AERO-ENGINE EXHAUST GAS TEMPERATURE PREDICTION BASED ON LIGHTGBM OPTIMIZED BY IMPROVED BAT ALGORITHM

Dingzhe LI *, Jingbo PENG, Dawei HE

Aeronautical Engineering College, Air Force Engineering University, Xi'an 710038, China

*Corresponding author: Dingzhe Li; E-mail: ldz710038@163.com

In this paper, an aero-engine exhaust gas temperature (EGT) prediction model based on LightGBM optimized by the chaotic rate bat algorithm (CRBA) is proposed to monitor aero-engine performance effectively. By introducing chaotic rate, the convergence speed and precision of bat algorithm are improved, which CRBA is obtained. LightGBM is optimized by CRBA and it is used to predict EGT. Taking a type of aero-engine for example, some relevant performance parameters from the flight data measured by airborne sensors were selected as input variables and EGT as output variables. The data set is divided into training and test sets, and the CRBA-LightGBM model is trained and tested, and compared with ensemble algorithms such as RF, XGBoost, GBDT, LightGBM and BA-LightGBM. The results show that the mean absolute error (MAE) of this method in the prediction of EGT (after normalization) is 0.0065, the mean absolute percentage error (MAPE) is 0.77% and goodness of fit R^2 *has reached to 0.9469. The prediction effect of CRBA-LightGBM is better than other comparison algorithms and it is suitable for aero-engine condition monitoring.*

Key words*: aero-engine, exhaust gas temperature prediction, LightGBM, improved bat algorithm, flight data*

1. Introduction

The exhaust gas temperature (EGT) of aero-engine is one of the main indicators of the aero-engine condition monitoring. As the service time increases, the performance of the aero-engine will decline and the EGT will rise accordingly. When the EGT exceeds a certain threshold, it may seriously affect the normal operation of the aero-engine and the flight safety of the aircraft [1]. Therefore, the prediction of EGT can effectively monitor engine performance degradation and reduce aircraft failure rates.

Aero-engine is a complex non-linear time-varying system, the EGT varies with the different aero-engine working condition. Up to now, there is no definite mathematical model to describe the change law [2]. It is an effective method to predict the EGT by data mining the historical flying parameter data collected based on the airborne sensors. In the past few years, a series of data-driven methods based on machine learning have been proposed for the prediction of EGT [3-7], and have achieved certain results. Artificial neural networks [3] was applied to predict the EGT of CFM56-7B engines; Zhong and Ding et al. [4, 5] introduced time aggregation operators and used process neural networks to predict EGT; Kumar et al. [6] used auto-regression and moving average technology to predict engine exhaust temperature; On the basis of the process neural network, Pi et al. [7] used the improved drosophila optimization algorithm to optimize the relevant parameters and improved the prediction effect. However, there are still problems such as low accuracy and low prediction efficiency, and the prediction performance of neural network has a great correlation with the number of samples and the setting of hyper parameters.

On account of excellent performance of higher accuracy and shorter running time than individual learner [8, 9]. Ensemble learning has been widely used in classification and regression in the field of machine learning in the past 20 years. However, the new achievements of ensemble learning are very limited in the research of aero-engine. An improved Adaboost model [10] was used to aero-engine PHM, and an improved random forest algorithm [11] was applied to the aero-engine maintenance level decision. LightGBM is an ensemble algorithm based on decision tree published by Microsoft Research Asia [12], which has unique advantages in processing nonlinear models, supports efficient parallel training, and has high accuracy and efficiency in regression and classification. In the past two years, research results can be found in the fields of medicine [13], economy [14], agriculture [15], and meteorology [16]. It is worth noting that the prediction effect of LightGBM is related to its own adjustment parameters. The selection of parameters such as Learning_rate and Max_depth directly affects the prediction accuracy and training speed of the model. At the same time, manual adjustment of parameters takes a long time and may still not be able to make LightGBM has the best predictive performance.

In this paper, the chaotic rate bat algorithm (CRBA) is studied. To further improve the prediction performance, the mean absolute error (MAE) of the predicted value is used as the objective function to optimize the *Learning_rate* and *Max_depth* of LightGBM. The LightGBM prediction model optimized by CRBA is established. The prediction results of several ensemble learning methods are compared. The study provides an application reference for aero-engine condition monitoring.

2. Theory

2.1. LightGBM Model

2.1.1 Decision tree

As a method of classification and regression, decision tree has a tree structure, and mostly uses binary trees [14]. On each leaf node, according to the test results of the judgment condition, the data set is distributed to two or more child nodes, and the child nodes continue to split until the leaf nodes are generated, including the final data category [17].

However, the problem of overfitting will be caused by the transition of decision tree growth, and the classification performance of unbalanced samples is poor, and the information gain tends to be biased to the feature of large sample size. Fig. 1 shows the basic structure of a decision tree.

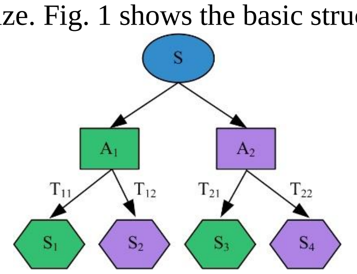


Fig. 1. The basic structure of the decision tree.

2.1.2 Gradient Boosting

Gradient boosting is a machine learning technique used for regression and classification problems.

