 \times 10^{-8}}$
Servo	1.89×10^{-4}	6.42×10^{-5}	3.22×10^{-5}	1.25×10^{-5}	8.13×10^{-4}	8.62×10^{-4}	$\mathbf{3.81 \times 10^{-7}}$	$\mathbf{2.39 \times 10^{-7}}$

Table 5. The MAPE of the four algorithms on the training set and testing set.

Datasets	RELM		KELM		BLS		SSA-BLS	
	Train	Test	Train	Test	Train	Test	Train	Test
Gasoline octane	8.16×10^{-1}	1.33	2.57	3.68	9.29×10^{-1}	1.00	$\mathbf{9.24 \times 10^{-1}}$	$\mathbf{9.76 \times 10^{-1}}$
Fuel consumption	2.56	**2.59**	2.41×10^{-1}	4.16×10^{-1}	2.63	2.71	**1.09**	**2.59**
Auto MPG	3.30×10^{-2}	3.51×10^{-2}	1.11×10^{-1}	1.17×10^{-1}	6.60×10^{-4}	$\mathbf{6.23 \times 10^{-4}}$	$\mathbf{6.26 \times 10^{-4}}$	$\mathbf{8.14 \times 10^{-4}}$
Abalone	2.56×10^{-2}	2.01×10^{-2}	2.00×10^{-2}	2.13×10^{-2}	2.51×10^{-1}	2.56×10^{-1}	$\mathbf{2.18 \times 10^{-6}}$	$\mathbf{2.30 \times 10^{-6}}$
Bank domains	3.62×10^{-1}	3.67×10^{-1}	3.68×10^{-2}	3.71×10^{-2}	7.99×10^{-2}	8.04×10^{-2}	$\mathbf{9.60 \times 10^{-7}}$	$\mathbf{9.68 \times 10^{-7}}$
Boston housing	9.81×10^{-1}	1.23	3.86×10^{-2}	4.31×10^{-2}	4.51×10^{-4}	4.57×10^{-4}	$\mathbf{8.54 \times 10^{-6}}$	$\mathbf{8.32 \times 10^{-6}}$
Delta elevators	2.51×10^{-2}	2.56×10^{-2}	1.31×10^{-2}	1.31×10^{-2}	3.21×10^{-2}	3.21×10^{-2}	$\mathbf{5.04 \times 10^{-4}}$	$\mathbf{5.05 \times 10^{-4}}$
Forest fires	4.78×10^{-1}	6.21×10^{-1}	5.31×10^{-2}	5.61×10^{-2}	1.57×10^{-2}	1.59×10^{-2}	$\mathbf{2.13 \times 10^{-5}}$	$\mathbf{2.14 \times 10^{-5}}$
Machine CPU	2.05×10^{-2}	1.27×10^{-1}	1.41×10^{-1}	1.76×10^{-1}	5.73×10^{-5}	5.69×10^{-5}	$\mathbf{1.46 \times 10^{-6}}$	$\mathbf{1.47 \times 10^{-6}}$
Servo	2.62×10^{-3}	3.29×10^{-3}	5.95×10^{-2}	6.44×10^{-2}	6.06×10^{-3}	7.91×10^{-3}	$\mathbf{1.12 \times 10^{-5}}$	$\mathbf{1.39 \times 10^{-5}}$

As shown in Table 4, for the testing samples of the datasets, compared with RELM, KELM and BLS, the proposed SSA-BLS obtains better model accuracy on nine benchmark regression problems (gasoline octane, fuel consumption, abalone, bank domains, Boston housing, delta elevators, forest fires, machine CPU, servo) and better model stability on seven benchmark regression problems (bank domains, Boston housing, forest fires, machine CPU, servo).

As shown in Table 5, for the training samples of the datasets, compared with RELM, KELM and BLS, the proposed SSA-BLS obtains better model performance on all ten benchmark regression problems and for the testing samples of the datasets, the proposed SSA-BLS obtains better model performance on nine benchmark regression problems (except auto MPG).

The effectiveness of SSA-BLS is proved by the above simulation experiments. However, SSA-BLS requires more computing time to establish the model compared with other related algorithms, so it is not suitable for online learning. In this paper, model training and testing belong to offline learning, so this algorithm mainly pursues the accuracy and stability of the model.

5. Real-World Design Problem

As a new neural network algorithm, BLS can effectively solve the modeling problems of complex systems. In this paper, the proposed SSA-BLS was applied to establish the prediction models for thermal efficiency (TE), NOx emission concentration and SO_2 emission concentration of a 330 MW circulating fluidized bed boiler (CFBB).

There are 27 variables affecting the thermal efficiency and harmful gas emission concentration of a CFBB, mainly including load, coal feeder feeding rate, the primary air velocity, the secondary air velocity, oxygen concentration in the flue gas and the carbon content of fly ash. The symbols and descriptions of each variable are shown in Table 6. A total of 10,000 data samples are collected from a 330MW CFBB under different operating loads, some of which are shown in Table 7.

Table 6. Description of variable symbols.

Symbol Name	Description
17ANO037	Boiler load
AFCOALQ	The first coal feeder coal volume
BFCOALQ	The second coal feeder coal volume
CFCOALQ	The third coal feeder coal volume
DFCOALQ	The fourth coal feeder coal volume
18ANO074	Average bed temperature in the upper part of the dense phase zone of the furnace
05F051	Primary air flow at the left duct burner inlet
05F061	Right duct burner inlet primary air flow
05T457	Primary air temperature at the left duct burner inlet
05T467	Right duct burner inlet primary air temperature
06F061	Total right side secondary air flow
06F052	Left side internal secondary air distribution flow
06F062	Right side internal secondary air distribution flow
06T453	The second secondary fan motor drive end bearing temperature
06T463	The first secondary fan motor drive end bearing temperature
17I021	The first limestone powder conveying motor current
17I011	The second limestone powder conveying motor current
CEMSO2	CEMS flue gas O2 concentration
CEMSTEMP	CEMS flue gas temperature
08A051	Carbon content of fly ash at the inlet of the left EDC
08A061	Carbon content of fly ash at the inlet of the right EDC
05T402	The first Primary fan inlet temperature
05T403	The second Primary fan inlet temperature
12T612	The first old slagger outlet temperature
12T622	The second cold slagger outlet temperature
12T632	The third cold slagger outlet temperature
12T642	The fourth cold slagger outlet temperature
CEMSNOX	CEMS flue gas NOx concentration
CEMSSO2	CEMS flue gas SO_2 concentration
TE	Boiler thermal efficiency

The boiler data is normalized and divided into training sets and testing sets in the ratio of 7:3.

