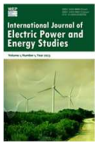
WEP Warwick Evans Publishing

International Journal of Electric Power and Energy Studies

ISSN: 3006-2004 (Print), ISSN: 3006-0826 (Online)
Volume 5, Number 3, Year 2026
DOI: https://doi.org/10.62051/ijepes.v5n3.01



Air Quality Prediction Using a Multi-strategy Improved Moss Gross Optimization Algorithm for Optimizing BiGRU Data Fusion

Wenzhe Zhang

College of Electrical Information Engineering, Northeast Petroleum University, Daqing 163318, China

ABSTRACT

The Air Quality Index (AQI) serves as a crucial indicator of dynamic trends in air pollution. However, achieving high prediction accuracy remains challenging due to the complexity of the atmospheric system, which is driven by intricate interactions between air pollutants and meteorological conditions. To address this and achieve precise air quality forecasting, this paper proposes a multi-strategy Improved Moss Growth Optimization (IMGO) algorithm designed to optimize a Bidirectional Gated Recurrent Unit (BiGRU) network. First, strategies including Fitness-Distance Balance (FDB), adaptive t-distribution, and Opposition-Based Learning (OBL) are integrated to enhance the optimization precision and convergence speed of the algorithm. Evaluations on six standard benchmark functions-supported by statistical results, convergence curves, and box plots-demonstrate that IMGO exhibits superior global optimization capabilities compared to the standard MGO. Second, IMGO is utilized to optimize the hyperparameters of the BiGRU model, significantly bolstering its stability. Finally, precise data fusion is accomplished leveraging the nonlinear mapping capability of the optimized BiGRU. The proposed IMGO-BiGRU model, alongside standard BiGRU and MGO-BiGRU, was applied to AQI prediction tasks. Experimental results indicate that IMGO-BiGRU achieves a high prediction accuracy of 0.982, with a Mean Absolute Error (MAE) of 9.710, a Mean Absolute Percentage Error (MAPE) of 8.107%, and a Root Mean Square Error (RMSE) of 17.660. These findings confirm that the proposed model delivers high precision and robust error reduction in air quality forecasting.

KEYWORDS

IMGO algorithm; Data fusion; BiGRU neural network; FDB strategy.

1. INTRODUCTION

Air pollution is widespread in many countries and cities and has caused numerous health problems. There are six major types of air pollutants, and the Air Quality Index (AQI) is commonly introduced to describe the dynamic trend of air pollution [1] . However, because AQI is random and non-stationary, its prediction often suffers from low accuracy and poor stability. In addition, the atmosphere is a complex dynamic system whose evolution is easily affected by pollutant concentrations and multiple meteorological factors [2]. Traditional air-quality prediction models, such as the autoregressive integrated moving average model [3],and the autoregressive moving average model [4] , are not well suited to accurately predicting nonlinear and non-stationary indices such as AQI. Common data-fusion methods, including Bayesian inference [5] , Kalman filtering [6] , and clustering [7] , are also limited in handling high-dimensional complex data and are sensitive to noisy observations [8].

Content from this work may be used under the terms of CC BY-NC 4.0 licence (https://creativecommons.org/licenses/by-nc/4.0/).
Published by Warwick Evans Publishing.

In recent years, with the development of digital intelligence technologies, data-fusion algorithms and deep-learning methods have provided new ideas for accurate AQI prediction. Kalantari et al. [9] applied recurrent neural networks (RNNs) to AQI prediction, but RNNs suffer from gradient vanishing and gradient explosion. Das et al.[10] used long short-term memory (LSTM) networks and RNNs to predict air-pollutant concentrations, and the results showed that LSTM outperformed RNNs and alleviated the gradient problem, although its model structure is relatively complex. Song et al. [11] employed a BiGRU model, whose simple architecture and bidirectional GRU structure enable more complete exploitation of contextual information and thus improve prediction accuracy, but its hyperparameters still need to be set manually. Because air-quality prediction involves high-dimensional and complex atmospheric data, metaheuristic algorithms are usually adopted to search for optimal parameters. Singh et al. [12] addressed LSTM hyperparameter optimization with a genetic algorithm, but the convergence speed was slow. Huang et al. [13] proposed using the sparrow search algorithm to determine GRU hyperparameters for AQI prediction. Although the method converges quickly and yields high prediction accuracy, its optimization ability still has room for improvement.

The moss growth optimization (MGO) algorithm was proposed by Zheng et al. [14] based on the growth behavior of moss. Its dual reproduction search and cryptobiosis mechanism enhance flexibility and help avoid local optima, but its convergence accuracy and speed are still unsatisfactory when dealing with high-dimensional, nonlinear, and complex atmospheric-environment data. Therefore, this paper proposes a multi-strategy improved moss growth optimization algorithm (IMGO) to optimize a bidirectional gated recurrent unit (BiGRU) data-fusion model. By exploiting the efficient search capability of IMGO for hyperparameter optimization and the advantages of BiGRU in terms of a small number of parameters and strong large-scale data-processing capability, the proposed data-fusion algorithm improves the accuracy and stability of AQI prediction.