And it produces a prediction model in the form of a collection of weak prediction models (usually decision trees). The idea of gradient boosting is to iterate the variables at one time. During the iteration process, the sub-models are added one by one [17-18], and at the same time the cost function is continuously reduced.

The model can be represented by the following expression:

$$F_m(x) = \partial_0 f_0(x) + \partial_1 f_1(x) + \cdots + \partial_m f_m(x) \tag{1}$$

Where $f_i(X)$ is the sub-model in each iteration and $L[F_m(x),Y]$ is the cost function, and Y is the ob served value. With the gradual addition of the sub-model, the cost function will decrease along th e variable gradient with the second highest information content:

$$L[F_m(x),Y] < L[F_{m-1}(x),Y] \tag{2}$$

2.1.3 Gradient Boosting Decision Tree

Gradient Boosting Decision Tree (GBDT) is an algorithm for data classification or regression by using a linear combination of primary functions and continuously reducing the residuals generated during training process. In brief, GBDT is equivalent to a decision tree algorithm using gradient boosting. It is a decision tree algorithm with Boosting iteration process, which has the advantages of not easy overfitting and good training effect. GBDT produces a weak classifier in each round of iterations, and each classifier is trained on the basis of the residual of the previous round [12]. In multiple rounds of iterations, the accuracy is continuously improved by reducing the deviation.

2.1.4 LightGBM

LightGBM, as an efficient implementation algorithm of GBDT, is good at processing high-dimensional data and improving calculation efficiency, while ensuring high model accuracy [12]. As shown in Fig. 2, the traditional decision tree algorithm grows the tree through a level-wise strategy and treats the leaves of the same layer indiscriminately, bringing unnecessary overhead. In order to reduce the dimension of training data, the decision tree in LightGBM grows by leaf-wise strategy. Each time from all the leaves, find the one with the largest split gain, and then split to complete a cycle. In order to avoid overfitting when the sample size is insufficient, it is necessary to increase the maximum depth limit of the tree.

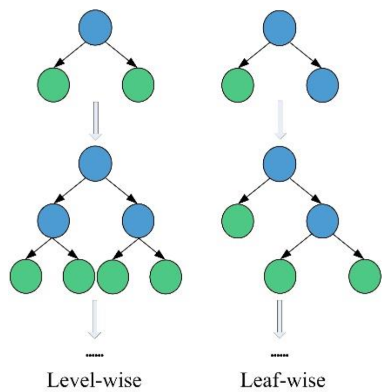


Fig. 2. The generation strategy of tree in LightGBM.

The main parameters for implementing the control and optimization of the LightGBM algorithm are: *Num_leaves*, which is used to set the number of leaves that make up each tree. Setting too large will lead to overfitting while improving accuracy. *Learning_rate*, the learning rate, whose setting is mainly related to the running time. *Max_depth*, it specifies the maximum learning depth or the upper limit of the number of growth layers per tree, which is the main parameter that determines the prediction accuracy; *Min_data*, the minimum amount of data in a leaf; *Feature_fraction*, which selects features that account for the total number of features, scaling from 0 to 1. *Bagging_fraction*, it plays the role of random selection of data, indicating the proportion of selected data in the total data volume, and the value is also between 0 and 1. Otherwise, *Max_depth* and *Min_data* are used to prevent overfitting, and *Feature_fraction* and *Bagging_fraction* are used to control the ratio of the selected total feature number.

Although the LightGBM framework performs very well in all aspects, if the parameters of the model are not properly selected by the user, it will lead to problems such as overfitting or underfitting and insufficient prediction accuracy. It is necessary to select the global optimal hyper parameter combination in a short time.

2.2 Chaotic Rate Bat Algorithm

2.2.1 Bat Algorithm

Cambridge University scholar Yang [19] proposed a new heuristic swarm intelligence optimization algorithm—the bat algorithm (BA) in 2010. The algorithm idealizes the echolocation of bats. By simulating the biological behavior of bat populations using ultrasonic reflection in space to avoid obstacles and search and capture targets. Iteratively updates the speed, position, and optimal fitness function of bat population value [19-20], and then choose the optimal solution until the target stops or the conditions are met, and finally the best solution is obtained.

The position of each bat in the search space corresponds to a solution in the solution space, with corresponding speed and fitness function. The bat population generates a new solution set by updating the emission frequency, pulse rate and loudness, and gradually evolves to a state includes a global or near-optimal solution. The mathematical expression of the iteration process can be written as:

$$f_i = f_{\min} + (f_{\max} + f_{\min})\zeta, \zeta \in [0,1] \tag{3}$$

$$V_i^l = V_i^{l-1} + (X_i^{l-1} - X_{best}) f_i \tag{4}$$

$$X_i^l = X_i^{l-1} + V_i^{l-1} \tag{5}$$

where f_i represents the emission frequency of the bat *i*, $f_{\min}$ and $f_{\max}$ correspond to the minimum and maximum emission frequency of the entire population respectively. ζ is a random variable and its range is limited to [0,1]. X_i^l and V_i^l is the position and speed of the bat *i* in search space in the *l*-th iteration, $i = 1, 2, \cdots, N$. X_{best} is the optimal global position in the *i*-th iteration.

When the algorithm converges to the optimal solution area, the optimal position is perturbed to achieve the purpose of local search again and ensure the ergodicity of the optimal solution. The update equation is as follows:

$$X_{new} = X_{best} + \alpha A^l \tag{6}$$

where α is a random number in the interval [-1, 1], A^l is the average loudness of this bat population[21].

