Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 7372984, 20 pages
https://doi.org/10.1155/2022/7372984

Research Article

Equalization Optimizer-Based LSTM Application in Reservoir Identification

Fan Yang, Kewen Xia, Shurui Fan, and Zhiwei Zhang

Hebei University of Technology, College of Electronic Information Engineering, Tianjin 300401, China

Correspondence should be addressed to Kewen Xia; kwxia@hebut.edu.cn

Received 26 April 2022; Accepted 19 July 2022; Published 9 September 2022

Academic Editor: Daniele Bibbo

In recent years, the use of long short-term memory (LSTM) has made significant contributions to various fields and the use of intelligent optimization algorithms combined with LSTM is also one of the best ways to improve model shortcomings and increase classification accuracy. Reservoir identification is a key and difficult point in the process of logging, so using LSTM to identify the reservoir is very important. To improve the logging reservoir identification accuracy of LSTM, an improved equalization optimizer algorithm (TAFEO) is proposed in this paper to optimize the number of neurons and various parameters of LSTM. The TAFEO algorithm mainly employs tent chaotic mapping to enhance the population diversity of the algorithm, convergence factor is introduced to better balance the local and global search, and then, a premature disturbance strategy is employed to overcome the shortcomings of local minima. The optimization performance of the TAFEO algorithm is tested with 16 benchmark test functions and Wilcoxon rank-sum test for optimization results. The improved algorithm is superior to many intelligent optimization algorithms in accuracy and convergence speed and has good robustness. The receiver operating characteristic (ROC) curve is used to evaluate the performance of the optimized LSTM model. Through the simulation and comparison of UCI datasets, the results show that the performance of the LSTM model based on TAFEO has been significantly improved, and the maximum area under the ROC curve value can get 99.43%. In practical logging applications, LSTM based on an equalization optimizer is effective in well-logging reservoir identification, the highest recognition accuracy can get 95.01%, and the accuracy of reservoir identification is better than other existing identification methods.

1. Introduction

With the development of logging technology, the interpretation technology associated with it is gradually moving from qualitative and quantitative manual processing to the era of quantitative processing using machines. Traditional reservoir identification mainly relies on expert experience, construction of rendezvous plates, and other methods, but the conventional reservoir identification methods are subject to many human influence factors. Currently, increasingly scholars are proposing the use of artificial neural networks to solve the reservoir identification problem, thus effectively avoiding errors and improving production efficiency [1–3]. This study provides a basis for reservoir quality and oil-bearing evaluation of terrestrial shale reservoirs. However, these methods ignore the time-series nature of logging data and do not conform to the practical geological thinking and the logic of traditional geological analysis.

The reservoir data are temporal in nature, with strong backward and forward correlation, so the use of long short-term memory networks is considered for the processing of reservoir data. Long short-term memory (LSTM) is a special type of RNN that can solve the gradient explosion and gradient disappearance problems during the training of long sequences. They can also improve the performance of long sequences. At present, some scholars use LSTM to identify reservoirs. For example, Zhou et al. [4] established a Bi-LSTM network model, which can accurately identify different types of strata developed in storage space and significantly improve the accuracy of reservoir identification. Chen et al. [5]constructed a multilayer LSTM for fine reservoir parameter prediction. The results show that multilayer LSTM has better robustness and accuracy in prediction.

In recent years, LSTM has made breakthroughs in many fields, and the use of intelligent optimization algorithms combined with LSTM is also one of the best ways to improve model shortcomings and increase classification accuracy. For example, Xie et al. [6] used the enhanced gray wolf optimization(GWO) algorithm for the CNN-LSTM model of time-series prediction and indicated that the classification accuracy was improved. Peng et al. [7] applied the fruit fly algorithm (FOA) to optimize the hyperparameters of the LSTM neural network, and the results showed that the prediction accuracy of the FOA-LSTM model was greatly improved. Yang et al. [8] build an improved lion swarm algorithm (LSO) for the LSTM to optimize the hyperparameters of the LSTM model, and the results showed that the enhanced model has strong generalization ability and higher prediction accuracy. When using the LSTM model for reservoir identification, various parameters of the LSTM need to be artificially selected, resulting in insufficient accuracy in reservoir identification. A strategy is proposed to optimize the number of neurons and hyperparameters of the LSTM model by gradient descent using an improved equilibrium optimizer.

With the rapid development of intelligent algorithms, more and more intelligent algorithms are available. For example, in 2017, Mohamed et al. [9] proposed the moth swarm algorithm; Yang et al. [10] proposed the hunger games search algorithm in 2021; in 2022, Ia et al. [11] proposed the weighted mean of vector algorithm. The equilibrium optimizer (EO) is a new optimization algorithm inspired by the physical phenomenon of control volume mass balance proposed in 2020, and it is characterized by its high optimization finding capability and simple parameters [12], but it still suffers from a tendency to fall into local optima and slow convergence in practical applications. Therefore, it is necessary to improve the algorithm of the equilibrium optimizer to ensure the stability and effectiveness of the algorithm. Wang et al. [13] used backpropagation neural networks to predict more output data, which can achieve more efficient optimization and more reasonable fitness functions; Fan et al. [14] proposed a definition of certain particle concentrations based on OBL, a new nonlinear time control strategy, a novel population update, and a chaos-based strategy. Fu et al. [15] combined the strategies of the modal algorithm and fused EO and heat exchange optimization (TEO) to obtain a new highly equilibrium optimizer (HEO). Gupta et al. [16] used Gaussian variation and an additional exploratory search mechanism based on the concept of population partitioning and reconstruction to improve the convergence speed of the algorithm and obtain more accurate optimal solutions; although these methods achieved good results, the EO algorithm still needs to be improved in terms of convergence speed and accuracy.

This paper proposes a TAFEO algorithm to increase the convergence speed, improve the convergence accuracy, and avoid falling into the local optimum. To verify the effectiveness of the improved algorithm, an LSTM model based on the equilibrium optimizer is then constructed and applied to log reservoir identification to achieve desirable practical application results.

2. Materials and Methods

2.1. The Equilibrium Optimizer Algorithm and Its Improvements

2.1.1. Equalization Optimizer Algorithm. The equilibrium optimizer (EO) is primarily a physically heuristic optimization algorithm for dynamic mass balance in a strongly mixed type of controlled volume. The mass balance equation embodies the physical processes of mass entry, departure, and generation in the controlled volume and is generally described using a first-order differential equation as shown in

$$V\frac{dC}{dt} = QC_{eq} - QC + G, \tag{1}$$

where V is the control volume; C is the concentration in the control volume; Q is the volumetric flow rate into or out of the control volume; C_{eq} is the concentration within the control volume in the absence of mass production (i.e., at equilibrium); and G is the mass production rate within the control volume.

By solving the differential equation described by equation (1), we can find that

$$C = C_{eq} + \left(C_0 - C_{eq}\right)F + \frac{G(1-F)}{\lambda V}, \tag{2}$$

$$F = \exp\left(-\lambda\left(t - t_0\right)\right), \tag{3}$$

where F is the exponential term factor, λ is the flow rate, and C_0 is the initial concentration of the control volume at time t_0.

2.2. Improvement of the Equalization Optimizer Algorithm. At present, the equilibrium optimizer algorithm has an excellent performance in intelligent algorithms and has been applied to specific problems with significant effects, but it still suffers from slow convergence speed, insufficient initial population diversity, and ease to fall into local extremes. In this paper, three strategies are proposed to better the equilibrium optimizer algorithm: using tent chaotic mapping to enhance population diversity; introducing convergence factors to accelerate the convergence speed of the algorithm in the early stage, and the ability to search locally in the late stage; and using an early perturbation strategy to enhance the ability of the algorithm to jump out of the local optimum.

2.2.1. Improvement Strategies

(1) Tent Chaotic Mapping. The traversal of tent mapping has uniformity and randomness, which enables the algorithm to easily escape from local optimal solutions, thus maintaining the diversity of the population while improving the global search ability. Therefore, to obtain a good initial solution

position with a greater chance and speed up the convergence of the population, this paper adopts the tent chaotic mapping method with better traversal uniformity and faster iteration speed to improve the coverage space of the initial solution, which is calculated as shown

$$x(t+1)=\begin{cases}\dfrac{x(t)}{0.6}, & x(t)<0.6,\\[2ex] \dfrac{1-x(t)}{0.4}, & x(t)\geq 0.6.\end{cases} \tag{4}$$

Among others, $x(t) \in [0, 1]$.

(2) Convergence Factor. The exponential term coefficient F used to balance the local search and global search ability in the equalization optimizer algorithm uses constant coefficient weights, and the tendency of the obtained coefficients to change tends to be constant, which does not conform to the nonlinear optimization search law in the algorithm iteration process. To address the problem of slow convergence and low precision in the EO algorithm, a nonlinear decreasing strategy is proposed to balance the local and global search capabilities of the algorithm so that the algorithm has enough steps to search for spatially dispersed populations in the early iterative stage, and reduces the step size to facilitate local search in the late iterative stage of the algorithm, with the convergence factor A defined as shown in

$$A=\frac{2\left[e^{1-\left(t/t_{\max}\right)^{2}}-1\right]}{(e-1)}, \tag{5}$$

where t is the current number of iterations, and $t_{\max}$ is the maximum number of iterations.

(3) Early Perturbation Strategy. For the disadvantage that the equilibrium optimizer algorithm is easy to fall into the local optimum set, condition to determine whether the particles fall into the local optimum, and update the particle positions when the condition of falling into the local optimum is satisfied, thus introducing the early perturbation strategy, as shown in equations (6) and (7), when equation (12) is satisfied, we reset the positions of the particles so that they are randomly distributed around the gbest thereby jumping out of the local optimum, i.e.,

$$\left|F_g(t)-F_g(t-1)\right|<0.01\cdot\left|F_g(t)\right|, \tag{6}$$

$$x(t)=(\text{gbest}(t)+\text{gbest}(t-1))\cdot r_a, \tag{7}$$

where $F_g(t)/F_g(t-1)$ is the value of the function corresponding to the global optimum of the $t/t-1$ generation, respectively, and r_a is a random number of [−1, 1].

2.2.2. Improved Algorithms. Based on the above three improvement strategies, this paper's specific operation process and parameters of the improved algorithm are designed as follows:

(1) *Initialization.* The algorithm performs random initialization within the upper and lower bounds of each optimization variable, as shown in

$$C_i^0=C_{\min}+r_i\left(C_{\max}-C_{\min}\right),\quad i=1,2,\ldots,n, \tag{8}$$

where $C_{\min}$ and $C_{\max}$ are the lower and upper bound vectors of the optimization variables, respectively; r_i represents a vector of random numbers for individual i, whose dimension is the same as the dimension of the optimization space, with each element value being a random number from 0 to 1.

Using equation (4), the solution is generated by the random initialization.

(2) *Equilibrium State Pool.* To improve the global search capability of the algorithm and avoid falling into low-quality local optimal solutions, the equilibrium state (i.e., the optimal individual) in equation (9) will be selected from within the five currently optimal candidate solutions, which constitute the equilibrium state pool as follows:

$$C_{eq,\text{pool}}=\left\{C_{eq,1},C_{eq,2},C_{eq,3},C_{eq,4},C_{eq,\text{ave}}\right\}, \tag{9}$$

where $C_{eq,1}, C_{eq,2}, C_{eq,3}, C_{eq,4}$ are the best four solutions found as of the current iteration; $C_{eq,\text{ave}}$ represents the average state of these four solutions. It is worth noting that these five candidate solutions are chosen with the same probability of 0.2.