2. MULTI-STRATEGY IMPROVED MOSS GROWTH OPTIMIZATION ALGORITHM

2.1. Moss Growth Optimization Algorithm

The MGO algorithm is inspired by the wind-driven dispersal of moss in nature and introduces a wind-direction determination mechanism to define the evolutionary direction of the population and partition the population. The most distinctive individual in the population is M_{best}, j is taken as the threshold. The formula for selecting the subpopulation with the larger number of individuals is given as follows:

$$divX_j = \begin{cases} DX_{j1}, count\left(DX_{j1}\right) \geq count\left(DX_{j2}\right) \\ DX_{j2}, count\left(DX_{j1}\right) < count\left(DX_{j2}\right) \end{cases} \tag{1}$$

where X denotes the moss population, j denotes the dimensionality of the problem to be solved, DX_{j2} and DX_{j2} denote the sets of individuals greater than and less than the threshold M_{best}, respectively, $count()$ is the function used to calculate the number of moss individuals in a set, and it denotes the subpopulation with the larger number of individuals.

The wind direction is calculated as follows:

$$D_wind = \frac{1}{num}\sum_{i=1}^{num} dM_i, dM_i \in dirX \tag{2}$$

where D_wind denotes the wind-direction calculation, M_i denotes the i th moss individual, num denotes the total number of individuals in $dirX$, and $dirX$ denotes the set of distances among individuals in region $divX$.

Meanwhile, two new search strategies, namely spore dispersal search and dual reproduction search, are proposed for exploration and exploitation, respectively, as follows:

$$M_i^{\text{new}} = \begin{cases} M_i + step1 \cdot D_, r_1 > d_1 \\ M_i + step2 \cdot D_, r_1 \le d_1 \end{cases} \tag{3}$$

$$step1 = w \cdot (r_2 - 0.5) \cdot E \tag{4}$$

$$step2 = 0.1w \cdot (r_3 - 0.5) \cdot E \cdot \left[1 + \frac{E}{2} \cdot \left(1 + \tanh\left(\beta \sqrt{1 - \beta^2} \right) \right) \right] \tag{5}$$

where M_i^{new} denotes the new moss individual generated by the spore, $step1$ denotes the dispersal distance of spores under a stable wind direction, $step2$ denotes the dispersal distance of spores under a turbulent wind direction, r_1, r_2, and r_3 are random numbers in (0, 1), d_1 is a constant, E is the wind force, and β is the proportion of individuals in $divX$ relative to those in X.

Dual reproduction search generates new individuals around the original individual through sexual reproduction and vegetative reproduction. It changes only the proportion of a single dimension of an individual, thereby enhancing the overall local exploration capability. The dual reproduction search is formulated as follows:

$$\begin{cases} M_i^{\text{new}} = (1 - act) \cdot M_i + act \cdot M_{\text{best}}^{\text{new}}, & r_4 > d_2 \\ M_{i,j}^{\text{new}} = M_{\text{best},j} + step3 \cdot D_wind_j, & r_4 \le d_2 \end{cases} \tag{6}$$

$$step3 = 0.1 \cdot (r_5 - 0.5) \cdot E \tag{7}$$

where j is a random number not exceeding the maximum dimension of an individual, $M_{i,j}^{\text{new}}$ denotes the M_i^{new} th particle in j, M_{best} denotes the current best individual, $D_{\text{wind},j}$ denotes the wind direction of particles in generation j, $step3$ denotes the distance of dual reproduction search, act is used to evaluate whether the particles in M_{best} have been sufficiently exploited, r_4 and r_5 are random numbers in (0, 1), and d_2 is a constant parameter set to 0.5 in this study.

Finally, based on the cryptobiosis phenomenon of moss, a cryptobiosis mechanism is introduced. The moss individuals generated in each iteration are recorded, and the maximum number of records is set to rec = 10. When the maximum record number is reached or the iteration process ends, the best individual is restored and replaces the current individual M_i.