On the basis of Eq. 6, the pulse rate R_i and the loudness A_i are updated as the iteration progress. The update equation is as follows:

$$A_i^{l+1} = \omega A_i^l \tag{7}$$

$$R_i^{l+1} = R_i^0 (1 - \exp(-\beta t)) \tag{8}$$

where β is constant as well as ω, and $\beta > 0$, $0 < \omega < 1$.

In summary, BA has the advantages of simple structure, few input parameters, and good readability. It realizes the conversion between global search and local search of dynamic control [22], and has been shown to perform better than unconstrained optimization Genetic algorithms and particle swarm optimization algorithms [21] and it have a wide range of applications to expand [23-24]. However, the algorithm is easy to reach local optimum and it has disadvantages of low convergence accuracy as well as slow convergence speed.

In view of these shortcomings, scholars have studied and improved them. For example, Rahimi et al. [25] proposed an adaptive learning heuristic bat algorithm to enhance the convergence accuracy of BA. Li et al. [26] merged uniform mutation and Gaussian mutation mechanism to perform selective mutation update on the bat position, which improves the optimization accuracy and convergence speed of the improved algorithm. Liu and Ye [27] proposed to use chaos optimization to help BA achieve better ergodicity and avoid local optimization. Although the above-mentioned literatures have improved BA to some extent, they are all optimizations to update the equation of bat position and speed, without considering the impact of pulse rate and loudness on optimization of model. However, these two parameters are the trigger condition and important measurement parameters for Eq. 6 to perform local traversal optimization. The ability of the bat population to locate the echo is controlled by the pulse rate and loudness [28]. Thus, the optimization of pulse rate and loudness is significant and valuable to increase the overall efficiency of the algorithm.

2.2.2 Optimization Strategy of Chaotic Pulse Rate

Eq. 7 and Eq. 8 are used to update and iterate the pulse rate in BA, that is $R_i^{l+1} \le R_i^0$, $A_i^{l+1} \le A_i^0$. The initial value selection of R_i^0 and A_i^0 will directly affect the ergodicity of the local search of the algorithm. However, due to the manual selection of the initial value, it has some randomness, which may also cause time-consuming and laborious troubles, which is not conducive to the optimal performance of the algorithm. In order to avoid the above problems and improve the optimization performance of the algorithm, this paper improves the pulse rate R_i^{l+1} and the loudness A_i^{l+1} as follows:

$$\begin{cases} R_i^{l+1} = \tau (R_i^l)^2 \sin(\pi R_i^l) \\ \tau = 2.3 \\ R_0 = 0.7 \\ A_0 = 0.9 \end{cases} \tag{9}$$

where R_i^{l+1} is the chaotic pulse rate; τ is the iteration parameter of the pulse rate; R_0 and A_0 are the pulse loudness value and the initial loudness, respectively.

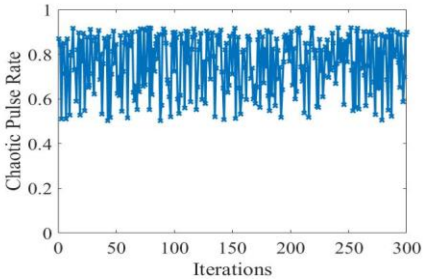


Fig. 3. Value range of chaotic pulse rate.

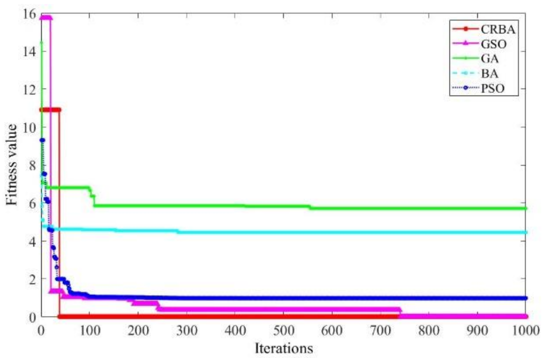
Fig.3 shows that the chaotic pulse rate R_i^{l+1} varies from 0.5 to 1 and is controlled by sinusoidal inverse mapping, therefore it has chaotic ergodicity. Eq. 9 makes the pulse rate have both sensitivity to the initial value and certainty of the chaotic variation range, and can avoid falling into a local optimal value. Meanwhile, the global search ability of the algorithm can be improved.

2.2.3 Confirmatory Analysis

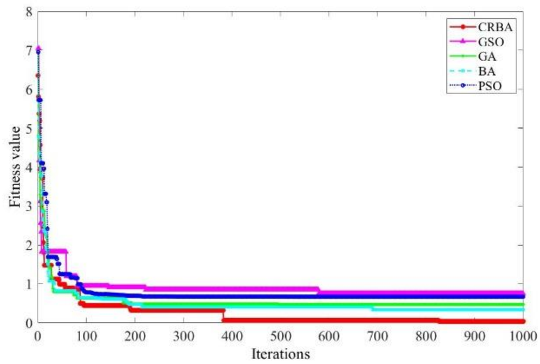
Genetic algorithm (GA), particle swarm optimization algorithm (PSO), glowworm swarm algorithm (GSO) and BA are selected as contrast function, and the test functions are used for comparative simulation to test the optimization performance of CRBA. Tab.1 lists the three test functions.