(3) *Exponential Term CoefficientF.* To better balance the local and global search of the algorithm, equation (3) is improved as follows:

$$F=a_1*\operatorname{sign}(r-0.5)\left[e^{-\lambda t}-1\right], \tag{10}$$

where a_1 is the constant coefficient of the weight of the global search; sign is the symbolic function; and r, λ all represent vectors of random numbers whose dimensions are the same as the dimension of the optimization space, with each element value being a random number from 0 to 1.

The constant factor weights a_1 are replaced by A in equation (5).

(4) *Mass Generation RateG.* To enhance the local search capability of the algorithm, the generation rate is designed as shown in

$$G=G_{CP}\left(C_{eq}-\lambda C\right), \tag{11}$$

$$G_{CP}=\begin{cases}0.5r_i, & \text{if } r_2\geq 0.5,\\ 0, & \text{otherwise},\end{cases} \tag{12}$$

where G_{CP} is a vector of generation rate control parameters; r_i is a vector of random numbers whose dimension is the same as the dimension of the optimization space, with each element value being a random number from 0 to 1; and r_2 is a random number range from 0 to 1.

Table 1: Algorithm steps of TAFEO.

```
Algorithm: TAFEO
Input: Population size: $N$, the maximum number of iterations: $T$, number of dimensions: $D$, parameters: $a_1, a_2$, GP
Output: Global optimal position $g_{\text{best}}$ and global optimal position adaptation $f_{g_{\text{best}}}$
Initializing the location of the population using the tent chaotic mapping
Set up four initial equilibrium candidate solutions
While $t < T$
    For $i = i$: D
        Calculate the fitness of the four equilibrium candidate solutions
    find$\overrightarrow{C}_{eq1} \sim \overrightarrow{C}_{eq4}$
    End for
    For $i = i$: N
        Find the best fit by comparing the fit between particles
    End
    Calculation $\overrightarrow{C}_{\text{ave}}$
        Building a balanced pool
    For $i = i$: N
        Generate the exponential term coefficients $F$, the mass generation rate $G$ and calculate the optimal solution according to
        equations (5), (10)–(12) and according to equation (13)
        Introduce a premature perturbation strategy to determine whether to fall into a local optimum solution and to determine
        whether to reset the particle position
    End
    Update the global optimal position $g_{\text{best}}$ and the global optimal position adaptation $f_{g_{best}}$
        $t = t+1$
End while
```

(5) *Solution Update.* For optimization problems, the individual solutions can be updated as follows, based on what is shown in

$$C = C_{eq} + \left(C_0 - C_{eq}\right)F + \frac{G(1-F)}{\lambda V}. \tag{13}$$

We use equation (6) to determine whether it is the optimal solution. If the current output is not the optimal solution, we use equation (7) to reset the particle position.

In summary, the above improvement algorithm that incorporates the three improvement strategies is named TAFEO, and the specific steps of the TAFEO algorithm are shown in Table 1.

2.3. Simulation Experiments and Analysis of Results. The computer configuration used for the simulation experiments was Intel Core i7 6700HQ with 3.6 GHz main frequency, 16 GB of memory, 64-bit operating system, and MATLAB R2020b as the computing environment. In the following experiments, we set the number of evaluations to $M = N * T = 30000$, the number of populations of all intelligent optimization algorithms was set to $N = 30$, and the maximum number of iterations to $T = 1000$. For each basic algorithm, the internal parameter settings are shown in Table 2.

To verify the effectiveness and generalization of the TAFEO algorithm, 16 international standard test functions are used, F1-F10 [17] is chosen from the common test functions, and F11-F16 is from CEC2017 [18]. The details of the test functions are shown in Table 3.

Table 2: Parameter settings of the optimization algorithms.

Algorithms	Parameter settings
EO	$a_1 = 2, a_2 = 1, GP = 0.5$
m-EO	$a_1 = 2, a_2 = 1, GP = 0.5$
MDSGEO	$a_1 = 2, a_2 = 1, GP = 0.5$
TAFEO	$a_1 = 2, a_2 = 1, GP = 0.5$
WOA	$a \in [2, 0], b = 1, l \in [-1, 1]$
GWO	$a \in [2, 0]$
MPA	$\text{FAD}_s = 0.2, P = 0.5$

2.3.1. Performance Comparison of Various Improved EO Algorithms. To verify the effectiveness of the improved TAFEO algorithm, the performance of the search iterations was compared with the basic EO algorithm, the m-EO algorithm [16], and the MDSGEO [18] on test functions. Among them, m-EO is proposed in the literature [16] as an improved equilibrium optimizer algorithm using Gaussian variance and based on population partitioning and reconstruction, and the MDSGEO algorithm in literature [18] as an enhanced equilibrium optimizer algorithm using sinusoidal pooling strategy and adaptive preferential gravity strategy. Sixteen benchmark test functions of Table 3 were selected to test the four algorithms. The simulation-optimization-seeking iteration curves are shown in Figure 1.

As shown in Figure 1, the improved TAFEO algorithm performs well in the tests of all the functions, the convergence accuracy is better than all algorithms, and the convergence speed is faster than other algorithms on most test functions except F8, F12, and F14. The comparative analysis shows that the improved strategy of this paper is feasible.

Table 3: Sixteen benchmark functions.

Function	Dim	Range	Optimal value
$F_1(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10,10]	0
$F_2(x) = \sum_{i=1}^{n} [x_i + 0.5]$	30	[−100, 100]	0
$F_3(x) = \sum_{i=1}^{n} i x_i^4 + \mathrm{random}[0, 1)$	30	[−1.28, 1.28]	0
$F_4(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[−500, 500]	0
$F_5(x) = -20\exp(-0.2\sqrt{1/n\sum_{i=1}^{n} x_i^2}) - \exp(1/n\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	[−5.12, 5.12]	0
$F_6(x) = \sum_{i=1}^{11} [a_i - x_1(b_i^2 + b_i x_2)/b_i^2 + b_i x_3 + x_4]^2$	4	[−5, 5]	0.00030
$F_7(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]\times$ $[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2, 2]	3
$F_8(x) = -\sum_{i=1}^{4} c_i \exp(\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]	−3.32
$F_9(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^I + c_i]^{-1}$	4	[0, 10]	−10.1532
$F_{10}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.5363
$F_{11}(\mathbf{x}) = f(\mathbf{M}(2.048(\mathbf{x} - \mathbf{o}_4)/100) + 1) + {F_{11}}^*$ $f(\mathbf{x}) = \sum_{i=1}^{D-1}(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	30	[−100, 100]	400
$F_{12}(\mathbf{x}) = f(\mathbf{M}(\lvert\mathbf{x} - \mathbf{o}_5)) + {F_{12}}^*$ $f(x) = \sum_{i=1}^{D-1}(x_i^2 - 10\cos(2\pi x_i) + 10)$	30	[−100, 100]	500
$F_{13}(\mathbf{x}) = f(\mathbf{M}(0.5(\mathbf{x} - \mathbf{o}_6)/\lvert 100)) + {F_{13}}^*$ $f(\mathbf{x}) = g(x_1, x_2) + g(x_2, x_3) + \cdots + g(x_{D-1}, x_D) + g(x_D, x_1)$ $g(x, y) = 0.5 + (\sin^2(\sqrt{x^2 + y^2}) - 0.5)/(1 + 0.001(x^2 + y^2))^2$	30	[−100, 100]	600
$F_{14}(\mathbf{x}) = f(\mathbf{M}(600(\mathbf{x} - \mathbf{o}_7)/100)) + {F_{14}}^*$ $f(x) = \min(\sum_{i=1}^{D-1}(\hat{x}_i - \mu_0)^2, dD + s\sum_{i=1}^{D-1}(\hat{x}_i - \mu_1)^2) + 10(D - \sum_{i=1}^{D-1}\cos(2\pi\hat{z}_i))$ $\mu_0 = 2.5, \mu_1 = -\sqrt{\mu_0^2 - d/s}, s = 1 - 1/2\sqrt{D + 20} - 8.2, d = 1$ $y = 10(\mathbf{x} - \mathbf{o})/100, \hat{x}_i = 2\mathrm{sign}(x_i^*)y_i + \mu_0, \quad \text{for } i = 1, 2, \ldots, D$ $z = \Lambda^{100}(\hat{x} - \mu_0)$	30	[−100, 100]	700
F15 composition function 1 $N = 3$ $p = [0.2, 0.4, 0.4]$ g_1: *Zakharov function* g_2: *Rosenbrock function* g_3: *Rastrigin's function*	30	[−100, 100]	1100
F16 composition function 2 $N = 4$ $p = [0.2, 0.2, 0.4, 0.4]$ g_1: *High Conditioned elliptic function* g_2: *Ackley's function* g_3: *Schaffer's F7 function* g_4: *Rastrigin's function*	30	[−100, 100]	1400

Table 4 shows the results of the different algorithms. To demonstrate the repeatability of each algorithm, the optimal solution and standard deviation in Table 4 are the averages of 30 optimization calculations for each algorithm on sixteen benchmark test functions.