The proposed SSA-BLS and BLS are applied to this boiler data and the experimental results are shown in Tables 8–11. The hyper-parameters of BLS are determined by using the nested cross-validation method [42]. And the bolds in the table indicate the best experimental results of the four algorithms on each objective.

Table 7. The partial data of a 330MW CFBB operational conditions.

NO.	17ANO037	AFCOALQ	BFCOALQ	CFCOALQ	DFCOALQ	18ANO074	05F051	05F061	05T457	05T467	06F061	06F052	06F062	06T453	06T463
1	73.401	38.065	39.174	39.122	38	864.328	202.548	220.4	269.342	267.375	385.664	182.617	163.927	278.823	267.433
2	73.401	38.065	39.174	39.122	38	864.328	266.631	232.072	269.342	267.375	385.31	195.301	152.674	278.823	267.433
3	73.52	38.065	39.174	39.122	38	864.207	249.237	263.656	269.342	267.375	435.575	178.803	171.079	278.823	267.433
4	73.52	38.065	39.174	39.122	38	864.03	263.656	242.371	269.342	267.375	405.929	209.128	186.146	278.823	267.433
5	73.52	37.962	39.174	39.122	37.862	863.881	298.673	248.093	269.342	267.375	402.92	183.762	177.659	278.823	267.433
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
9996	96.318	56.966	52.712	52.953	56.528	866.957	419.973	368.935	273.729	270.847	976.991	569.881	669.725	284.775	268.966
9997	96.427	56.966	52.556	52.953	56.528	867.103	338.267	273.955	273.729	270.847	973.451	626.621	598.299	284.775	268.966
9998	96.427	56.966	52.403	52.953	56.528	867.278	386.329	349.71	273.729	270.847	1017.08	631.294	601.922	284.775	268.966
9999	96.427	56.966	52.273	52.801	56.528	867.39	405.325	343.073	273.729	270.847	1020.796	569.118	612.603	284.775	268.966
10000	96.427	56.966	52.273	52.689	56.528	867.557	313.549	324.306	273.729	270.847	1036.903	543.752	645.026	284.775	268.966
NO.	**17I021**	**17I011**	**6CEMSO2**	**6CEMSTEMP**	**08A051**	**08A061**	**05T402**	**05T403**	**12T612**	**12T622**	**12T632**	**12T642**	**CEMSNOX**	**CEMSNOX**	**TE**
1	102.876	116.074	5.554	152.655	0.847	0.316	25.65	24.901	42.858	45.355	51.829	36.858	128.395	225.285	90.55405
2	103.944	114.815	5.554	152.655	0.847	0.316	25.65	24.901	42.858	45.355	51.829	36.858	128.395	224.141	90.55405
3	103.296	113.175	5.554	152.655	0.847	0.316	25.65	24.901	42.858	45.355	51.829	36.858	128.929	223.378	90.55405
4	103.334	114.701	5.554	152.655	0.847	0.316	25.65	24.901	42.858	45.355	51.829	36.858	129.463	220.517	90.55405
5	104.059	113.328	5.554	152.655	0.847	0.316	25.65	24.901	42.858	45.355	51.829	36.858	129.463	218.61	90.55405
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
9996	102.914	116.684	4.921	156.637	1.405	0.167	30.235	28.193	39.127	37.046	36.846	47.32	160.436	144.991	90.30709
9997	102.914	116.99	4.921	156.637	1.405	0.167	30.235	28.193	39.127	37.046	36.846	47.32	160.436	147.089	90.30709
9998	103.792	117.714	4.921	156.637	1.405	0.167	30.235	28.193	39.127	37.046	36.846	47.32	160.436	148.615	90.30709
9999	102.418	115.998	4.921	156.637	1.405	0.167	30.235	28.193	39.127	37.046	36.192	47.32	160.436	150.14	90.30709
10000	101.541	115.731	4.921	156.637	1.405	0.167	30.235	28.193	39.127	37.046	36.192	47.32	159.826	151.666	90.30709

Table 8. The hyper-parameters of SSA-BLS and BLS for boiler data modeling.

Objectives	SSA-BLS		
	N1	N2	N3
NOx	2	37	478
SO_2	1	13	459
TE	2	27	296

Table 9. The RMSE of the four algorithms on the training set of boiler data.

Objectives	RELM		KELM		BLS		SSA-BLS	
	RMSE		RMSE		RMSE		RMSE	
	AVG	SD	AVG	SD	AVG	SD	AVG	SD
NO_X	9.05×10^{-2}	6.51×10^{-3}	3.48×10^{-2}	$\mathbf{4.26 \times 10^{-5}}$	4.22×10^{-2}	9.50×10^{-4}	$\mathbf{2.17 \times 10^{-2}}$	4.23×10^{-4}
SO_2	8.31×10^{-2}	2.37×10^{-3}	$\mathbf{1.99 \times 10^{-2}}$	1.49×10^{-4}	5.56×10^{-2}	7.12×10^{-4}	3.37×10^{-2}	$\mathbf{1.13 \times 10^{-3}}$
TE	4.52×10^{-2}	1.01×10^{-2}	2.04×10^{-2}	$\mathbf{3.36 \times 10^{-5}}$	5.96×10^{-2}	1.20×10^{-3}	$\mathbf{4.41 \times 10^{-3}}$	1.72×10^{-4}

Table 10. The RMSE of the four algorithms on the testing set of boiler data.