Tab. 1. Function expressions and their characteristics

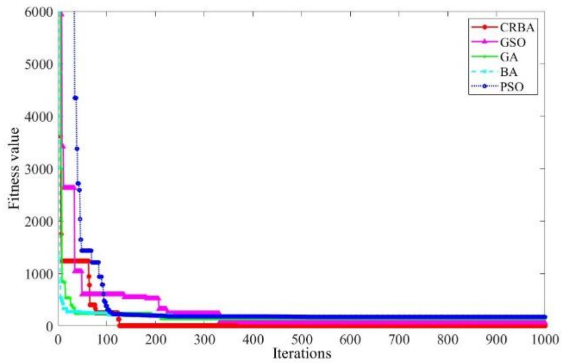
Function	Expression	Search space	Global minimum
Sphere	$f(x)=\sum_{i=1}^{n} x_i^{\,2}$	$\left[-100,100\right]^d$	$x_i=0, f(x)=0$
Griewank	$f\left(x\right)=\sum_{i=1}^{n}\left(100(x_i^2-100\cos(2\pi x_i)+10)\right)$	$\left[-600,600\right]^d$	$x_i=0, f(x)=0$
Rastrigin	$f\left(x\right)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	$\left[-5.12,5.12\right]^d$	$x_i=0, f(x)=0$



(a) Sphere Function



(b) Griewank Function



(c) Rastrigin Function

Fig. 4. Convergence curves of test function.

In order to test the effectiveness of the improved algorithm and compare it with the optimization performance of other algorithms, the experimental algorithms have been standardized. Set the maximum number of iterations i to 1000 and the population size N to 50. The test function and search space are determined according to the range of each test function in Tab.1, in order to reduce the influence of the setting of the population parameters on the performance of BA and other algorithms. Set the pulse rate and loudness of BA and CRBA to be consistent to avoid the influence of parameter setting on the optimization performance, as shown in Eq. 7, $R_0 = 0.7, A_0 = 0.9$ .The initial position of each algorithm is set as: the maximum *Pop_Max*=15 and the minimum *Pop_Min*=-15, and the initial speed is a random value which generates randomly on the basis of the initial position.

Fig.4 shows the test results of each algorithm. As can be seen from Fig.4, the GA and BA algorithms have faster convergence speed among all the algorithms tested, but the convergence accuracy of the two algorithms is not as good as that shown in Fig.4a. From the results of Fig.4b, it can be found that the convergence accuracy of PSO and GSO is slightly inferior. The analysis of results in Fig.4c reflect that the convergence speed of CRBA is the fastest when the convergence accuracy is roughly the same. Therefore, compared with other four algorithms, CRBA has both higher convergence accuracy and faster convergence speed, and has the best comprehensive performance.

2.3. Parameter Optimization of LightGBM Based on CRBA

In order to improve the prediction performance of LightGBM, the parameters need to be adjusted. There are two important parameters related to prediction performance: *Learning_rate* and *Max_depth*, both of which are the main factors influencing the running time and accuracy of the model [18].

Tab. 2. Model parameter settings

Parameter	Value/Option	Parameter	Value/Option
Num_leaves	31	*Min_data*	30
Bagging_ fraction	0.6	*Num_ Iteration*	100
Application	Regression	*Boosting*	GBDT

The other parameter settings mentioned in Section 2.1.4 of this article are shown in Tab. 2. Since boosting defaults to GBDT, *Feature_fraction* is not set. Select MAE as the measure of prediction accuracy:

$$MAE(y, \hat{y}) = \frac{1}{N}\left(\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right|\right) \tag{10}$$

Where, N is the total number of samples in the test set, y_i is the i-th observation sample value, and $\hat{y}_i$ is the i-th prediction sample value.

With MAE as the target function, the CRBA algorithm is used to optimize the *Learning_rate* and *Num_leaves* and find the optimal parameters ultimately. The steps of optimization are as follows:

Step 1 The bat population parameters of CRBA: maximum *Pop_Max* and minimum *Pop_Min* of the initial position are randomly initialized, and the corresponding population position X_i and speed V_i are generated accordingly. Pulse rate (R_0 =0.7), loudness (A_0 =0.9), algorithm dimension (DIM = 2), pulse rate iteration parameters (τ =2.3) and frequency range are set. And set the range of *Learning_rate* (L) to [0.001, 0.5], the range of *Max_depth* (M) to [2, 31], and the bat individual X_i = (L, M) corresponds to the population position.

Step 2 Input the training set samples, and then the algorithm calculates and generates the parameter values (L, M), from which the objective function value of each bat in the first iteration can be obtained, and find the optimal value, record the optimal value of the bat individual location X_{best}.

Step 3 The bat population calculates the emission frequency f_i of each bat within the number of iterations by Eq. 3. The motion speed a is obtained by Eq. 4, and the position b in the search space is updated according to Eq. 5, and perform out-of-bounds processing on speed and position.

Step 4 Generate uniformly distributed random numbers *rand* and ε . If $rand > R_i$, a new global optimal solution is needed, which is generated by ε perturbing the current solution. And perform out-of-bounds processing on it and then calculates the new objective function value.

Step 5 Generate a uniformly distributed random number *rand*. If the random number $rand < A_i$ and $f(X_i) < f(X_{best})$, accept the new solution generated in step 4, and update the loudness and chaotic pulse rate according to Eq. 7 and Eq. 9.

Step 6 Sort the objective function values of all bat individuals, find the optimal value in the current population, and record the position of the optimal value.

Step 7 Repeat step 3 to step 6 until the set optimal solution conditions are satisfied or the algorithm reaches the maximum iterations.