2.2. Fitness-Distance Balance Strategy for Individual Iteration Optimization

In the individual-update stage of the MGO algorithm, the calculation of two step sizes increases the computational cost and may slow convergence in the later stages of optimization. In this paper, the

fitness-distance balance (FDB) method developed by Kahraman et al. [15] is adopted because it can stably and rapidly identify the individuals that contribute most to the search process. By balancing the fitness of a solution and its distance from the current best solution and dynamically adjusting their weights, FDB simultaneously pursues high-quality solutions and maintains diversity, thereby improving global search capability, avoiding premature convergence, and enhancing the ability to escape local optima. The FDB strategy is applied to the individual-update stage of MGO. Candidate individuals are selected through the FDB function according to the balance between fitness and inter-individual distance and are then used for subsequent position mutation and updating. The formula is given as follows:

$$\forall M_i, \quad D_{x,i} = \sqrt{\begin{aligned}&\left(M_{i[1]} - M_{\text{best}[1]}\right)^2 + \left(M_{i[2]} - M_{\text{best}[2]}\right)^2 \\ &+ \cdots + \left(M_{i[m]} - M_{\text{best}[m]}\right)^2\end{aligned}} \tag{8}$$

$$\forall x_i, \quad S_{x_{[i]}} = w \cdot \text{norm}F_{[i]} + (1-w) \cdot \text{norm}D_{x_{[i]}} \tag{9}$$

where $D_{x,i}$ denotes the distance between candidate M_i and the current best solution M_{best} in the population, $S_{x_{[i]}}$ is the FDB score of an individual, w is the weight coefficient, $\text{norm}F_{[i]}$ and $\text{norm}D_{x_{[i]}}$ are the normalized fitness value and normalized distance value, respectively.

2.3. Adaptive t-Distribution Mutation Operator Strategy

As can be seen from Eqs. (3) and (7), step-size adjustment mainly depends on random numbers and the wind-force coefficient E, making it difficult to dynamically balance global exploration and local exploitation. To improve the search efficiency of the algorithm at different evolutionary stages, this paper introduces an adaptive t-distribution mutation operator. By recording the current iteration number to dynamically adjust the distribution form, the method enhances the ability to escape local optima and improves convergence accuracy. The t distribution [16] contains the degree-of-freedom parameter n, and the standard Gaussian distribution and the Cauchy distribution are two boundary special cases of the t distribution. Cauchy mutation has strong global search capability and can effectively maintain population diversity, whereas the Gaussian distribution has strong local exploitation capability and guarantees the convergence speed in the later evolutionary stage. The distribution curves of the three functions are shown in Fig. 1(a).

A t-distribution mutation operator that uses the iteration number $iter$ as the degree-of-freedom parameter is employed to perturb the solution position. During each iteration, a t-distribution random number is generated according to the current iteration count to construct the mutation operator. In the early stage of the algorithm, $iter$ is small, and the t distribution resembles the Cauchy distribution, producing a larger perturbation step size that guides particles to explore a wider range and prevents premature convergence. Subsequently, the generated adaptive t-distribution operator is used to perturb the position of the current best individual M_{best}, thereby generating a new candidate solution. The position-update formula is as follows:

$$M_{best,j}^{new} = M_{best,j} + M_{best,j} \cdot t(iter) \tag{10}$$

where $M_{best,j}^{new}$ is the position of the best solution after adaptive t-distribution mutation perturbation in the j dimension, and $M_{best,j}$ is the position of the original best solution in the j dimension.

As iterations progress, the gradual increase of $iter$ causes the t-distribution to converge toward a Gaussian distribution. Consequently, the perturbation step size diminishes, and the mutation range becomes increasingly concentrated within the proximity of the current global optimum. This mechanism facilitates a seamless transition from global exploration to local exploitation, thereby enhancing the model's predictive stability when addressing non-stationary AQI data.

2.4. Opposition-Based Learning Optimization Strategy

In the MGO algorithm, the sexual reproduction mechanism generates new solutions through a linear combination of an individual and the current best solution. This mechanism can easily cause M_{best} to fall into a local optimum, leading to stagnation of population exploitation. Therefore, an opposition-based learning (OBL) strategy is introduced in the sexual reproduction stage. Its core idea is to generate the opposite solution of an arbitrary candidate solution simultaneously and retain the better one according to the fitness value for the next iteration, thereby accelerating the search process and enhancing global escape ability[17] . Assuming an dim -dimensional search space, the position variable of the i th moss individual in the j th dimension is $M_{i,j}$, the search boundaries are $[LB_j, UB_j]$, and after a new individual is generated by sexual reproduction, its opposite solution $M_{i,j}^*$ is immediately computed as follows:

$$M_{i,j}^* = LB_j + UB_j - M_{i,j} \tag{11}$$

The fitness values of the two solutions are then compared. Let $f(M_{i,j})$ and $f(M_{i,j}^*)$ denote the fitness values of $M_{i,j}$ and $M_{i,j}^*$, respectively. When $f(M_{i,j}) > f(M_{i,j}^*)$, $M_{i,j}$ is selected for further optimization; otherwise, $M_{i,j}^*$ is selected. After each iteration, the search space is reduced by half, thereby accelerating the search process.

2.5. IMGO Algorithm Procedure

The proposed algorithm sequentially incorporates the FDB strategy, adaptive t-distribution mutation, and opposition-based learning strategy, which effectively enhance global and local search capability, accelerate convergence, increase solution diversity, and improve the ability to escape local optima. The flowchart of IMGO is shown in Fig. 2(a).