Objectives	RELM		KELM		BLS		SSA-BLS	
	RMSE		RMSE		RMSE		RMSE	
	AVG	SD	AVG	SD	AVG	SD	AVG	SD
NO_X	8.91×10^{-2}	5.94×10^{-3}	7.49×10^{-2}	2.36×10^{-3}	6.53×10^{-2}	6.55×10^{-2}	$\mathbf{2.75 \times 10^{-2}}$	$\mathbf{1.83 \times 10^{-3}}$
SO_2	7.91×10^{-2}	2.09×10^{-2}	4.12×10^{-2}	5.53×10^{-3}	5.82×10^{-2}	4.98×10^{-3}	$\mathbf{3.59 \times 10^{-2}}$	$\mathbf{1.94 \times 10^{-3}}$
TE	4.48×10^{-2}	9.68×10^{-3}	5.37×10^{-2}	2.85×10^{-3}	6.26×10^{-3}	1.18×10^{-3}	$\mathbf{4.58 \times 10^{-3}}$	$\mathbf{1.60 \times 10^{-4}}$

Table 11. The MAPE of the four algorithms on the training set and testing set of boiler data.

Objectives	RELM		KELM		BLS		SSA-BLS	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
NOX	4.22	4.13	1.74	2.46	2.20	2.22	1.57	1.83
SO_2	4.14	4.10	1.07	1.20	3.14	3.32	1.92	2.06
TE	2.28	2.26	9.26×10^{-2}	1.98	3.03×10^{-1}	3.09×10^{-1}	2.10×10^{-1}	2.18×10^{-1}

As shown in Tables 9–11, compared with RELM, KELM and BLS, the proposed SSA-BLS obtained better model accuracy and model stability in the prediction models established for the NOx emissions concentration and thermal efficiency of CFBB both on the training set and testing set. However, its model prediction accuracy for the SO_2 emission concentration of CFBB was not as good as that of KELM.

The fitting diagrams and error diagrams of SSA-BLS for modeling the three objectives of CFBB on the testing set are shown in Figures 4–9, where the red line and the blue line in the fitting diagram indicate the true values and predicted values of the partial testing set, and the curve in the error diagram represent the error of the predicted values compared to the true values for the testing set.

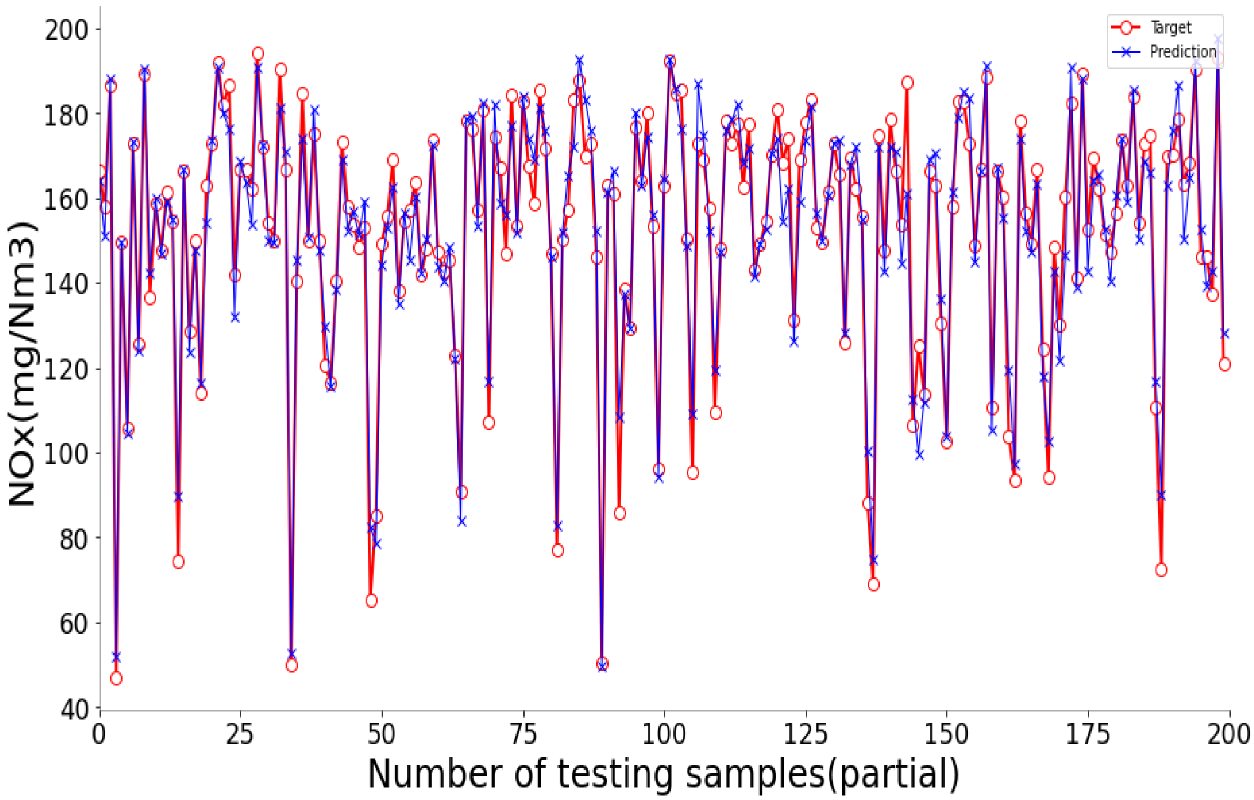

Figure 4. The fitting diagram of NOx emission concentration model.