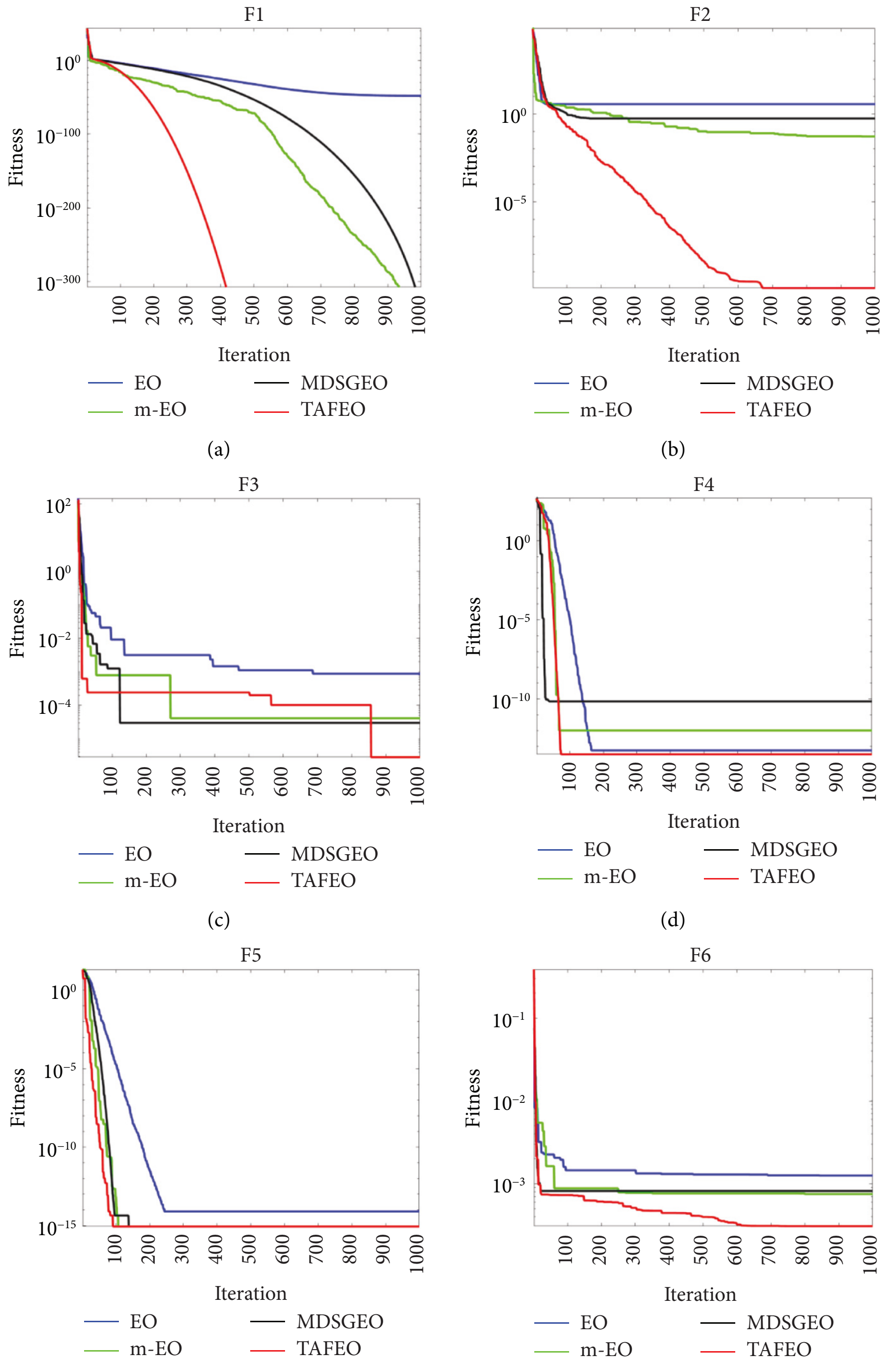
As can be seen from Table 4, the convergence accuracy of the improved TAFEO algorithm is significantly better than that of the original EO algorithm and the two improved EO algorithms in sixteen benchmark functions. The comparative analysis proves that the enhanced TAFEO algorithm in this paper is effective.

2.3.2. Performance Comparison of the Improved EO with Various Intelligent Algorithms. The TAFEO algorithm was compared with the whale algorithm (WOA) [19], the marine predator algorithm (MPA) [20], the wolf pack (GWO) algorithm [21], and the EO algorithm to verify the performance of the TAFEO algorithm.

To observe the performance of the five algorithms more clearly, sixteen benchmark test functions were chosen for the iterative graph of the simulation search, and the results are shown in Figure 2.

In Figure 2, the improved TAFEO has the highest convergence accuracy in all test functions except F12 and F16, and the convergence speed is faster than other algorithms in all test functions. Table 5 shows the results of the different algorithms.

To demonstrate the repeatability of each algorithm, the optimal solution and standard deviation in Table 5 are the averages of 30 optimization calculations for each algorithm on sixteen benchmark test functions.



(a)

(b)

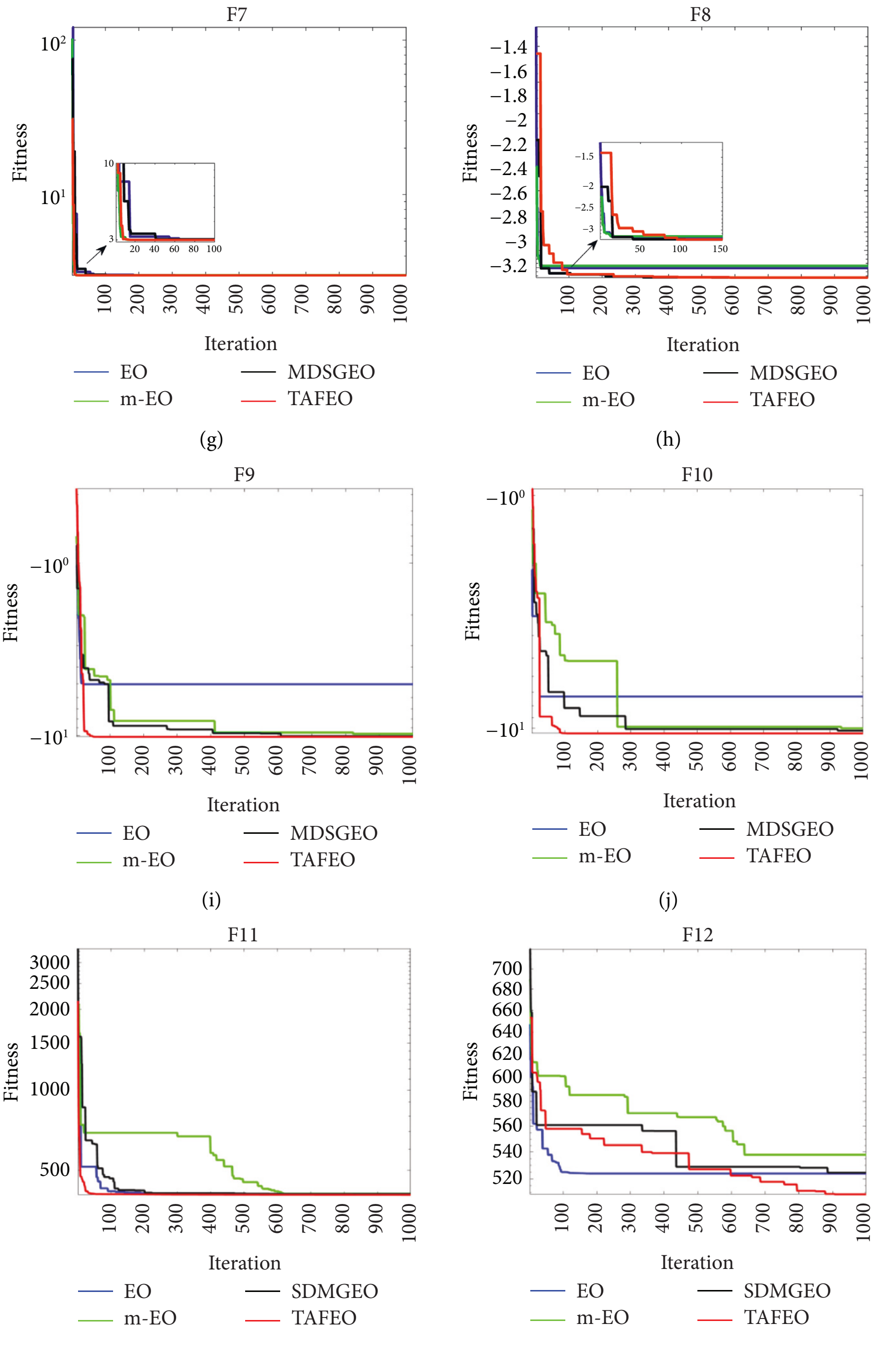
(c)

(d)

(e)

(f)

Figure 1: Continued.



(g)

(h)

(i)

(j)

(k)

(l)

Figure 1: Continued.

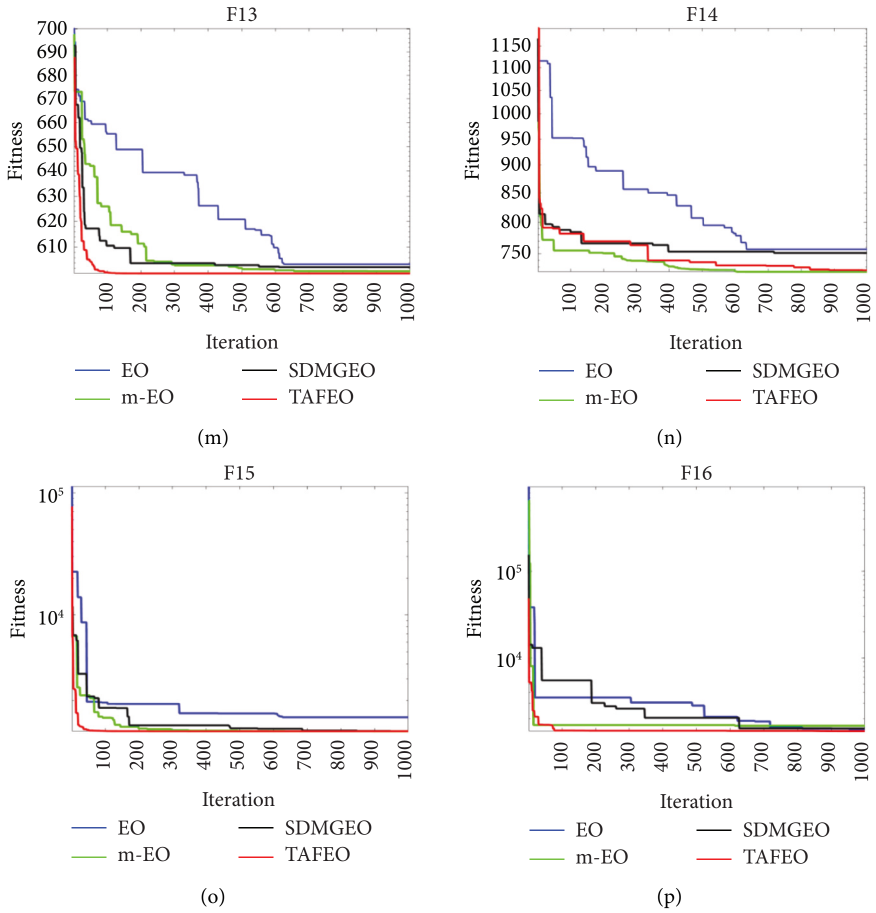


Figure 1: Simulation optimization iteration curve of 16 test benchmark functions. (a) Function F1. (b) Function F2. (c) Function F3. (d) Function F4. (e) Function F5. (f) Function F6. (g) Function F7. (h) Function F8. (i) Function F9. (j) Function F10. (k) Function F11. (l) Function F12. (m)Function F13. (n) Function F14. (o) Function F15. (p) Function F16.

As can be seen in Table 5, TAFEO outperformed the other four algorithms in terms of both search accuracy and performance stability in the benchmark function test. Therefore, it can be verified that the convergence accuracy and convergence speed of TAFEO are both higher than the other four algorithms.

In summary, the improvements to the equilibrium optimizer algorithm in this paper are highly effective and TAFEO outperforms several other intelligent algorithms tested for the benchmark functions.

2.3.3. Wilcoxon Rank-Sum Test. To observe the statistical difference between the TAFEO algorithm and other algorithms, Wilcoxon rank-sum test[22] was used to verify the results. We compare TAFEO with the EO algorithm, WOA algorithm, MP algorithm, GWO algorithm, m-EO algorithm, and MDSFEO algorithm through 18 test functions in Table 3 to verify the statistical superiority of the TAFEO algorithm. If the p-value is greater than 0.05 or NAN, it means that TAFEO is not statistically significantly different on this function. The Wilcoxon rank-sum test results are shown in Table 6.