Step 8 Output global optimal value (i.e. minimum MAE value) and optimal solution (i.e. optimal CRBA-LightGBM parameter value).

3. Experiments and Discussion

3.1. Flight Data Selection and Preprocessing

Since the change of aero-engine EGT depends on the working condition of the aero-engine and external conditions, it is necessary to select flight data that can characterize the EGT. The data format and its source is shown in Tab.3.

Tab. 3. Flight parameter format of a type of aero-engine

Parameters	Contents	Parameters	Contents
N_1/%	Low compressor rotor speed	*PLA*/(°)	Throttle Angle
N_2/%	High compressor rotor speed	W_f/kg	Fuel flow
T_6/ °C	Gas temperature after turbine	P_m/MPa	Oil pressure
T_9/ °C	Exhaust gas temperature	T_1/ °C	Inlet temperature
P_6/kPa	Pressure after turbine		

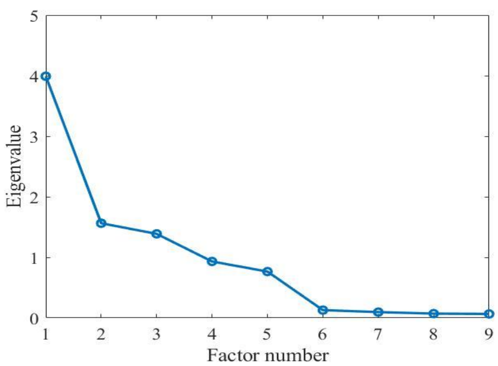


Fig. 5. Scree plot

SPSS (statistical product and service solutions) software is used for factor analysis, and the most influential factors are selected according to the decreasing condition of the eigenvalue and the cumulative variance contribution rate (as shown in Fig.5). After analysis, the first five factors with cumulative variance contribution rate of 95.98% are selected for EGT prediction.

Finally, high compressor rotor speed N_2, low compressor rotor speed N_1, fuel flow W_f, and inlet temperature T_1 are selected as input parameters.

When extracting the above characteristic parameters, the following preprocessing is performed:

- Outlier rejection. In order to avoid affecting the classification effect, for points that deviate significantly from the normal range of the parameters and the remaining parameters are normal at the same time point should be eliminated
- Synchronous processing. The flight data recorder records 4 frames of flight data in 1s, but due to the different sampling frequency of different parameters, they are not synchronized in time, which requires synchronization processing. The processing method is to average the parameters within 1s.
- Data normalization. Due to the different measurement accuracy and dimension of the selected parameters, as well as the need for data confidentiality, all parameters are normalized to 0 and 1.

Tab. 4. Processed sample data

Data points	N_1	N_2	T_1	W_f	T_9
1	0.9828	0.9939	0.4820	0.8133	0.9575
2	0.9802	0.9927	0.4740	0.8130	0.9550
			⋮		
500	0.0495	0.0992	0.8599	0.0240	0.4325
501	0.0513	0.1034	0.8556	0.248	0.4325
			⋮		
999	0.1880	0.2186	0.6113	0.2982	0.5750
1000	0.1830	0.2126	0.6108	0.2902	0.5736

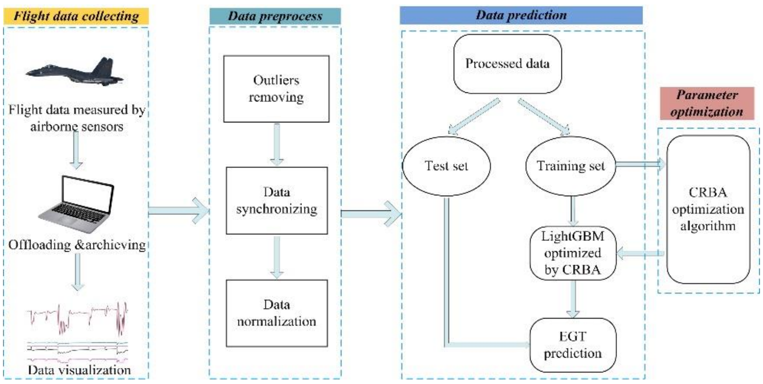


Fig. 6. CRBA-LightGBM EGT prediction flow chart.

According to the above principles and processing methods, a total of 1000 sample data are obtained from 5 sorties. Part of the data has been shown in Tab.4. Divide the above data into training set and test set in a ratio of 4:1. And the prediction flow chart of the prediction model is shown in Fig.6.

3.2 Model evaluation index

MAE value can directly describe the error between predicted value and observed value, thus MAE could be introduced as one of the evaluation indexes. However, since the predicted value is normalized, the MAE value will be too small, and there will be small difference between MAE value of different model. Therefore, the mean absolute percentage error (MAPE) is introduced in order to intuitively reflect the actual prediction error:

$$MAPE(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i-\hat{y}_i|}{y_i}\times 100\% \tag{11}$$

Where, N is the total number of samples in the test set, y_i is the i-th observation sample value, and $\hat{y}_i$ is the i-th prediction sample value.