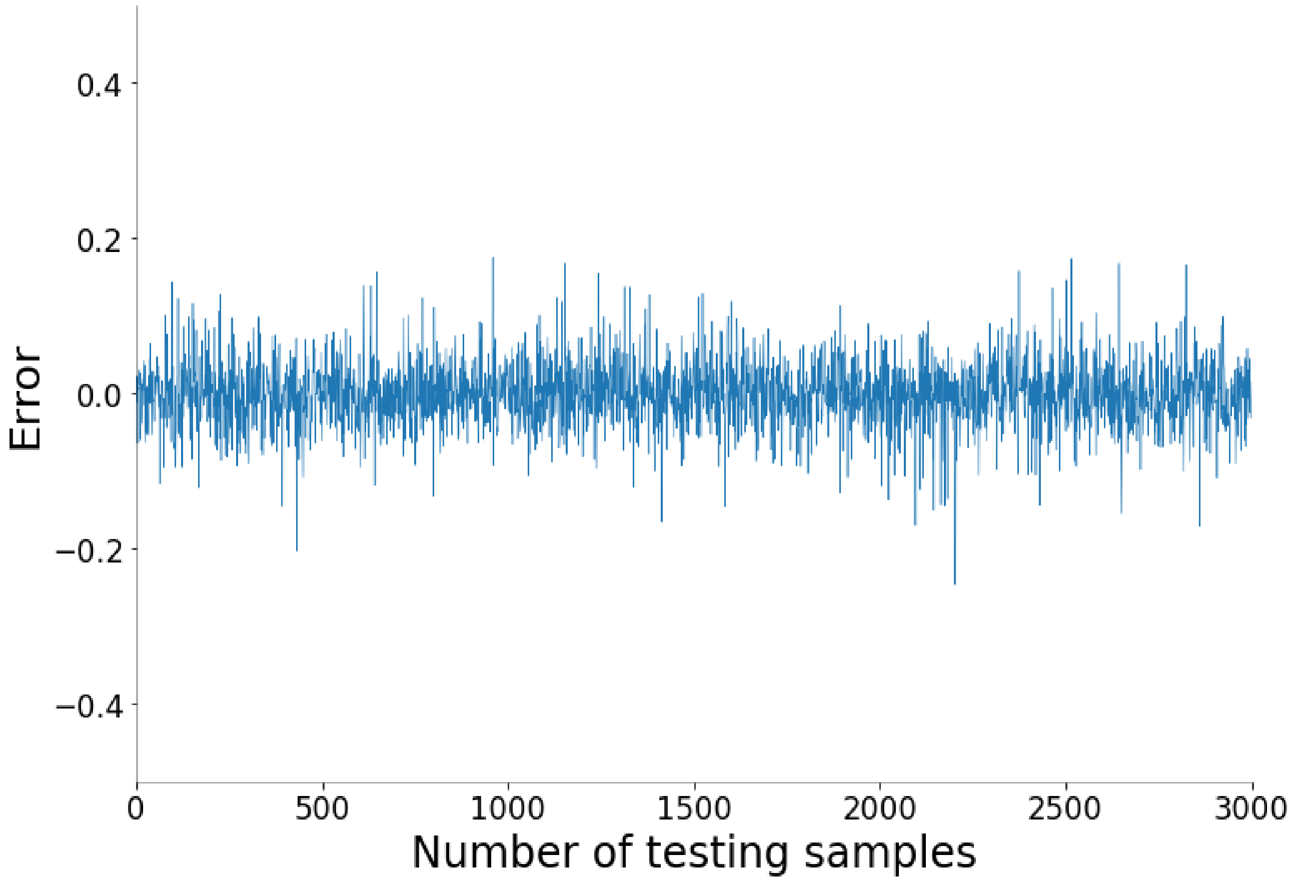

Figure 5. The error diagram of NOx emission concentration model.