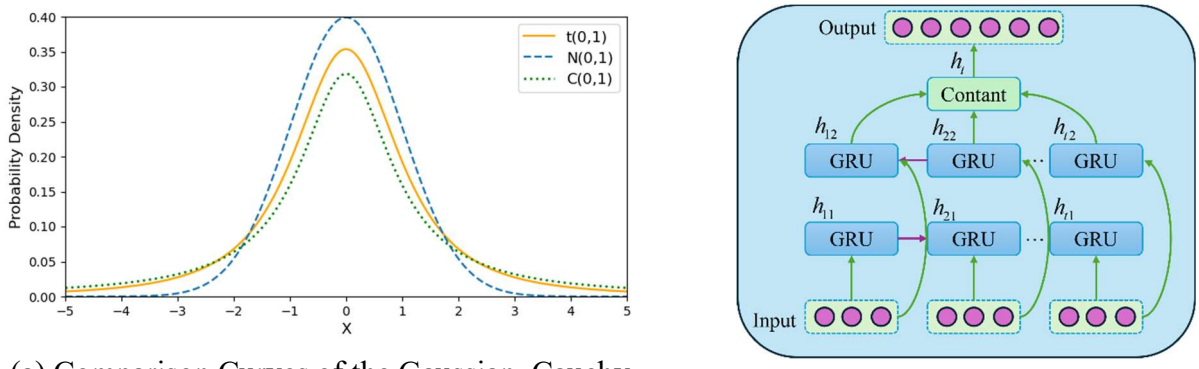
2.6. IMGO-Optimized Bidirectional Gated Recurrent Unit

BiGRU combines a bidirectional recurrent structure with a gating mechanism. In AQI prediction, it can process sequence data in both the forward and backward directions. This bidirectional structure enables BiGRU to capture long-range dependencies in the sequence and thus improves model performance [18]. In the GRU network[19], the reset gate r_t and the update gate z_t are used to alleviate gradient explosion and gradient vanishing. Compared with LSTM, GRU has a simpler structure, fewer parameters, and faster computation, but a unidirectional GRU may lead to partial information loss[20] . Therefore, a bidirectional BiGRU is adopted, and its structure is shown in Fig. 1(b).

The BiGRU equations are as follows:

where h_{t1} is the forward hidden state, h_{t2} is the backward hidden state, h_t denotes the concatenation of the forward and backward hidden states, and x_t is the input at time t.

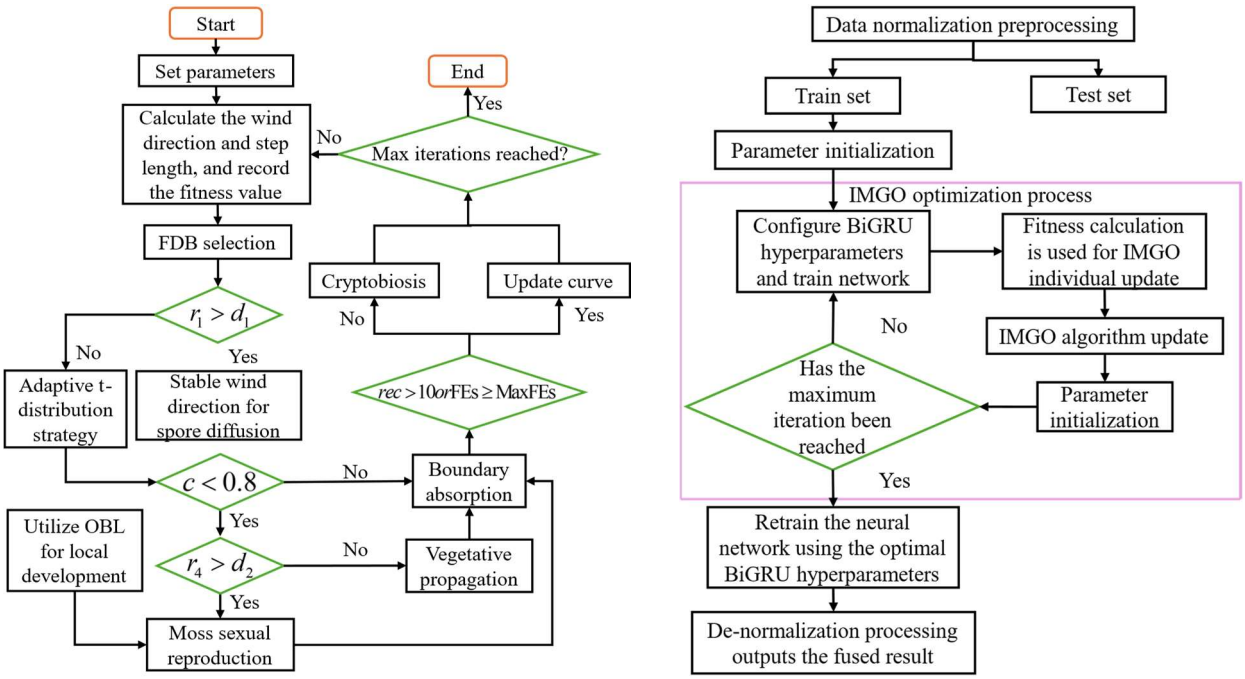
A key challenge in constructing a high-accuracy BiGRU model lies in determining hyperparameters such as the number of hidden-layer neurons, the learning rate, and the batch size. Manual tuning is time-consuming and inefficient. Therefore, an IMGO-BiGRU data-fusion model is constructed in this study, in which IMGO is used to automatically optimize the BiGRU hyperparameters. The corresponding flowchart is shown in Fig. 2(b).