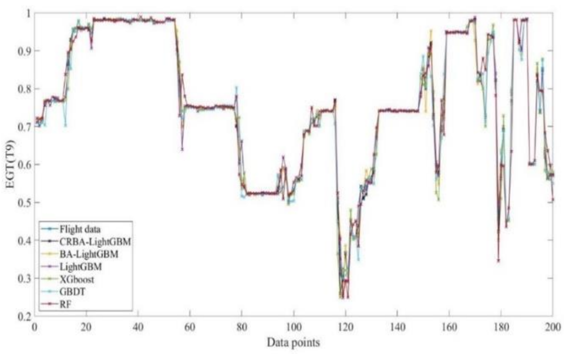
Goodness of fit R^2 is an effective index to measure the degree of fit between regression curves and observed values. This paper also introduces goodness of fit to analyze the performance of each prediction model. The calculation equation of goodness of fit R^2 is shown in Eq. 12:

$$R^2 = \frac{\sum_{i=1}^{N}(\hat{y}_i-\bar{y})^2}{\sum_{i=1}^{N}(y_i-\bar{y})^2} \tag{12}$$

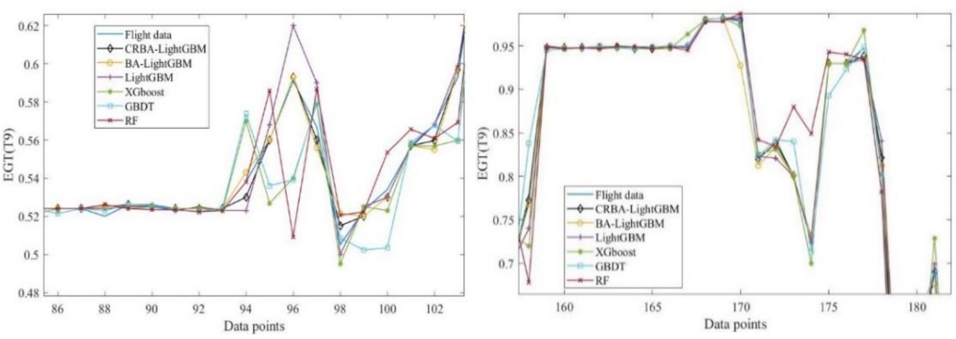
Where, N is the total number of samples in the test set, y_i is the i-th observation sample value, is the mean of the observations, and $\hat{y}_i$ is the i-th prediction sample value.

3.3 Aero-engine EGT prediction

In this part, the prediction performance of the proposed algorithm will be discussed and we will compare it with other ensemble algorithms such as Random Forest (RF), Gradient Boosting Decision Tree (GBDT), Extreme Gradient Boosting (XGBoost), BA-LightGBM and LightGBM. The first 800 sets of sample data are trained, and then the next 200 sets are used for testing.



(a) Prediction results

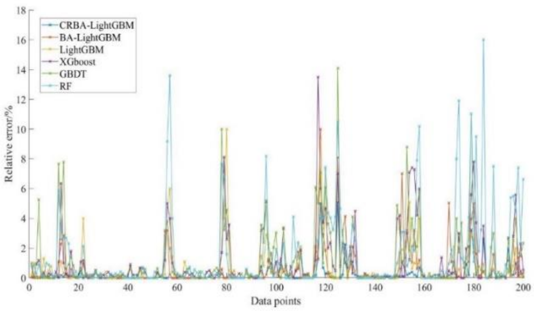


(b) Local magnification

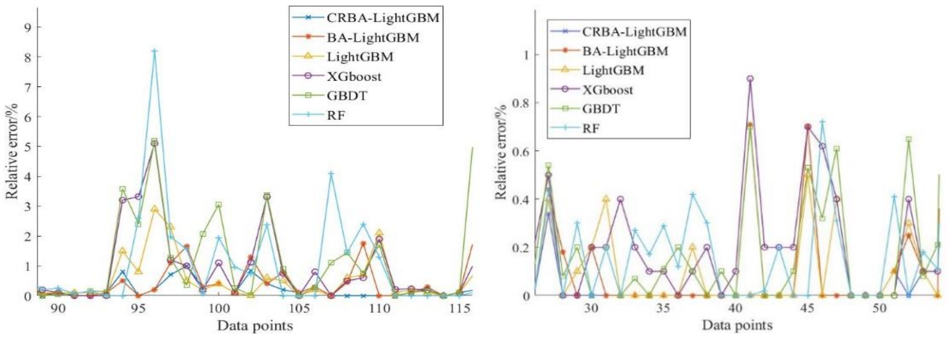
Fig. 7. Comparison of EGT prediction results

Explanation of the parameter setting of the comparison algorithm: all the prediction algorithms involving LightGBM have the same parameter setting except *Learning_rate* and *Max_depth*. For LightGBM without parameter optimization, For LightGBM without parameter optimization, set *Learning_rate*=0.1 and *Max_depth*=10 by default. XGBoost prediction model: adjust *Max_depth*=6, *Eta*=0.2, and select default values for the remaining parameters. GBDT model: Adjust *Max_depth*=6, and select default values for the remaining parameters. RF prediction model: adjust N_{tree}=100, *Max_features* =2, and select default values for the remaining parameters.

Fig.7 shows the prediction results of the six algorithms. In the case of 800 training samples and 200 test samples, the predicted value of the normalized EGT is compared with the observed value. By analyzing the results of Fig.7, the predicted values of the six methods can basically follow the change trend of normalized EGT. According to Fig.7b, between data points 110 and 115, the predicted values of the six methods are basically consistent with the observed values, and the predicted values will fluctuate relatively from the 117th data point, which may be related to the imbalance of training sample.