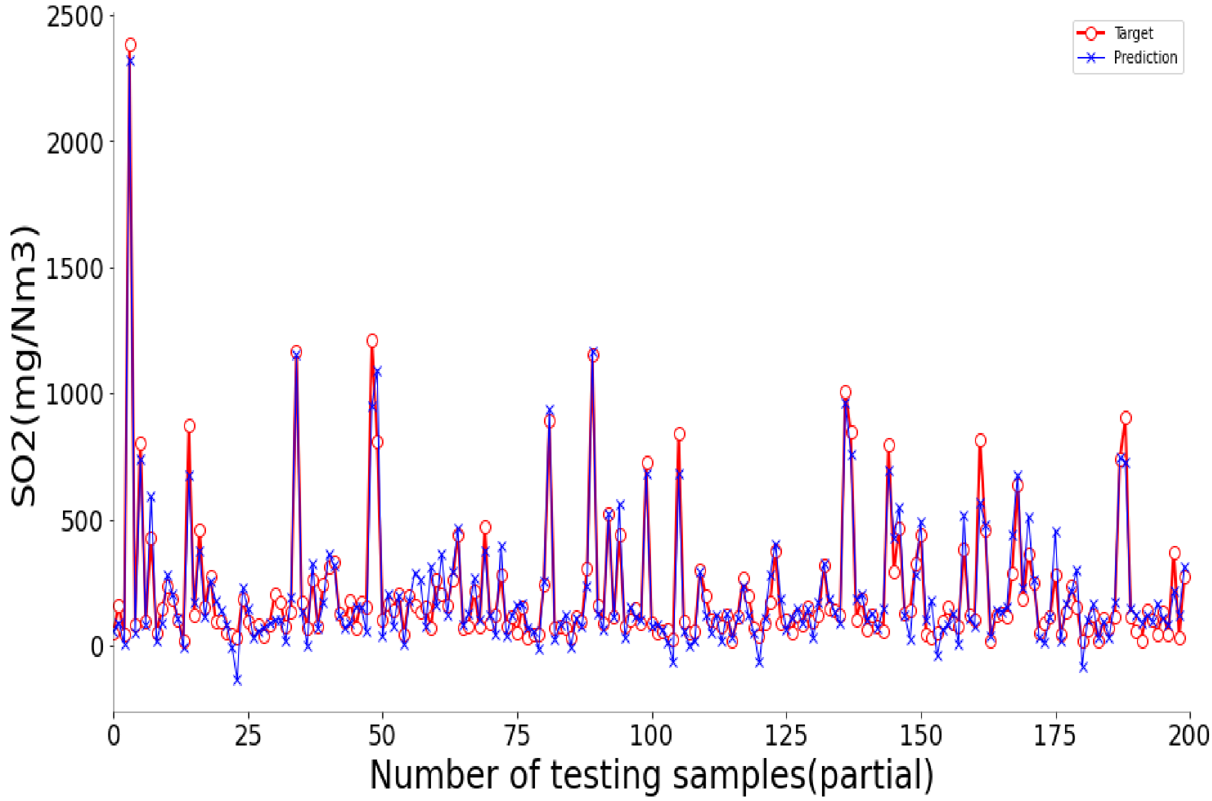

Figure 6. The fitting diagram of SO_2 emission concentration model.

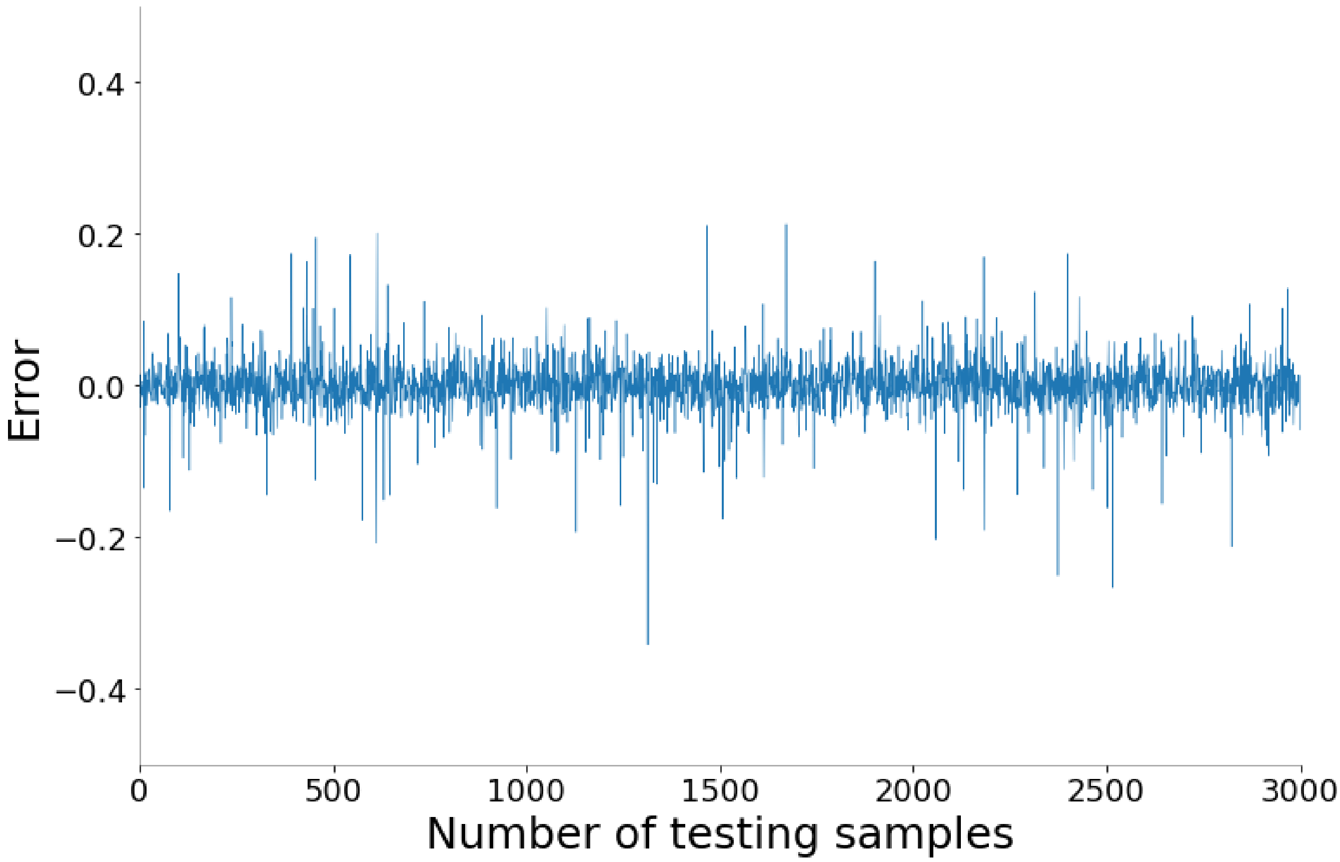

Figure 7. The error diagram of SO_2 emission concentration model.