(a) Comparison Curves of the Gaussian, Cauchy, and t Distributions

(b) BiGRU structure

Fig. 1 Distribution Comparison Curves and the BiGRU structure



(a) IMGO Algorithm

(b) IMGO-BiGRU Flowchart

Fig. 2 Flowcharts of the IMGO Algorithm and the IMGO-BiGRU Optimization Procedure

3. EXPERIMENTAL RESULTS AND ANALYSIS

3.1. Validation of the IMGO Algorithm

3.1.1. Experimental Setup and Test Description

To verify that the proposed IMGO algorithm possesses stronger optimization capability and stability, six classical benchmark functions were selected for simulation experiments. The test functions are

listed in Table 1. F1-F3 are unimodal benchmark functions, whereas F4-F6 are multimodal benchmark functions. Such functions contain multiple local optima, which interfere with the optimization process and therefore require stronger ability to escape local optima. The optimal value of each test function is 0, and the dimensionality of the functions is 30.

In the experiments, each algorithm was independently run 30 times, and the minimum value and mean value were used to characterize the convergence capability and optimization accuracy, while the standard deviation was used to measure stability. The maximum number of iterations for each algorithm was set to 500, and the population size N was 50 for all methods. The simulation environment was as follows: Win11 64-bit operating system, Intel(R) Core i5-12400F CPU at 2.50 GHz, 16 GB memory, NVIDIA GeForce GTX4060 8 GB GPU, and Matlab R2023b as the programming environment.

Table 1. Benchmark Functions

Fuction	Name	Dimension	Search Range	Optimum
F1	Sphere Function	30	[-100,100]	0
F2	Schwefel's Problem 1.2	30	[-100,100]	0
F3	Schwefel's Problem 2.21	30	[-100,100]	0
F4	Ackley's Function	30	[-32,32]	0
F5	Generalized Penalized Function 1	30	[-50,50]	0
F6	Generalized Penalized Function 2	30	[-50,50]	0

3.1.2. Convergence Analysis

The test results of the benchmark functions are listed in Table 2, where the best results are shown in bold. Figure 3 presents the fitness convergence curves of the algorithms on different benchmark functions, and Fig. 4 shows the boxplots of the six optimization algorithms.