The bold text in Table 6 indicates values greater than 5% or NAN. From Table 6, on F1, the result of the MDSGEO test is NAN, because both TAFEO and MDSGEO seek the best theoretical solution at the same time, so there is no statistical difference. There was no significant difference on F7 between TAFEO and m-EO. There was no significant difference between TAFEO and MPA on F7 and F15. There was no significant difference between TAFEO and GWO on F12 and F16.

In addition to the above, there are statistical differences between the TAFEO and other algorithms in the Wilcoxon rank-sum test, which shows that the TAFEO algorithm has great statistical advantages in the optimization results of benchmark function and verifies the robustness of the algorithm.

3. Improvements to the LSTM

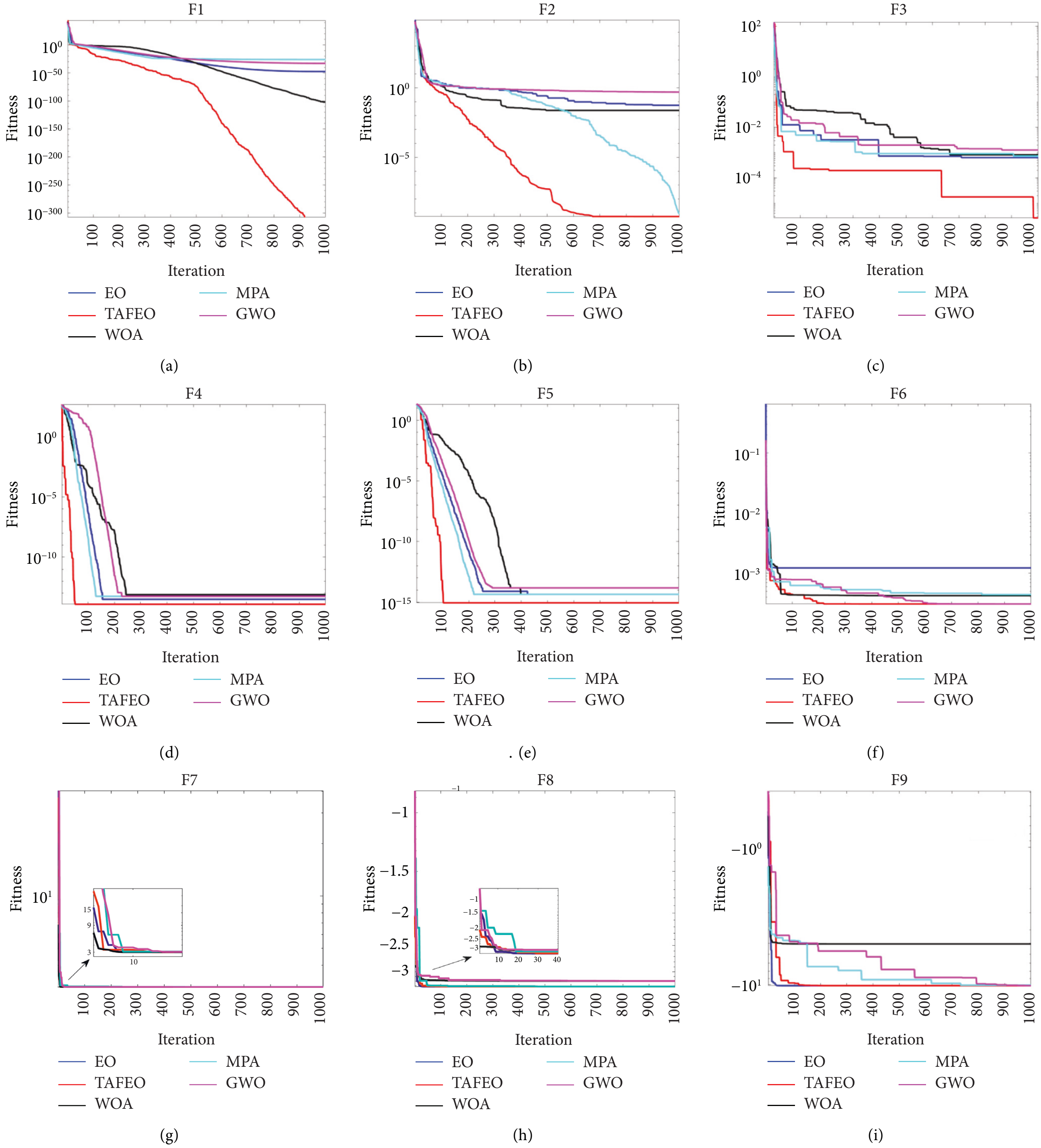
3.1. LSTM Principle. The long short-term memory (LSTM) network is a recurrent neural network responsible for computing the dependencies between observations in a time series. As such, it is commonly used for forecasting. Because logging attribute data are time-series data, this paper uses

Table 4: Test results of different algorithms.

Functions	Algorithms	Optimum solution	Standard deviation
F1	EO	4.950e − 48	8.168e − 48
	m-EO	3.911–315	1.256e − 314
	MDSGEO	0.000 + 00	0.000e + 00
	TAFEO	0.000 + 00	0.000e + 00
F2	EO	4.205e + 00	3.730e − 01
	m-EO	3.492e − 01	1.435e − 01
	MDSGEO	2.752–01	2.015–01
	TAFEO	3.293e − 09	9.608–09
F3	EO	5.467e − 04	2.592e − 03
	m-EO	6.068e − 05	5.926e − 05
	MDSGEO	5.395e − 05	4.392e − 05
	TAFEO	4.843e − 05	2.123e − 05
F4	EO	5.684e − 14	1.257e − 14
	m-EO	1.023e − 12	3.854e − 12
	MDSGEO	7.031e − 11	6.255e − 12
	TAFEO	3.215e − 14	5.665e − 15
F5	EO	5.388e − 15	1.597e − 15
	m-EO	9.613e − 16	3.236e − 16
	MDSGEO	8.881e − 16	7.556e − 15
	TAFEO	8.123e − 16	0.000e + 00
F6	EO	3.082e − 03	1.439e − 05
	m-EO	3.322e − 03	2.596e − 02
	MDSGEO	2.342e − 04	1.183e − 02
	TAFEO	1.235e − 05	1.233e − 07
F7	EO	3.000e + 00	0.000e + 00
	m-EO	3.000e + 00	0.000e + 00
	MDSGEO	3.001e + 00	3.694e − 03
	TAFEO	3.000e + 00	0.000e + 00
F8	EO	−3.264e + 00	6.378e − 02
	m-EO	−3.171e + 00	7.977e − 02
	MDSGEO	−3.296e + 00	4.939e − 02
	TAFEO	−3.304e + 00	4.342e − 02
F9	EO	−4.910e + 00	7.650e − 01
	m-EO	−8.793e + 00	2.292e + 00
	MDSGEO	−9.555e + 00	5.905e − 01
	TAFEO	−9.581e + 00	5.871e − 01
F10	EO	−5.077e + 00	9.873e − 01
	m-EO	−10.08e + 00	7.358e − 01
	MDSGEO	−10.15e + 00	6.072e − 01
	TAFEO	−10.35e + 00	3.018e − 01
F11	EO	4.231e + 02	2.523e + 01
	m-EO	4.143e + 02	3.311e + 01
	MDSGEO	4.169e + 02	2.202e + 01
	TAFEO	4.046e + 02	1.648e + 00
F12	EO	5.182e + 02	5.445e + 01
	m-EO	5.384e + 02	6.240e + 01
	MDSGEO	5.318e + 02	6.654e + 01
	TAFEO	5.121e + 02	4.701e + 01
F13	EO	6.036e + 02	1.105e + 00
	m-EO	6.006e + 02	2.452e + 00
	MDSGEO	6.021e + 02	7.086e − 01
	TAFEO	6.000e + 02	2.986e − 04
F14	EO	7.323e + 02	6.890e + 00
	m-EO	7.490e + 02	6.892e + 00
	MDSGEO	7.441e + 02	7.576e + 00
	TAFEO	7.242e + 02	6.527e + 00

Table 4: Continued.

Functions	Algorithms	Optimum solution	Standard deviation
F15	EO	$1.319e+03$	$1.431e+02$
	m-EO	$1.155e+03$	$8.634e+01$
	MDSGEO	$1.161e+03$	$7.213e+01$
	TAFEO	$1.109e+03$	$6.124e+00$
F16	EO	$1.518e+03$	$4.208e+01$
	m-EO	$1.623e+03$	$1.075e+02$
	MDSGEO	$1.625e+03$	$1.482e+02$
	TAFEO	$1.449e+03$	$1.829e+01$



(a)

(b)

(c)

(d)

. (e)

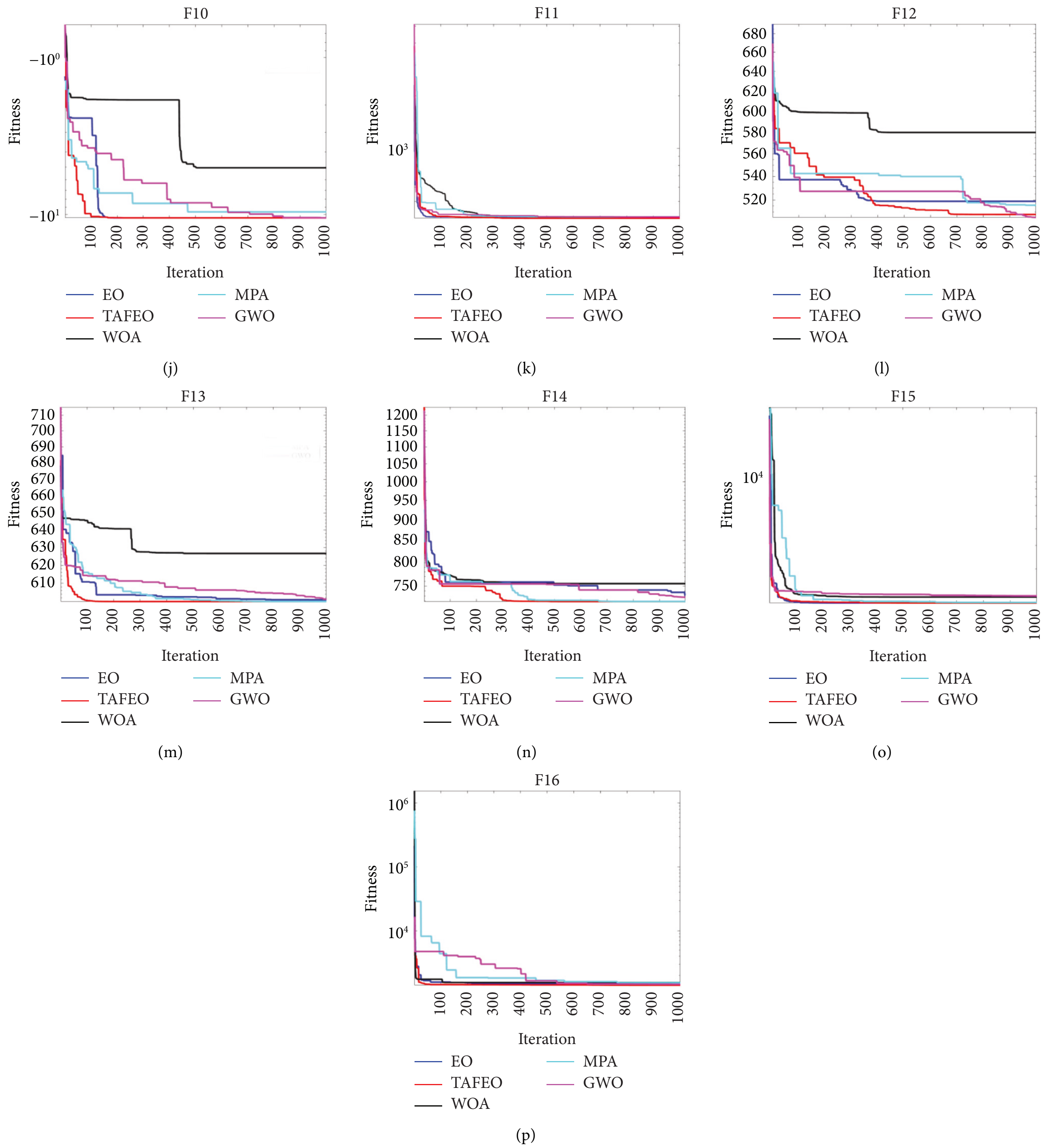
(f)

(g)

(h)

(i)

Figure 2: Continued.