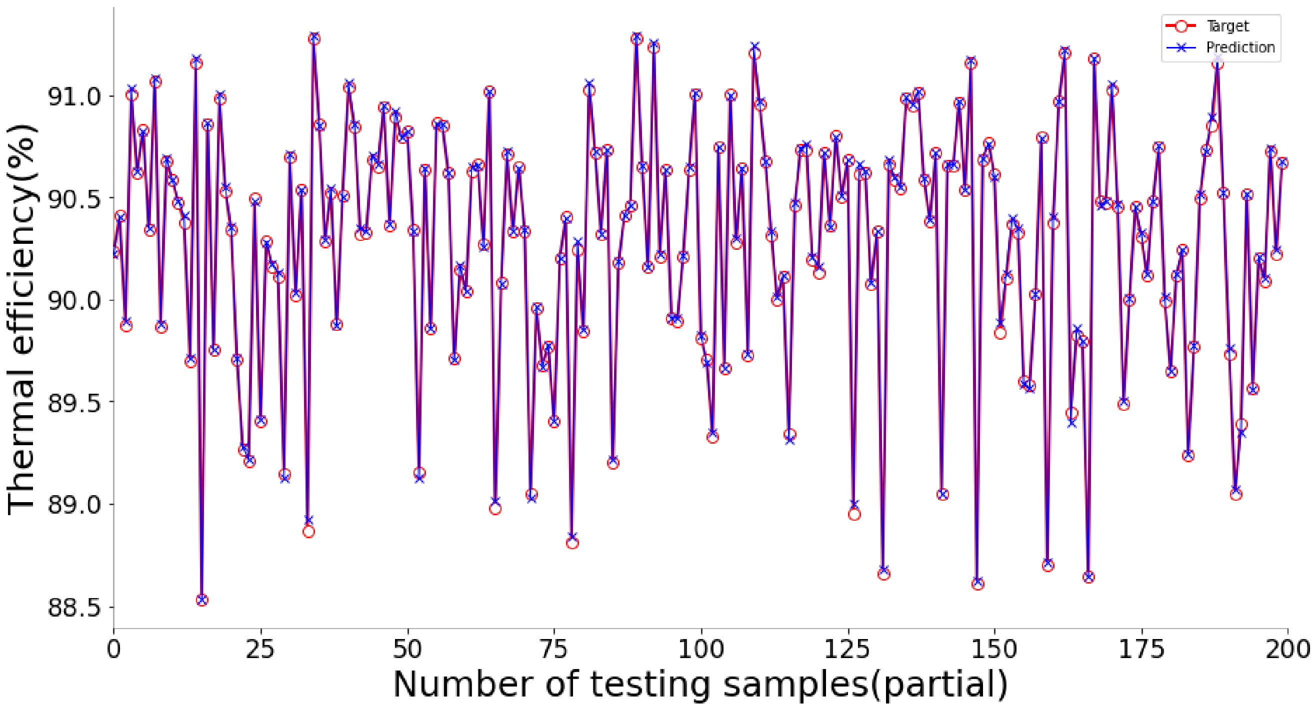

Figure 8. The fitting diagram of thermal efficiency model.

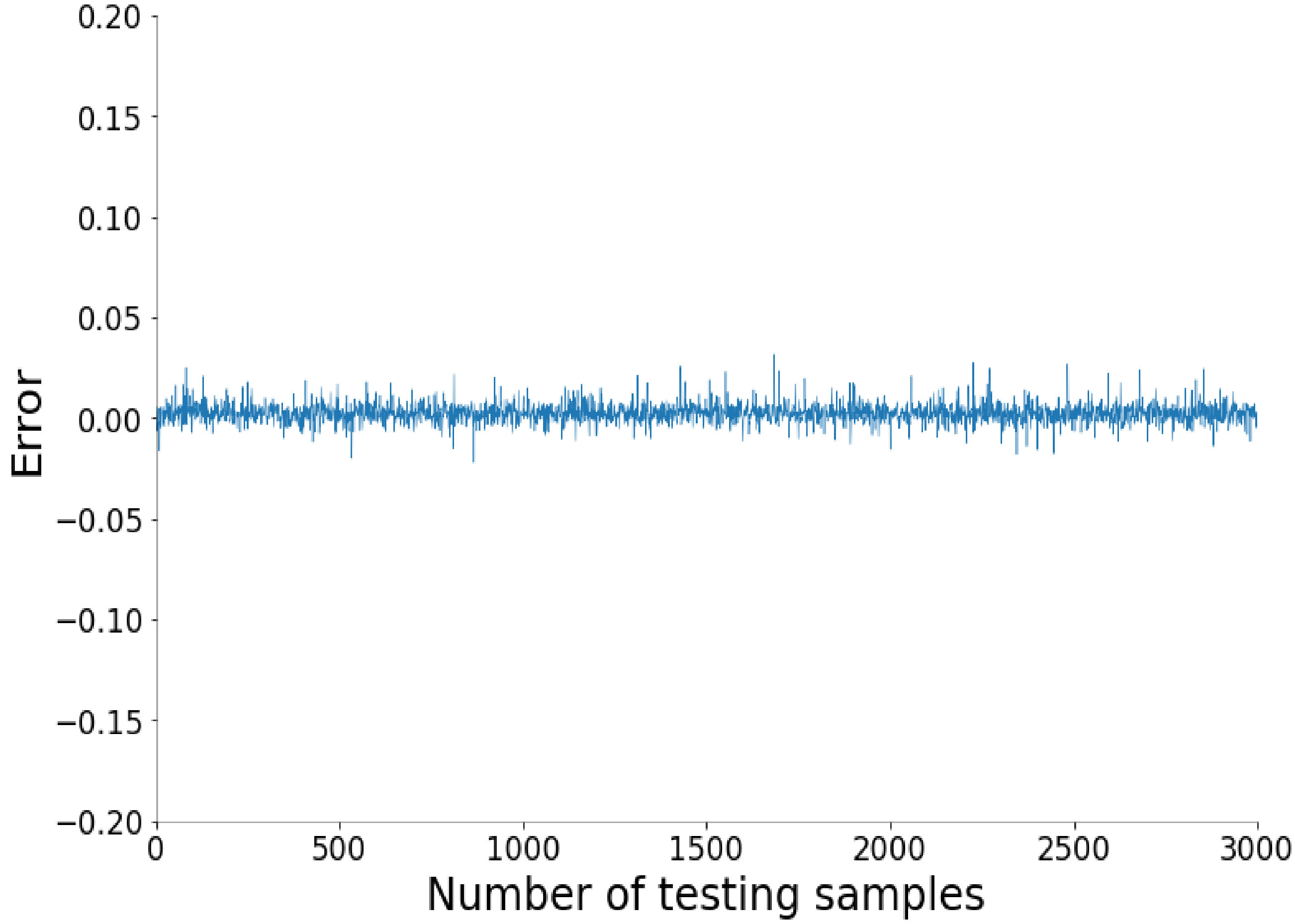

Figure 9. The error diagram of thermal efficiency model.