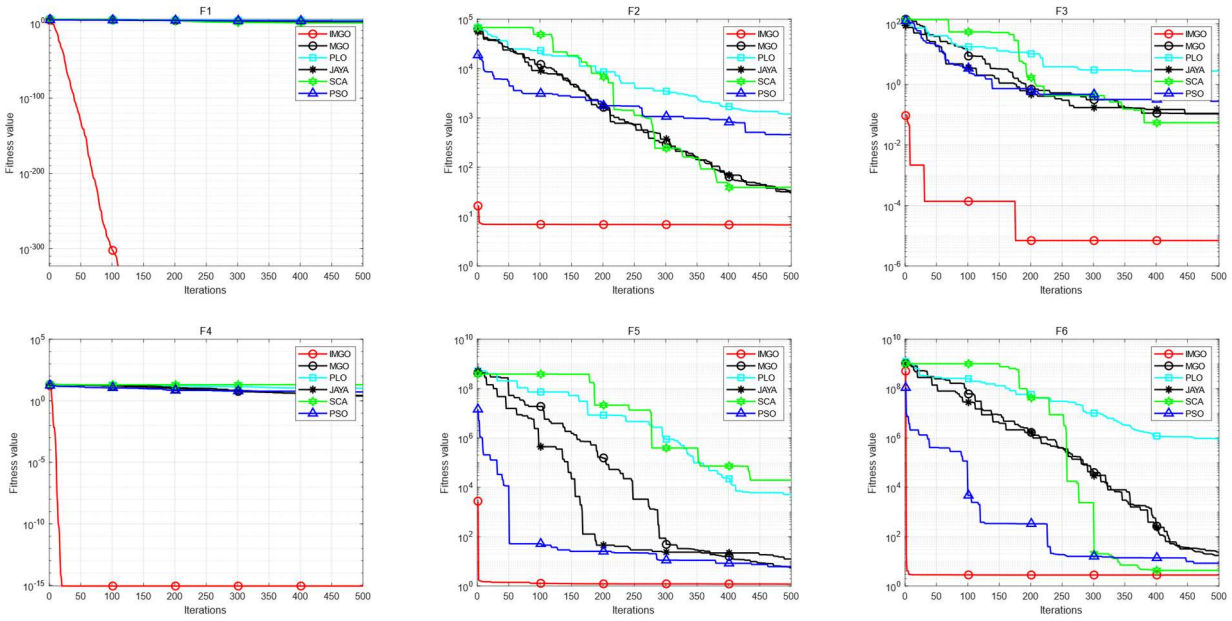


Fig. 3 Fitness Convergence Curves of the F1-F6 Test Functions

Table 2. Optimization Results of Benchmark Functions for Different Algorithms

Function	Metric	**IMGO**	MGO	PLO	JAYA	SCA	PSO
F1	Best	**0.00E+00**	1.10E+01	1.03E+03	2.23E+01	1.75E-02	3.49E+01
	Std. dev.	**0.00E+00**	2.64E+01	1.71E+03	4.22E+01	5.57E+00	1.61E+02
	Mean	**0.00E+00**	1.16E+01	3.68E+02	1.13E+01	1.38E+01	1.21E+02
F2	Best	**4.22E+00**	1.03E+01	8.58E+02	4.14E+01	3.95E+00	2.64E+01
	Std. dev.	**5.69E+00**	2.65E+01	1.62E+03	6.33E+01	1.04E+01	1.42E+02
	Mean	**6.51E-01**	1.24E+01	4.29E+02	1.25E+01	1.30E+01	7.34E+01
F3	Best	**1.75E-06**	3.82E-02	9.81E-01	9.09E-02	1.69E-02	1.81E-02
	Std. dev.	**4.91E-05**	1.05E-01	2.57E+00	1.71E-01	8.85E-02	3.78E-01
	Mean	**4.32E-05**	2.70E-02	7.76E-01	5.96E-02	8.98E-02	9.49E-01
F4	Best	**8.88E-16**	1.95E+00	9.62E+00	2.91E+00	1.26E-02	2.17E+00
	Std. dev.	**8.88E-16**	2.79E+00	1.10E+01	7.33E+00	1.45E+01	4.64E+00
	Mean	**0.00E+00**	3.59E-01	5.98E-01	6.16E+00	8.67E+00	9.68E-01
F5	Best	**2.86E-01**	1.81E+00	5.10E+01	5.99E+00	8.84E-01	8.47E-01
	Std. dev.	**9.93E-01**	4.05E+00	1.34E+04	1.65E+01	5.75E+03	4.11E+00
	Mean	**3.27E-01**	1.68E+00	2.41E+04	6.32E+00	3.11E+04	1.85E+00
F6	Best	**2.64E+00**	1.67E+01	6.79E+04	9.61E+00	2.86E+00	5.31E+00
	Std. dev.	**2.83E+00**	5.25E+02	4.91E+05	5.51E+01	1.49E+03	1.12E+01
	Mean	**8.27E-02**	1.67E+03	2.63E+05	1.02E+02	3.51E+03	3.33E+00