(j) (k) (l)

(m) (n) (o)

(p)

Figure 2: Simulation optimization iteration curve of 16 test benchmark functions. (a) Function F1. (b) Function F2. (c) Function F3. (d) Function F4. (e) Function F5. (f) Function F6. (g) Function F7. (h) Function F8. (i) Function F9. (j) Function F10. (k) Function F11. (l) Function F12. (m) Function F13. (n) Function F14. (o) Function F15. (p) Function F16.

LSTM as a classification prediction model for reservoir identification. The cell structure of the primary LSTM neural network is shown in Figure 3.

An LSTM cell element includes the forgetting gate f_t, the input gate i_t, and the output gate o_t, which are used to protect and control.

Information i_t determines the new information being stored in the cell state, $\widetilde{C}_t$ is used to determine the updated information, and finally, the o_t gate determines the output value to the next LSTM cell element.

The equation for each variable in the LSTM network is shown in

Table 5: Test results of different algorithms.

Functions	Algorithms	Optimum solution	Standard deviation
F1	EO	2.372e − 48	4.099e − 48
	WOA	1.467e − 101	7.826e − 101
	GWO	1.830e − 33	1.377e − 33
	MPA	6.855e − 27	1.199e − 26
	TAFEO	**4.129e − 309**	**0.000e + 00**
F2	EO	2.192e + 00	1.504e + 00
	WOA	5.741e − 02	7.447e − 02
	GWO	5.227e − 01	7.467e − 03
	MPA	1.984e − 09	1.693e − 09
	TAFEO	**9.569e − 10**	**9.929e − 10**
F3	EO	6.115e − 04	3.729e − 04
	WOA	2.475e − 03	3.002e − 03
	GWO	7.951e − 04	5.157e − 04
	MPA	9.046e − 04	4.133e − 04
	TAFEO	**4.952e − 05**	**4.662e − 05**
F4	EO	3.215e − 14	2.321e − 15
	WOA	7.684e − 14	5.288e − 14
	GWO	5.625e − 14	1.734e − 14
	MPA	5.684e − 14	3.131e − 15
	TAFEO	**1.241 e − 14**	**8.654e − 16**
F5	EO	5.033e − 15	1.346e − 15
	WOA	4.204e − 15	2.627e − 15
	GWO	1.604e − 14	4.268e − 15
	MPA	4.204e − 15	9.013e − 16
	TAFEO	**8.881e − 16**	**1.256e − 16**
F6	EO	3.082e − 04	1.439e − 06
	WOA	3326e − 03	1.821e − 02
	GWO	5.034e − 03	8.604e − 03
	MPA	7.443e − 04	3.619e − 04
	TAFEO	**3.075e − 04**	**2.965e − 19**
F7	EO	**3.000e + 00**	**0.000e + 00**
	WOA	**3.001e + 00**	3.731e − 05
	GWO	**3.001e + 00**	9.491e − 14
	MPA	**3.000e + 00**	**0.000e + 00**
	TAFEO	**3.000e + 00**	**0.000e + 00**
F8	EO	−3.252e + 00	6.920e − 02
	WOA	−3.206e + 00	1.868e − 01
	GWO	−3.263e + 00	7.182e − 02
	MPA	−3.266e + 00	7.754e − 02
	TAFEO	**−3.321e + 00**	**1.529e − 15**
F9	EO	−8.628e + 00	2.369e + 00
	WOA	−8.128e + 00	2.990e + 00
	GWO	−9.646e + 00	1.546e + 00
	MPA	−9.651e + 00	5.676e − 01
	TAFEO	**−1.153e + 01**	**5.760e − 15**
F10	EO	−9.905e + 00	1.968e + 00
	WOA	−9.587e + 00	2.156e + 00
	GWO	−9.995e + 00	2.058e + 00
	MPA	−1.018e + 01	3.707e − 01
	TAFEO	**−1.015e + 01**	**1.581e − 15**
F11	EO	4.041e + 02	1.923e + 00
	WOA	4.149e + 02	2.371e + 01
	GWO	4.413e + 02	5.487e + 01
	MPA	4.198e + 02	2.541e + 02
	TAFEO	**4.000e + 02**	**1.599e − 02**

Table 5: Continued.

Functions	Algorithms	Optimum solution	Standard deviation
F12	EO	5.113e + 02	4.856e + 00
	WOA	5.165e + 02	4.753e + 00
	GWO	5.551e + 02	2.008e + 01
	MPA	**5.095e + 02**	7.990e + 00
	TAFEO	5.129e + 02	**3.401e + 00**
F13	EO	6.015e + 02	2.472e + 00
	WOA	6.005e + 02	1.591e − 01
	GWO	6.356e + 02	1.278e − 01
	MPA	6.000e + 02	**4.117e − 03**
	TAFEO	**6.001e + 02**	4.577e − 01
F14	EO	7.329e + 02	1.197e + 01
	WOA	7.317e + 02	6.667e + 00
	GWO	7.880e + 02	2.475e + 01
	MPA	7.228e + 02	**4.043e + 00**
	TAFEO	**7.217e + 02**	6.230e + 00
F15	EO	1.106e + 03	3.223e + 00
	WOA	1.169e + 03	1.028e + 00
	GWO	1.225e + 03	6.739e + 01
	MPA	1.102e + 03	5.712e + 01
	TAFEO	**1.147e + 03**	**1.28e + 00**
F16	EO	1.460e + 03	3.483e + 01
	WOA	2.442e + 03	4.952e + 03
	GWO	**1.403e + 03**	7.159e + 01
	MPA	2.411e + 03	1.583e + 03
	TAFEO	1.448e + 03	**2.150e + 00**

$$\begin{aligned} f_t &= \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right),\\ i_t &= \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right),\\ o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right),\\ \widetilde{C}_t &= \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right),\\ C_t &= f_t C_{t-1} + \widetilde{C}_t,\\ h_t &= O_t \tanh\left(C_t\right), \end{aligned} \tag{14}$$

where W_f, W_i, W_o, W_c are the weight matrix; b_f, b_i, b_o, b_c are the offset vector; σ is the sigmoid activation function and takes values in the range [0, 1]; and tanh is the tangent activation function and takes values in the range [−1, 1].

3.2. Research on the TAFEO-LSTM Model

3.2.1. Improvement Strategies. In the basic LSTM neural network, the number of neurons in the hidden layer is mainly chosen randomly or empirically, which leads to the problems of low classification accuracy and unstable classification effect. When the number of Batchsize is too large, it will increase the memory capacity and make the gradient descent direction no longer change, which will easily fall into the local optimal solution and reduce the accuracy. The size of the Maxepoch determines whether the model can be fitted or not; when the Maxepoch is too small, the model will be underfitted, and when the Maxepoch is too large, the model

Table 6: Wilcoxon rank-sum test results.

Function	P_{EO}	P_{WOA}	P_{MPA}	P_{GWO}	P_{m-EO}	P_{MDSGEO}
F1	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$	**NAN**
F2	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$
F3	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$
F4	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$	$1.212e-12$
F5	$3.020e-11$	$3.368e-05$	$3.020e-11$	$6.414e-1$	$3.020e-11$	$3.020e-11$
F6	$7.389e-11$	$1.856e-09$	$2.609e-10$	$6.695e-11$	$3.019e-11$	$9.918e-11$
F7	**NAN**	$1.608e-13$	**NAN**	$1.351e-13$	**NAN**	$6.324e-03$
F8	$2.708e-14$	$1.997e-11$	$3.941e-12$	$1.195e-13$	$3.941e-12$	$3.941e-12$
F9	$4.113e-11$	$7.226e-11$	$4.193e-02$	$4.193e-02$	$5.327e-09$	$8.223e-03$
F10	$1.193e-06$	$2.068e-02$	$2.945e-11$	$2.773e-05$	$2.945e-11$	$2.945e-11$
F11	$4.077e-11$	$3.020e-11$	$3.020e-11$	$3.020e-11$	$3.020e-11$	$3.020e-11$
F12	$2.010e-11$	$3.020e-11$	$3.182e-04$	$\mathbf{2.196e-01}$	$3.020e-11$	$3.690e-11$
F13	$5.570e-10$	$3.020e-11$	$6.548e-04$	$3.157e-10$	$4.074e-11$	$2.370e-10$
F14	$4.801e-07$	$4.203e-11$	$1.907e-02$	$1.325e-04$	$3.690e-11$	$7.389e-11$
F15	$1.067e-07$	$3.005e-03$	$\mathbf{4.077e-01}$	$2.068e-02$	$2.377e-07$	$1.173e-03$
F16	$8.484e-09$	$3.020e-11$	$8.684e-03$	$\mathbf{5.692e-01}$	$1.040e-04$	$2.157e-03$