(a) Relative error



(b) Local magnification

Fig. 8. The relative error between the predicted value and the observed value

Tab. 5. Model prediction performance

Prediction model	CRBA-LightGBM	BA-LightGBM	LightGBM	XGBoost	GBDT	RF
MAE	0.0065	0.0076	0.0092	0.0106	0.0121	0.0162
MAPE/%	0.77	1.03	1.35	2.32	2.49	2.92
R^2	0.9469	0.9212	0.8928	0.8796	0.8501	0.8023
Running time/s	12.89	14.14	10.72	17.08	21.22	11.39

The relative error of the prediction model is shown in Fig.8. Tab.5 shows the prediction performance of the six prediction models, including MAE， MAPE, R^2 and running time. Among Fig. 8b shows that although each prediction model has errors in general, the relative error of CRBA-LightGBM has the minimum fluctuation. Furthermore, Tab.5 and Fig.9 both show more accurately and intuitively that CRBA-LightGBM has the smallest MAE value, followed by BA-LightGBM and LightGBM, the MAE values of the two are relatively close. As the predicted value is normalized, MAE value is close to 0, consequently, the comparison between MAPE values can obviously reflect that the prediction error of CRBA-LightGBM model is the smallest, which indicating that CRBA-LightGBM has the highest prediction accuracy. Furthermore, the prediction accuracy of LightGBM is generally higher in the ensemble algorithm. Meanwhile, CRBA-LightGBM has the largest R^2 value, which is closest to 1, indicating that it fits the observations best overall. From the perspective of running time, LightGBM has the shortest running time, followed by RF. The running time of CRBA-LightGBM is close to RF. In summary, CRBA-LightGBM maintains a high prediction accuracy, while also has a short running time. Therefore, it can be concluded that LightGBM optimized by CRBA is suitable for EGT prediction.

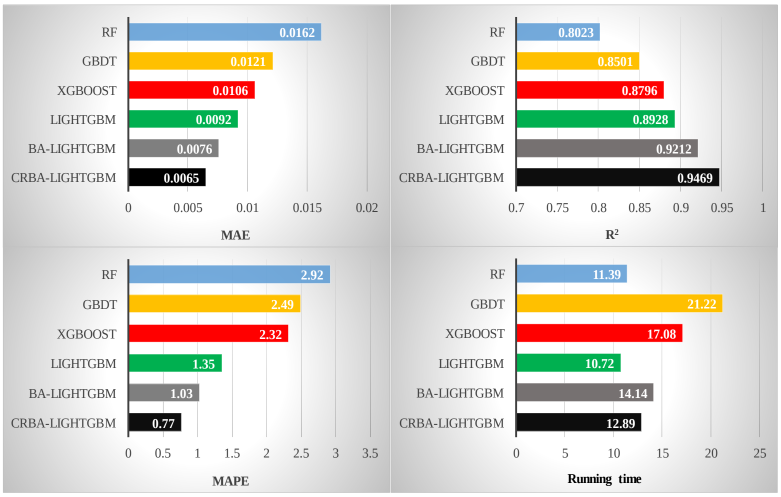


Fig. 9. Model prediction performance comparison

4. Conclusions

In this paper, we propose an aero-engine EGT prediction model based on LightGBM optimized by CRBA algorithm, which can predict the EGT based on the flight data from the airborne sensors.

By introducing sinusoidal inverse mapping to improve the pulse rate, the BA algorithm is optimized. The optimized BA algorithm has higher optimization accuracy and faster optimization speed. The MAE is used as the objective function, and the improved BA algorithm is used to optimize two important parameters in LightGBM. Compared with BA-LightGBM, LightGBM, XGBoost, GBDT and RF, the prediction effect of LightGBM has been better than XGBoost, GBDT and RF models, and the LightGBM optimized by BA and CRBA can further reduce MAE, and CRBA-LightGBM can reach the minimum MAE and MAPE, and it has the best fit to observed values as well as a higher efficiency. Consequently, it can be concluded that CRBA-LightGBM is applicable to aero-engine EGT prediction.

Acknowledgment

This research was funded by the National Natural Science Foundation of China (No. 51506221).

Nomenclature

$f_i(X)$ —sub-model in each iteration

$L[F_m(x),Y]$ —cost function

R_i —pulse rate

R_i^{l+1} —chaotic pulse rate

Num_leaves—number of leaves
Learning_rate—learning rate
Max_depth—maximum learning depth
Min_data—minimum amount of data
Feature_fraction —proportion of selected features in each iteration
Bagging_fraction—proportion of selected data in total data volume
Max_depth—maximum depth of tree
Min_data—minimum amount per tree
f_i —emission frequency of the bat *i*,
$f_{\min}$ —minimum emission frequency
$f_{\max}$ —maximum emission frequency
ζ —random variable limited to [0,1]
X_i^l —position of bat *i* in the *l*-th iteration
V_i^l —speed of bat *i* in the *l*-th iteration
X_{best} —optimal global position in the *i*-th iteration
α —a random number in the interval [-1, 1]
A^l —loudness
Y —observed value
τ —iteration parameter of the pulse rate
R_0 —pulse loudness value
A_0 —initial loudness
Pop_Max—maximum initial position
Pop_Min—minimum initial position
N—total number of samples in the test set
y_i —*i*-th observation sample value
$\hat{y}_i$ —*i*-th prediction sample value
N_1—low compressor rotor speed
N_2—high compressor rotor speed
PLA—throttle angle
W_f—fuel flow
P_m—oil pressure
T_1—inlet temperature
T_6—gas temperature after turbine
T_9—exhaust gas temperature
P_6—pressure after turbine
Eta—learning rate in XGBoost
N_{tree}—numbers of decision tree in RF
Max_features—numbers of features in subset

References

[1] Zolghadri, A., et al., *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles: From Theory to Application*., Springer, 2013.