As shown in Figures 4–9, the proposed SSA-BLS effectively establishes the prediction models for the three objectives of CFBB. The three prediction models all have great fitting effect and small fitting error on the testing set.

6. Conclusions

This paper proposed a novel optimized broad learning system by combining with a sparrow search algorithm. That is to say, the sparrow search algorithm was used to optimize the hyper-parameters of broad learning systems. Compared with other state-of-the-art methods, the proposed SSA-BLS reveals better regression accuracy and model stability on testing ten benchmark datasets. Additionally, the SSA-BLS was used to build the collective model of thermal efficiency, NOx and SO_2 emissions concentration of one 330MW circulation fluidized bed boiler. Experiment results show that the model accuracy can be achieved 10^{-2}–10^{-3}. The proposed SSA-BLS is an effective modelling method.

However, the proposed SSA-BLS takes more time to determine the optimal hyper-parameters. This method improves the accuracy of the model but reduces the speed of model computation. Moreover, this method is also not applicable to online modeling due to the long modeling time. In addition, SSA-BLS only tunes and optimizes the hyperparameters in terms of model accuracy, while ignoring the model stability aspects. In the future, the performance of SSA-BLS will be further improved, including computation speed, model complexity and generalization ability. Additionally, based on the established comprehensive model, we will use one heuristic optimization algorithm to adjust the boiler's operational parameters for enhancing thermal efficiency and reducing NOx/SO_2 emissions concentration.

Author Contributions: Conceptualization, Y.M. and C.X.; methodology, C.X.; software, H.W.; validation, H.W., C.X. and Y.M.; formal analysis, C.X.; investigation, X.G.; resources, Y.M.; data curation, H.W. and C.X.; writing—original draft preparation, C.X.; writing—review and editing, X.G.; visualization, H.W. and C.X.; supervision, R.W.; project administration, S.L.; funding acquisition, Y.M., X.G., R.W. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (Grant No. 62203332) and the Natural Science Foundation of Tianjin (Grant No. 20JCQNJC00430) and the Special fund Project of Tianjin Technology Innovation Guidance (grand No. 21YDTPJC00370) and College Students' Innovative Entrepreneurial Training Plan Program (Grant No. 202110069034, 202110069003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restriction.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yip, I. Cost analysis of power generation in coal-fired thermal power plants under market economy. *Guangdong Electr. Power* **2002**, *15*, 5.
2. Zhou, H. Study of Some Key Issues in NOx Control and Combustion Optimization of Large Power Plant Boilers. Ph.D. Thesis, Zhejiang University, Hangzhou, China, 2004.
3. Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*; Cambridge University Press: Cambridge, UK, 2000.
4. Steinwart, I.; Christmann, A. *Support Vector Machines*; Information Science and Statistics; Springer: New York, NY, USA, 2008.
5. Zhou, H.; Zhu, H.; Cen, K. A real-time combustion optimization system for thermal power plant boilers based on artificial neural network and genetic algorithm. *Power Eng.* **2003**, *23*, 5.
6. Zheng, L.; Zhou, H.; Cen, K.; Wang, C. A comparative study of optimization algorithms for low NOx combustion modification at a coal-fired utility boiler. *Expert Syst. Appl.* **2009**, *36*, 2780–2793. [CrossRef]
7. Wu, F.; Zhou, H.; Ren, T.; Zheng, L.; Cen, K. Combining support vector regression and cellular genetic algorithm for multi-objective optimization of coal-fired utility boilers. *Fuel* **2009**, *88*, 1864–1870. [CrossRef]