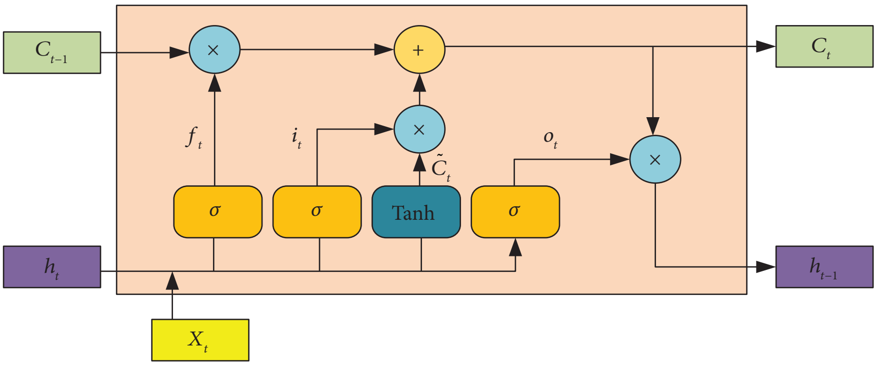
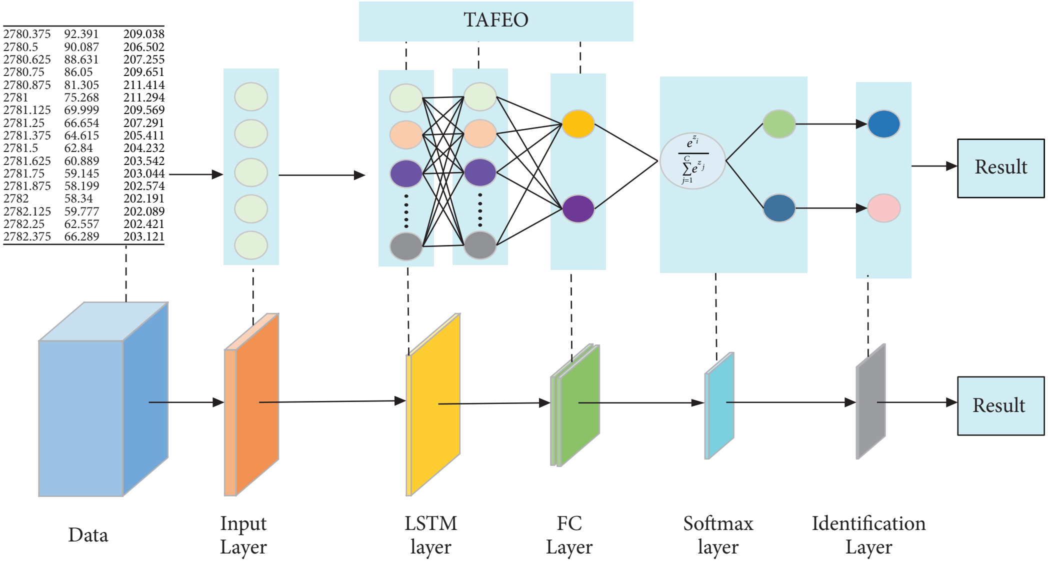


Figure 3: Cell unit structure of LSTM.



2780.375	92.391	209.038
2780.5	90.087	206.502
2780.625	88.631	207.255
2780.75	86.05	209.651
2780.875	81.305	211.414
2781	75.268	211.294
2781.125	69.999	209.569
2781.25	66.654	207.291
2781.375	64.615	205.411
2781.5	62.84	204.232
2781.625	60.889	203.542
2781.75	59.145	203.044
2781.875	58.199	202.574
2782	58.34	202.191
2782.125	59.777	202.089
2782.25	62.557	202.421
2782.375	66.289	203.121

Figure 4: Structure of the TAFEO-LSTM model.

Table 7: Algorithm steps of the TAFEO-LSTM model.

```
Algorithm: the TAFEO-LSTM model algorithm
Inputs: Training set, test set, parameters of TAFEO
Output: TAFEO-LSTM model, test set labels, accuracy
1.  Parameters for initializing the LSTM
2.  Normalized data processing
3.  Initializing the population
4.  Calculate the fitness function $f_{\text{gbest}} = 1 - 1/N \sum_{i=1}^{N} (Y_i = T_i)$ and find the current optimal solution
5.  While $t < T$
6.      For $i = i$: $N$
7.      Determine particle state and update particle position
8.      End for
9.      Updating the TAFEO-LSTM models to predict classification accuracy
10.     $t = t + 1$
11. End while
12. The optimal number of neurons with the hyperparameters batchsize and maxepoch is given to the TAFEO-LSTM model for
    retraining.
13. Building a TAFEO-LSTM model
14. Predictive classification of the test set
15. Calculate classification accuracy, AUC area, and draw ROC curves
```

Table 8: Datasets of UCI.

Datasets	Sample size	Number of features	Training set	Test set
Banknote	1000	4	700	300
Blood	700	4	490	210
Climate simulation	500	18	350	150
Indian	500	8	350	150
Pima	700	8	490	210
WDBC	500	30	350	150

Table 9: Comparison of classification results.

Data sets	LSTM	EO-LSTM	TAFEO-LSTM
Banknote	91.89	92.14	**99.43**
Blood	71.56	79.67	**96.53**
Climate simulation	72.10	73.53	**80.99**
Indian	80.71	88.08	**96.49**
Pima	78.81	80.62	**80.71**
WDBC	89.47	93.39	**94.72**

will be overfitted. Therefore, the TAFEO algorithm can be used to optimize the hyperparameters and the number of neurons of the LSTM network to enhance the classification accuracy and speed up the classification. In this paper, TAFEO is combined with the LSTM network to optimize the number of neurons in the hidden layer and the parameters batchsize and maxepoch in gradient descent of the LSTM model.

The improved TAFEO-LSTM model is shown in Figure 4.

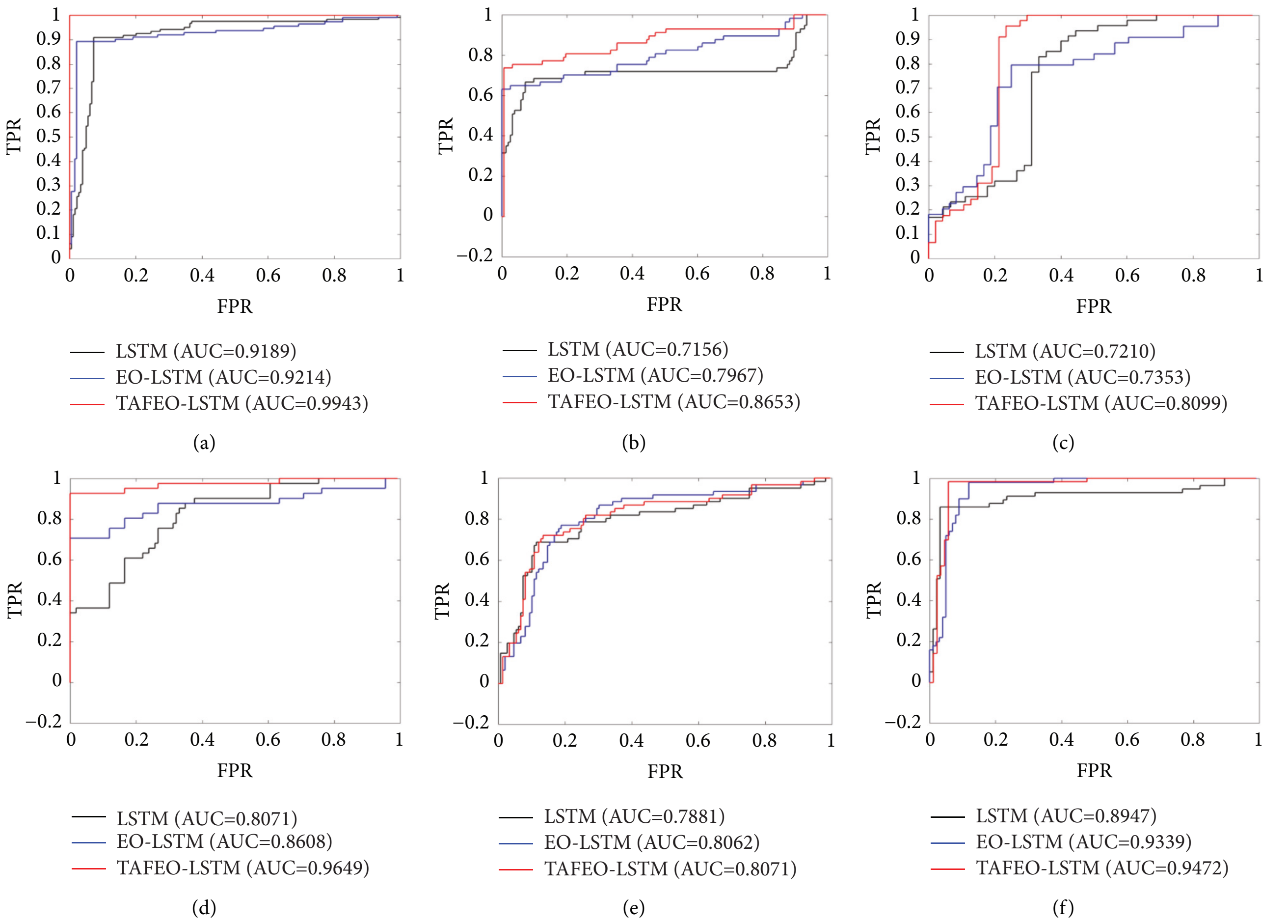
The algorithm table for the improved TAFEO-LSTM model is shown in Table 7.

3.3. UCI Dataset Simulation Experiments. To test the superiority of the improved algorithm optimization model, six sets of international general UCI binary classification datasets were selected for comparison. The comparison models include LSTM, EO-LSTM, and TAFEO-LSTM, and the six datasets are banknote, blood, climate simulation, Indian, Pima, and WDBC. The details of the six datasets are shown in Table 8.

To evaluate the performance of the model more accurately, the ROC [23] curve was introduced as an additional metric for model evaluation in addition to the accuracy.

The vertical coordinate of the ROC curve is the true-positive rate (TPR), and the horizontal coordinate is the false-positive rate (FPR). The true-positive rate represents the proportion of predicted positive samples to all positive samples, as shown in equation (15), and the false-positive rate represents the proportion of predicted positive samples to all negative samples, as shown in

$$\mathrm{TPR} = \frac{TP}{(TP + FN)}, \tag{15}$$



(a) (b) (c) (d) (e) (f)

Figure 5: ROC curves of six groups of UCI test sets. (a) Banknote. (b) Blood. (c) Climate simulation. (d) Indian. (e) Pima. (f) WDBC.

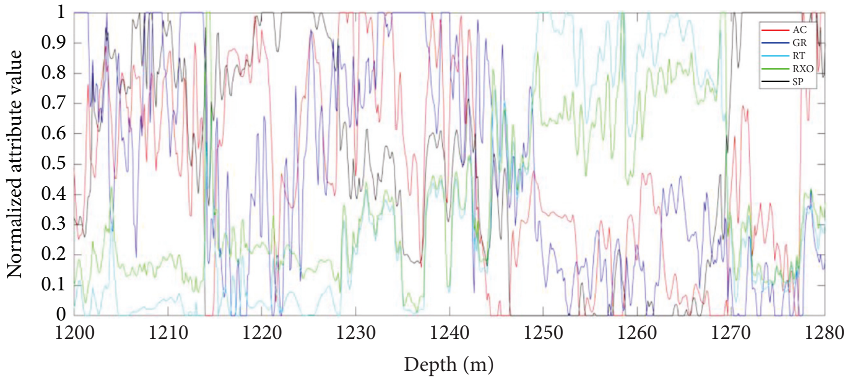


Figure 6: Attribute normalization graph of D1.

where TP is the number of positive examples of correctly classified labels, and FN is the number of negative examples of incorrectly classified labels.

$$\text{FPR} = \frac{FP}{(FP + TN)}, \tag{16}$$

where TN is the number of negative examples of correctly classified labels, and FP is the number of positive examples of incorrectly classified labels.

In the ROC curve, the performance of the model is usually evaluated by the value of AUC (area under ROC curve), which is the area under the ROC curve, and the larger the AUC value, the better the generalization performance of the model. The formula for calculating AUC is shown in

$$\text{AUC} = \frac{\sum_{i\in \text{positiveclass}} \text{rank}_i - (M(1+M)/2)}{M \times N}, \tag{17}$$

where M and N are the number of positive and negative samples, respectively.