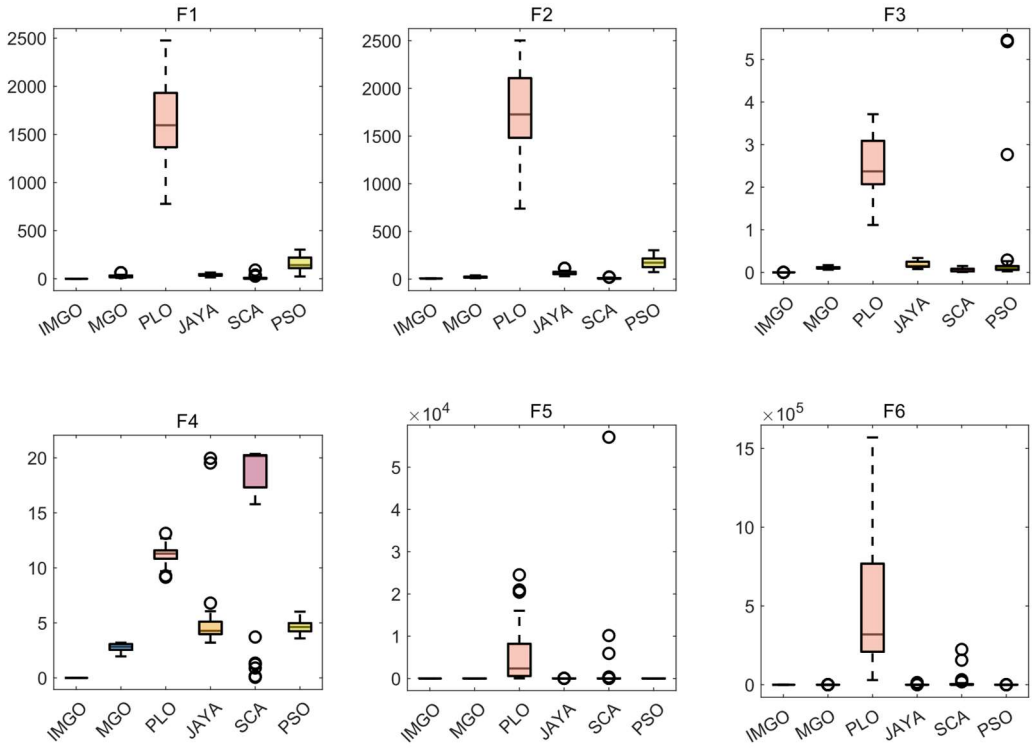


Fig. 4 Boxplots Comparing Six Optimization Algorithms

As shown in Table 2, for the first three unimodal functions F1-F3, the mean and standard deviation obtained by IMGO are significantly better than those of the other algorithms, indicating good stability. Functions F4-F6 are multimodal and impose higher requirements on optimization accuracy. As evidenced by the mean values and standard deviations, IMGO exhibits superior search accuracy and stability in the vicinity of the optimal solution. Figure 3 shows that IMGO basically converges within about 100 iterations, demonstrating high convergence efficiency. Overall, the experimental results indicate that IMGO has high convergence efficiency and fast convergence speed. The boxplots in Fig. 4 further show that the improved IMGO algorithm is stable, contains no outliers, and has medians equal to the optimal values with very small deviations.

3.2. AQI Simulation

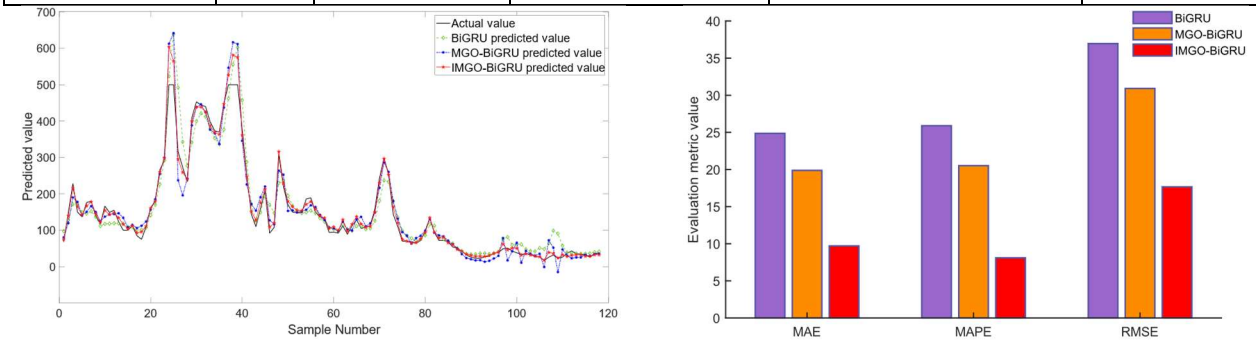
In this experiment, the IMGO-BiGRU data-fusion model was applied to atmospheric environmental quality prediction. The dataset consisted of atmospheric data collected in Daqing (46°35′1″ N, 124°59′58.7″ E) from November 1 to 31, 2024. The input variables included the concentrations of $PM_{2.5}$, PM_{10}, SO_2, NO_2, O_3, and CO, while AQI was used as the output variable. Under the same number of iterations and initial population size, 500 samples were used to train the BiGRU, MGO-BiGRU, and IMGO-BiGRU models, respectively, and the remaining 244 samples were used for testing. To ensure result reliability, 118 groups of consecutive random samples were tested, and the AQI prediction comparison is shown in Fig. 5(a).

To demonstrate the effectiveness of the proposed data-fusion algorithm, the coefficient of determination (R^2), mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE), and elapsed time were selected as evaluation metrics, as shown in Table 3 and Fig. 5.(b)

As shown in Table 3 and Fig. 5, compared with MGO-BiGRU, IMGO-BiGRU improves R^2 by 3 percentage points, while MAE, MAPE, and RMSE are reduced by 10.167, 12.414, and 13.259, respectively, with only a 0.6 s increase in runtime. Compared with the BiGRU model, R^2 increases by more than 7 percentage points, and MAE, MAPE, and RMSE decrease by 15.162, 17.788, and 19.301, respectively. These results indicate that the proposed method provides effective AQI fusion with low error and strong stability.