8. Zhou, H.; Zheng, L.; Cen, K. Computational intelligence approach for NOx emissions minimization in a coal-fired utility boiler. *Energy Convers. Manag.* **2010**, *51*, 580–586. [CrossRef]
9. Zhou, H.; Zhao, J.; Zheng, L.; Wang, C.; Cen, K. Modeling NOx emissions from coal-fired utility boilers using support vector regression with ant colony optimization. *Eng. Appl. Artif. Intell.* **2012**, *25*, 147–158. [CrossRef]
10. Zhou, H.; Cen, K. Combining Neural Network or Support Vector Machine with Optimization Algorithms to Optimize the Combustion. In *Combustion Optimization Based on Computational Intelligence*; Springer: Singapore, 2018.
11. López-Ibáñez, M. Ant Colony Optimization. In Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '10, Portland, OR, USA, 7–11 July 2010.
12. Huang, G.; Zhu, Q.; Siew, C.K. Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 985–990.
13. Huang, G.; Zhu, Q.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
14. Huang, G.; Zhou, H.; Ding, X.; Zhang, R. Extreme Learning Machine for Regression and Multiclass Classification. IEEE Trans. *Syst. Man Cybern. Part B Cybern.* **2012**, *42*, 513–529. [CrossRef]
15. Li, G.; Niu, P.; Ma, Y.; Wang, H.; Zhang, W. Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency. *Knowl. Based Syst.* **2014**, *67*, 278–289. [CrossRef]
16. Li, G.; Niu, P. Combustion optimization of a coal-fired boiler with double linear fast learning network. *Soft Comput.* **2016**, *20*, 149–156. [CrossRef]
17. Li, G.; Niu, P.; Duan, X.; Zhang, X. Fast learning network: A novel artificial neural network with a fast learning speed. *Neural Comput. Appl.* **2013**, *24*, 1683–1695. [CrossRef]
18. Li, G.; Niu, P.; Wang, H.; Liu, Y. Least Square Fast Learning Network for modeling the combustion efficiency of a 300WM coal-fired boiler. *Neural Netw. Off. J. Int. Neural Netw. Soc.* **2014**, *51*, 57–66. [CrossRef]
19. Niu, P.; Ma, Y.; Li, G. Model NOx emission and thermal efficiency of CFBB based on an ameliorated extreme learning machine. *Soft Comput.* **2018**, *22*, 4685–4701. [CrossRef]
20. Ma, Y.; Niu, P.; Zhang, X.; Li, G. Research and application of quantum-inspired double parallel feed-forward neural network. Knowl. *Based Syst.* **2017**, *136*, 140–149. [CrossRef]
21. Ma, Y.; Niu, P.; Yan, S.; Li, G. A modified online sequential extreme learning machine for building circulation fluidized bed boiler's NOx emission model. *Appl. Math. Comput.* **2018**, *334*, 214–226. [CrossRef]
22. Chen, C.L.; Liu, Z. Broad Learning System: An Effective and Efficient Incremental Learning System Without the Need for Deep Architecture. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 10–24. [CrossRef]
23. Hinton, G.E.; Osindero, S.; Teh, Y.W. A Fast-Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]
24. Salakhutdinov, R.; Hinton, G.E. Deep Boltzmann Machines. In Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater Beach, FL, USA, 16–18 April 2009.
25. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
26. Chen, C.L.; Liu, Z.; Feng, S. Universal Approximation Capability of Broad Learning System and Its Structural Variations. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 1191–1204. [CrossRef]
27. Jin, J.; Chen, C.L. Robust Broad Learning System for Uncertain Data Modeling. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; pp. 3524–3529.
28. Xu, M.; Han, M.; Chen, C.L.; Qiu, T. Recurrent Broad Learning Systems for Time Series Prediction. *IEEE Trans. Cybern.* **2020**, *50*, 1405–1417. [CrossRef]
29. Huang, H.; Zhang, T.; Yang, C.; Chen, C.L. Motor Learning and Generalization Using Broad Learning Adaptive Neural Control. *IEEE Trans. Ind. Electron.* **2020**, *67*, 8608–8617. [CrossRef]
30. Pu, X.; Li, C. Online Semisupervised Broad Learning System for Industrial Fault Diagnosis. *IEEE Trans. Ind. Inform.* **2021**, *17*, 6644–6654. [CrossRef]
31. Liu, Z.; Huang, S.; Jin, W.; Mu, Y. Broad Learning System for semi-supervised learning. *Neurocomputing* **2021**, *444*, 38–47. [CrossRef]
32. Zheng, Y.; Chen, B.; Wang, S.; Wang, W. Broad Learning System Based on Maximum Correntropy Criterion. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 3083–3097. [CrossRef]
33. Feng, S.; Chen, C.L. Fuzzy Broad Learning System: A Novel Neuro-Fuzzy Model for Regression and Classification. *IEEE Trans. Cybern.* **2020**, *50*, 414–424. [CrossRef]
34. Zhao, H.; Zheng, J.; Deng, W.; Song, Y. Semi-Supervised Broad Learning System Based on Manifold Regularization and Broad Network. In *IEEE Transactions on Circuits and Systems I: Regular Papers*; IEEE: Manhattan, NY, USA, 2020; Volume 67, pp. 983–994.
35. Nacef, A.; Kaci, A.; Aklouf, Y.; Dutra, D.L. Machine learning based fast self optimized and life cycle management network. *Comput. Netw.* **2022**, *209*, 108895. [CrossRef]
36. Ali, A.; Irshad, K.; Khan, M.F.; Hossain, M.M.; Al-Duais, I.N.; Malik, M.Z. Artificial Intelligence and Bio-Inspired Soft Computing-Based Maximum Power Plant Tracking for a Solar Photovoltaic System under Non-Uniform Solar Irradiance Shading Conditions—A Review. *Sustainability* **2021**, *13*, 10575. [CrossRef]

37. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow Search Algorithm. *Syst. Sci. Control. Eng.* **2020**, *8*, 22–34. [CrossRef]
38. Poli, R.; Kennedy, J.; Blackwell, T.M. Particle swarm optimization. *Swarm Intell.* **2007**, *1*, 33–57. [CrossRef]
39. Trelea, I.C. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Inf. Process. Lett.* **2003**, *85*, 317–325. [CrossRef]
40. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
41. Mirjalili, S.M.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
42. Saud, S.J.; Jamil, B.; Upadhyay, Y.; Irshad, K. Performance improvement of empirical models for estimation of global solar radiation in India: A k-fold cross-validation approach. *Sustain. Energy Technol. Assess.* **2020**, *40*, 100768. [CrossRef]
43. Zheng, W.; Qian, Y.; Lu, H. Text categorization based on regularization extreme learning machine. *Neural Comput. Appl.* **2011**, *22*, 447–456. [CrossRef]
44. Wang, M.; Chen, H.; Yang, B.; Zhao, X.; Hu, L.; Cai, Z.; Huang, H.; Tong, C. Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. *Neurocomputing* **2017**, *267*, 69–84. [CrossRef]
45. Chen, H.; Zhang, Q.; Luo, J.; Xu, Y.; Zhang, X. An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine. *Appl. Soft Comput.* **2020**, *86*, 105884. [CrossRef]
46. Parvandeh, S.; Yeh, H.; Paulus, M.P.; McKinney, B.A. Consensus features nested cross-validation. *Bioinformatics* **2020**, *36*, 3093–3098. [CrossRef]
47. Pao, Y.; Takefuji, Y. Functional-link net computing: Theory, system architecture, and functionalities. *Computer* **1992**, *25*, 76–79. [CrossRef]
48. Pao, Y.; Park, G.H.; Sobajic, D.J. Learning and generalization characteristics of the random vector Functional-link net. *Neurocomputing* **1994**, *6*, 163–180. [CrossRef]
49. Igelnik, B.; Pao, Y. Stochastic choice of basic functions in adaptive function approximation and the functional-link net. *IEEE Trans. Neural Netw.* **1995**, *6*, 1320–1329. [CrossRef]