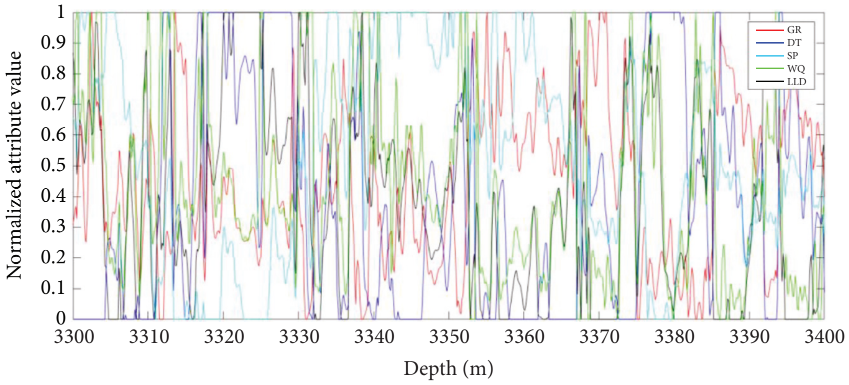


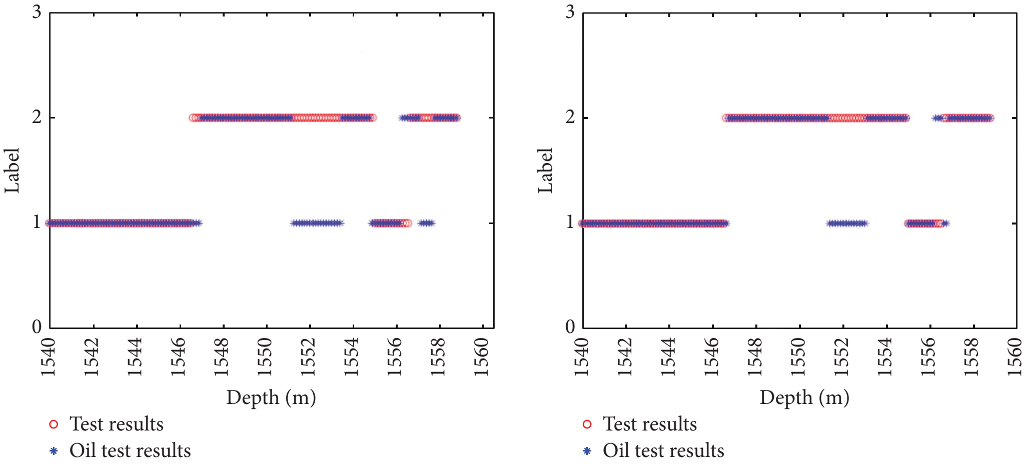
Figure 7: Attribute normalization graph of D2.

Table 10: Parameter settings in the model.

Algorithms	Number of neurons	Batchsize	Maxepoch
LSTM	150	15	18
MPA-LSTM	156	12	13
WOA-LSTM	140	13	16
GWO-LSTM	150	14	18
EO-LSTM	162	13	15
TAFEO-LSTM	141	12	15

Table 11: Data of two wells.

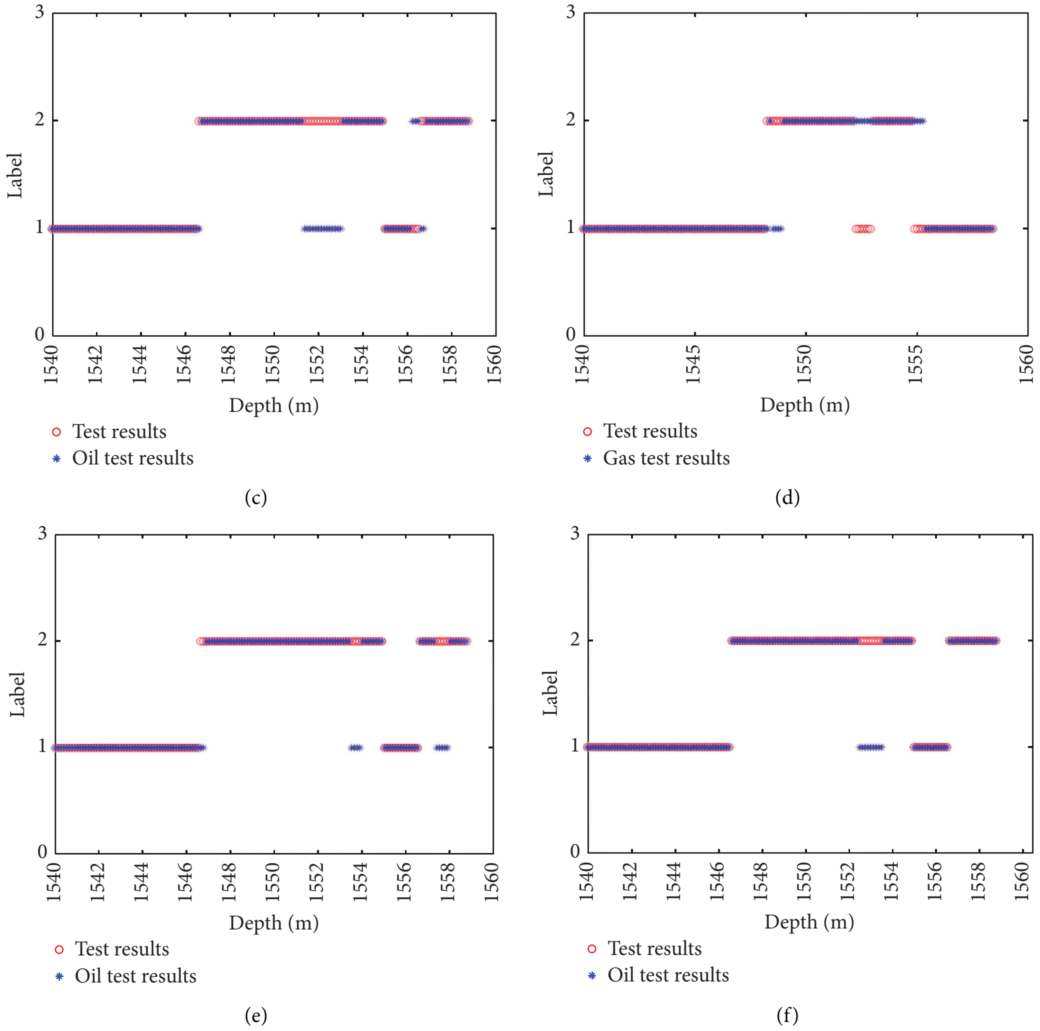
Well name		Number of data	Number of reservoir data	Number of nonreservoir data
D1	Training well section	467	141	326
	Test well section	200	70	130
D2	Training well section	1867	270	1597
	Test well section	800	172	623



(a)

(b)

Figure 8: Continued.



(c)

(d)

(e)

(f)

Figure 8: The actual oil layer distribution of D1 and classification of LSTM. (a) Classification of LSTM. (b) Classification of MPA-LSTM. (c) Classification of WOA-LSTM. (d) Classification of GWO-LSTM. (e) Classification of EO-LSTM. (f) Classification of TAFEO-LSTM.

In the simulation experiments, each UCI dataset was divided into a 70% training set and a 30% test set. The experimental results were averaged over ten experiments. The AUC values of classification results are shown in Table 9, and the ROC prediction curves are shown in Figure 5.

From Figure 5, it can be seen that the strategy of using TAFEO for the LSTM model with the number of neurons seeking and hyperparameters batchsize and maxepoch seeking is effective, and the AUC value of LSTM with the original LSTM model has been improved than after EO optimization. As can be seen from Table 8, the accuracy of the LSTM model optimized by TAFEO is also improved.

4. Logging Reservoir Identification

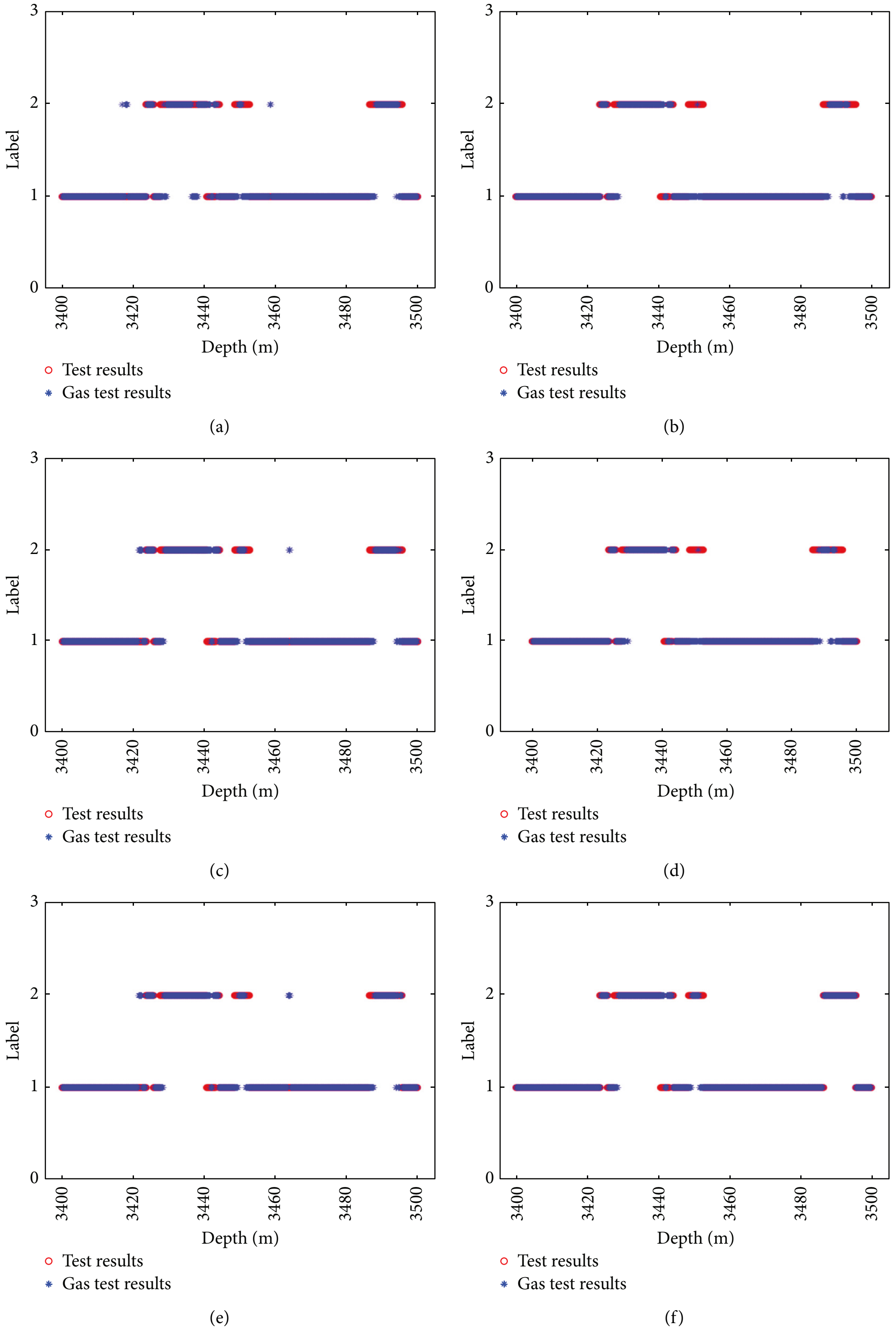
4.1. Logging Dataset. To verify the effectiveness of the TAFEO-LSTM model proposed in this paper in oil logging data mining, actual oil and gas field data (D1 and D2) were used for validation.