Table 3. Comparative Analysis of AQI Fusion Performance

Model	R^2	Mean absolute error	Mean absolute percentage error/%	Root mean square error	Elapsed time/s
BiGRU	0.919	24.872	25.895	36.961	12.3
MGO-BiGRU	0.941	19.877	20.521	30.919	16.3
IMGO-BiGRU	0.982	9.710	8.107	17.660	16.9



(a) Comparison Curves (b) Bar Chart of Evaluation Metrics

Fig. 5 Comparison Curves and Bar Chart of Evaluation Metrics

4. CONCLUSION

To enhance solution precision and facilitate the efficient fusion of multidimensional atmospheric data, this paper proposes a data-fusion algorithm based on an IMGO-BiGRU for air quality prediction. The algorithm introduces the FDB strategy, adaptive t-distribution mutation, and opposition-based learning to improve MGO, thereby enhancing optimization accuracy and speed, and IMGO is then used to optimize the hyperparameters of BiGRU. Experimental results show that the AQI prediction achieves a coefficient of determination of 0.982, a mean absolute error as low as 9.710, a mean absolute percentage error of 8.107%, and a root mean square error of 17.660. The proposed algorithm can obtain highly accurate fusion values for air-quality data and exhibits good generalization capability.

REFERENCES

[1] Luo, J., & Gong, Y. (2023). Air pollutant prediction based on ARIMA-WOA-LSTM model. Atmospheric Pollution Research, 14(6), 101761.

[2] Imam, M., Adam, S., Dev, S., et al. (2024). Air quality monitoring using statistical learning models for sustainable environment. Intelligent Systems with Applications, 22, 200333.

[3] Elsaraiti, M., & Merabet, A. (2021). A comparative analysis of the arima and lstm predictive models and their effectiveness for predicting wind speed. Energies, 14(20), 6782.

[4] Berardengo, M., Rossi, G. B., & Crenna, F. (2021). Sea spectral estimation using arma models. Sensors, 21(13), 4280.

[5] Chai, H. M., Chen, F. Z., Fang, M., et al. (2020). Mobile robot obstacle avoidance based on Bayesian networks and fuzzy theory. Control Engineering of China, 27(10), 1657–1664.

[6] Khodarahmi, M., & Maihami, V. (2023). A review on Kalman filter models. Archives of Computational Methods in Engineering, 30(1), 727–747.

[7] Zubair, M., Iqbal, M. A., Shil, A., et al. (2024). An improved K-means clustering algorithm towards an efficient data-driven modeling. Annals of Data Science, 11(5), 1525–1544.

[8] Zheng, Y. (2015). Methodologies for cross-domain data fusion: An overview. IEEE Transactions on Big Data, 1(1), 16–34.

[9] Kalantari, E., Gholami, H., Malakooti, H., et al. (2024). Machine learning for air quality index (AQI) forecasting: shallow learning or deep learning? Environmental Science and Pollution Research, 31(54), 62962–62982.

[10] Das, B., Dursun, Ö. O., & Toraman, S. (2022). Prediction of air pollutants for air quality using deep learning methods in a metropolitan city. Urban Climate, 46, 101291.

[11] Song, T., Li, Y., Meng, F., et al. (2022). A novel deep learning model by Bigru with attention mechanism for tropical cyclone track prediction in the Northwest Pacific. Journal of Applied Meteorology and Climatology, 61(1), 3–12.

[12] Singh, U., Saurabh, K., Trehan, N., et al. (2025). GA-LSTM: Performance Optimization of LSTM driven Time Series Forecasting. Computational Economics, 66(4), 2873–2908.

[13] Huang, W., Huang, X., Zhang, Y., et al. (2025). A Hybrid SSA-VMD-GRU Model for Real-Time Traffic-Related Air Quality Index Prediction: Development and Validation. Sustainability, 17(24), 11233.

[14] Zheng, B., Chen, Y., Wang, C., et al. (2024). The moss growth optimization (MGO): concepts and performance. Journal of Computational Design and Engineering, 11(5), 184–221.

[15] Kahraman, H. T., Aras, S., & Gedikli, E. (2020). Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms. Knowledge-Based Systems, 190, 105169.

[16] Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation, 3(2), 82–102.

[17] Mahdavi, S., Rahnamayan, S., & Deb, K. (2018). Opposition based learning: A literature review. Swarm and Evolutionary Computation, 39, 1–23.

[18] Zhou, X. Y., Liu, Y. Z., Zhao, Y. J., et al. (2025). A hierarchical spatial embedding BiGRU model for destination prediction. Journal of Zhejiang University (Engineering Science), (06), 1211–1218.

[19] Shiri, F. M., Perumal, T., Mustapha, N., et al. (2023). A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. arXiv preprint arXiv:230517473.

[20] Yu, Z. G., Xin, X. J., Fan, Y. H., et al. (2022). A study on oilfield water injection prediction based on a two-layer GRU neural network. Control and Instruments in Chemical Industry, 49(01), 41–46.