[2] Yilmaz, O., et al., A Repair and Overhaul Methodology for Aeroengine Components., *Robotics and Computer-Integrated Manufacturing*, 26(2010), 2, pp. 190–201.

[3] Ilbas, M., & Mahmut T., Estimation of exhaust gas temperature using artificial neural network in turbofan engines, *0 Isı bilimi ve tekniği dergisi = Journal of Thermal Sciences and Technology*, 32(2012), 2, pp. 11-18.

[4] Zhong, S., Lei, D., & Ding, G., Convolution sum discrete process neural network and its application in aeroengine exhausted gas temperature prediction, *Acta Aeronautica et Astronautica Sinica*, 33(2012), 3, pp. 438-445.

[5] Ding, G., et al., Prediction of aeroengine exhaust gas temperature using process neural network, *Journal of Aerospace Power*, 24(2009), 5, pp.1035-1039.

[6] Kumar, A., et al. Exhaust Gas Temperature Data Prediction by Autoregressive Models. 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), Nova scotia, Canada, 2015, pp. 976–981.

[7] Jun, P., et al., Aero-engine exhaust gas temperature prediction model based on IFOA-GRNN, *Journal of Aerospace Power,* 34(2019), 1, pp.8-17, 2019.

[8] Maragoudakis, M., & Loukis E., Using Ensemble Random Forests for the extraction and exploitation of knowledge on gas turbine blading faults identification, *Or Insight*, 25(2012), pp. 80-104.

[9] Pan, B., Application of XGBoost algorithm in hourly PM2.5 concentration prediction, *IOP Conference Series: Earth and Environmental Science*, 113(2012), 1, p. 012127.

[10] Huang, Q., et al., A prediction method for aero-engine health management based on nonlinear time series analysis, *ieee international conference on prognostics and health management*, 2016, pp. 1-8.

[11] Zhou, Y., et al., Research on Aero-Engine Maintenance Level Decision Based on Improved Artifi-

cial Fish-Swarm Optimization Random Forest Algorithm, *2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, Xi'an, China, 2018, pp. 606-610.
[12] Ke, G. *et al.*, LightGBM: a highly efficient gradient boosting decision tree, *the Neural Information Processing Systems*, 2017, pp.3146-3154.
[13] Wang, D., et al., LightGBM: An Effective miRNA Classification Method in Breast Cancer Patients, *Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics*, 2017, pp. 7-11.
[14] Ma. X, et al., Study on A Prediction of P2P Network Loan Default Based on the Machine Learning LightGBM and XGboost Algorithms according to Different High Dimensional Data Cleaning, *Electronic Commerce Research & Applications*, 31(2018), pp. 24-39.
[15] Ustuner, M., & Sanli, F., Polarimetric Target Decompositions and Light Gradient Boosting Machine for Crop Classification: A Comparative Evaluation, *Isprs International Journal of Geo Information*, 8(2019), 2, no. 2, p. 97.
[16] Ju, Y., et al., A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecasting, *IEEE Access*, 7(2019), pp. 28309-28318.
[17] Tan, P.-N., Steinbach, M. M., & Kumar, V. *Introduction to Data Mining*, Addison Wesley, Boston, 2015.
[18] Elith, J. Leathwick, J., & Hastie T., A working guide to boosted regression trees, *Journal of Animal Ecology*, 77(2008), 4, pp. 802-813.
[19] Yang, X., A New Metaheuristic Bat-Inspired Algorithm, *Computer Knowledge & Technology*, 284(2010), pp. 65-74.
[20] Wang, Y., et al., Study on Improved Singular Value Decomposition Denoising Method Applied to UAV Flight Parameter Data, *2019 IEEE 20th International Conference on High Performance Switching and Routing (HPSR)*, 2019, pp. 1–6.
[21] Yang, X., Nature-inspired metaheuristic algorithms. *Luniver press*, 2008, pp. 97-104.
[22] Yang, X., Bat Algorithm and Cuckoo Search: A Tutorial. *Artificial Intelligence, Evolutionary Computing and Metaheuristics*, 2013, pp. 421–434.
[23] Adarsh, B., et al., Economic dispatch using chaotic bat algorithm, *Energy*, 96(2016), pp. 666-675.
[24] Karri, C., & Jena, U. Fast vector quantization using a Bat algorithm for image compression, *Engineering Science & Technology An International Journal*, 19(2015), 2, pp. 769-781.
[25] Rahimi, A., et al., The online parameter identification of chaotic behaviour in permanent magnet synchronous motor by Self-Adaptive Learning Bat-inspired algorithm," *International Journal of Electrical Power & Energy Systems*, 78(2016), pp. 285-291.
[26] Dinh, B., et al., Bat optimal algorithm combined uniform mutation with Gaussian mutation, *Control & Decision*, 32(2017), pp. 1775-1781.
[27] Ye. C., Bat Algorithm with Chaotic Search Strategy and Analysis of Its Property, *computer simulation*, 25(2013), 1, pp. 1183-1188.
[28] Hamidzadeh, J., et al., Weighted support vector data description based on chaotic bat algorithm, *applied soft computing*, 60(2017), pp. 540-551.