Table 12: Comparison of application performance for D1.

Algorithms	Accuracy (%)
LSTM	80.13
MPA-LSTM	84.77
WOA-LSTM	86.75
GWO-LSTM	90.63
EO-LSTM	92.72
TAFEO-LSTM	94.04

Table 13: Comparison of application performance for D2.

Algorithms	Accuracy (%)
LSTM	87.52
MPA-LSTM	88.89
WOA-LSTM	90.76
GWO-LSTM	87.52
EO-LSTM	91.39
TAFEO-LSTM	95.01



(a)

(b)

(c)

(d)

(e)

(f)

Figure 9: The actual gas layer distribution of D2 and classification of LSTM. (a) Classification of LSTM. (b) Classification of MPA-LSTM. (c) Classification of WOA-LSTM. (d) Classification of GWO-LSTM. (e) Classification of EO-LSTM. (f) Classification of TAFEO-LSTM.

The D1 well was attribute reduced to obtain five attributes, namely, (AC, GR, RT, RXO, and SP), and the D2 well was attribute reduced to obtain 13 attributes, namely, (GR, DT, SP, WQ, LLD, LLS, DEN, NPHI, PE, U, TH, K, and CALI). The attributes of D1 and D2 wells were applied in the testing process after attribute simplification, and the data in the training well segment were divided into 70% training set and 30% test set. The attribute information from the D1 and

Table 14: Results of reservoir identification and comparison of models.

Model Name	Accuracy
Fisher different approach	76.00%
BP neural network	85.75%
ELM	91.13%
SVM	87.75%
CNN	93.25%
TAFEO-LSTM	95.01%

D2 wells was selected for normalization, and the five main attributes from each of the D1 and D2 wells were selected to draw their logging curves, as shown in Figures 6 and 7.

To verify the performance of the TAFEO-LSTM model, five models, LSTM, EO-LSTM, MPA-LSTM, WOA-LSTM, and GWO-LSTM, are constructed for comparison experiments. The parameter settings in each model are shown in Table 10.

Information on the data from the two selected wells is shown in Table 11.

4.2. Reservoir Identification Results. Figure 8 shows the actual formation results for each algorithm model in the test well section of well D1 compared to the test oil results, where vertical coordinate 2 represents the oil formation and vertical coordinate 1 represents the nonoil formation.

From Figure 8, the TAFEO-LSTM model has higher accuracy and the classification results can be closer to the actual oil layer distribution than the LSTM, EO-LSTM, MPA-LSTM, WOA-LSTM, and GWO-LSTM, and the accuracy was used to assess the model performance. The recognition accuracy was selected as the average value of each algorithm after 30 runs. The results are shown in Table 12.

From Table 11, the classification performance of the LSTM optimized by applying the intelligent algorithm has significantly improved, in which the recognition accuracy of the oil layer can reach 94.04% by applying the TAFEO-LSTM model, which indicates that it is feasible to apply TAFEO to optimize the LSTM and identify the oil layer with significant effect.

Similar to D1, accuracy was used to assess model performance and the average of results after 30 runs is shown in Table 13.

From Table 13, the classification performance of the LSTM optimized by applying the intelligent algorithm is significantly improved, in which the recognition accuracy of the gas layer can reach 95.01% by applying the TAFEO-LSTM model, which indicates that it is feasible to apply the TAFEO-optimized LSTM and the recognition of the gas layer is significant.

Figure 9 shows the actual gas formation results for each algorithm model in the test well section of well D2 compared to the test gas results, where vertical coordinate 2 represents the gas formation and vertical coordinate 1 represents the nongas formation.

As can be seen from Figure 9, compared with LSTM, EO-LSTM, WOA-LSTM, GWO-LSTM, and MPA-LSTM, the TAFEO-LSTM model has a higher accuracy rate and the classification results can be closer to the real gas layer distribution.

4.3. Comparison of Reservoir Identification Models. To better verify the validity of the improved model, several models that have been applied to logging are compared with those proposed in this paper. These models are Fisher's different approach, BP neural network, ELM [24], SVM [25], and CNN [26]. We use these models to identify well D2 logging data. The comparison results are shown in Table 14.

From Table 14, the TAFEO-LSTM model is more accurate in identifying reservoirs than other models because it takes into account the temporal characteristics of logging data.

In summary, the improved TAFEO strategy applying LSTM classification is effective in practical logging reservoir identification.

5. Conclusions

(1) An improved equilibrium optimizer algorithm, TAFEO, is proposed, and tent mapping is introduced to increase the population diversity. The introduction of a convergence factor is to effectively accelerate the convergence speed of the algorithm and balance the local and global optimization-seeking ability, and finally, adding a premature perturbation strategy can prevent the algorithm from falling into the local optimum. Through the simulation of 16 benchmark functions and Wilcoxon rank-sum test for optimization results, the improved algorithm outperforms various intelligent optimization algorithms in terms of accuracy and convergence speed and has good robustness.

(2) The TAFEO algorithm is applied to the LSTM parameter optimization; then, a reservoir identification TAFEO-LSTM model is established. Simulation experiments on the UCI dataset demonstrate that the improved model has a strong generalization ability and high recognition rate. The TAFEO-LSTM model is then applied to identify the reservoirs, and the results are compared with those of five reservoir identification models, namely, LSTM, EO-LSTM, WOA-LSTM, GWO-LSTM, and MPA-LSTM. The results show that the proposed TAFEO-LSTM model is more accurate than the other five models in identifying reservoirs, with the highest accuracy of 94.04% for the oil layer and 95.01% for the gas layer. Subsequently, TAFEO-LSTM still performs well compared with the other models used for reservoir identification. Obviously, the improved model is effective in reservoir identification and has broad application prospects.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest in the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 42075129), the Hebei Province Natural Science Foundation (no. E2021202179), and the Key Research and Development Project of Hebei Province (nos. 19210404D, 20351802D, and 21351803D).

References

[1] S. Yu, K. Zhu, and F. Diao, "A dynamic all parameters adaptive BP neural networks model and its application on oil reservoir prediction," *Applied Mathematics and Computation*, vol. 195, no. 1, pp. 66–75, 2008.

[2] K. Aminian and S. Ameri, "Application of artificial neural networks for reservoir characterization with limited data," *Journal of Petroleum Science and Engineering*, vol. 49, no. 3-4, pp. 212–222, 2005.

[3] L. Han, L. Fuqiang, D. Zheng, and X. Weixu, "A lithology identification method for continental shale oil reservoir based on BP neural network," *Journal of Geophysics and Engineering*, vol. 15, no. 3, pp. 895–908, 2018.

[4] X. Zhou, Z. Zhang, and C. Zhang, "Bi-LSTM deep neural network reservoir classification model based on the innovative input of logging curve response sequences," *IEEE Access*, vol. 2021, no. 99, p. 1, 2021.

[5] W. Chen, L. Yang, B. Zha, M. Zhang, and Y. Chen, "Deep learning reservoir porosity prediction based on multilayer long short-term memory network," *Geophysics*, vol. 85, no. 4, pp. WA213–WA225, 2020.

[6] H. Xie, L. Zhang, and C. P. Lim, "Evolving CNN-LSTM models for time series prediction using enhanced grey wolf optimizer," *IEEE Access*, vol. 8, pp. 161519–161541, 2020.

[7] L. Peng, Q. Zhu, S. X. Lv, and L. Wang, "Effective long short-term memory with fruit fly optimization algorithm for time series forecasting," *Soft Computing*, vol. 24, no. 19, pp. 15059–15079, 2020.

[8] Z. Yang and C. Wei, "Prediction of equipment performance index based on improved chaotic lion swarm optimization–LSTM," *Soft Computing*, vol. 24, no. 13, pp. 9441–9465, 2020.

[9] A. A. A. Mohamed, Y. S. Mohamed, A. A. M. El-Gaafary, and A. M. Hemeida, "Optimal power flow using moth swarm algorithm," *Electric Power Systems Research*, vol. 142, pp. 190–206, 2017.

[10] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, Article ID 114864, 2021.

[11] A. Iman, A. A. Heidari, N. Saeed, H. Chen, and H. G. Amir, "INFO: an efficient optimization algorithm based on weighted mean of vectors," *Expert Systems with Applications*, vol. 195, Article ID 116516, 2022.

[12] A. Faramarzi, M. Heidarinejad, B. Stephens, and S Mirjalili, "Equilibrium optimizer: a novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, Article ID 105190, 2020.

[13] J. Wang, B. Yang, D. Li et al., "Photovoltaic cell parameter estimation based on improved equilibrium optimizer algorithm," *Energy Conversion and Management*, vol. 236, Article ID 114051, 2021.

[14] Q. Fan, H. Huang, K. Yang, S. Zhang, L. Yao, and Q. Xiong, "A modified equilibrium optimizer using opposition-based learning and novel update rules," *Expert Systems with Applications*, vol. 170, Article ID 114575, 2021.

[15] Z. Fu, P. Hu, W. Li, J. S. Pan, and S. C. Chu, "Parallel equilibrium optimizer algorithm and its application in capacitated vehicle routing problem," *Intelligent Automation & Soft Computing*, vol. 27, no. 1, pp. 233–247, 2021.

[16] S. Gupta, K. Deep, and S. Mirjalili, "An efficient equilibrium optimizer with mutation strategy for numerical optimization," *Applied Soft Computing*, vol. 96, Article ID 106542, 2020.

[17] H. Shouyu, H. Qing, and J. Chen, "Application of improved equilibrium optimizer algorithm to constrained optimization problems," *Frontiers of Computer Science and Technology*, vol. 202114 pages, 2021.

[18] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization," 2016.

[19] S. Mirjalili and A. Lewis, "The Whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[20] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H Gandomi, "Marine predators algorithm: a nature-inspired metaheuristic," *Expert Systems with Applications*, vol. 152, Article ID 113377, 2020.

[21] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[22] F. . Wilcoxon, "Individual comparison by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.

[23] S. Pérez-Fernández, P. Martínez-Camblor, P. Filzmoser, and N Corral, "Visualizing the decision rules behind the ROC curves: understanding the classification process," *AStA Advances in Statistical Analysis*, vol. 105, no. 1, pp. 135–161, 2021.

[24] L. I. Min, K. Chen, and Z. Yang, "Complicated lithology identification in heavy oil reservoir based on pattern recognition," *Well Logging Technology*, vol. 2017, 2017.

[25] J. C. Kuang and Z. Wang, "A PLS-SVM model for reservoir identification in natural gas exploration," in *Proceedings of the International conference on partial least squares and related methods;PLS'09*, Chengdu University of Technology, Chengdu, China, February 2009.

[26] T. Li, Z. Wang, R. Wang, Y. Wang, and N. Yiu, "Pore type identification in carbonate rocks using convolutional neural network based on acoustic logging data," *Neural Computing & Applications*, vol. 33, no. 1, pp. 4151–4163, 2020.