ISSN 0543-5846
METABK 63(2) 212-214 (2024)
UDC – UDK 621.771:621.1.018:621.72:518.2=111

PREDICTION MODEL OF ROUGH ROLLING FORCE BASED ON CUCKOO-RANDOM FOREST ALGORITHM

Received – Primljeno: 2023-10-30
Accepted – Prihvaćeno: 2023-12-27
Original Scientific Paper – Izvorni znanstveni rad

In order to improve the prediction accuracy of the roughing force prediction model, a cuckoo-random forest algorithm is proposed in this paper to quickly solve the set value of rolling force based on the standard random forest algorithm. The experimental results show that the prediction accuracy of the cuckoo-random forest algorithm is higher than that of traditional models, and it can effectively predict the set value of rough rolling force.

Key words: roughing mill, rolling force, random forest model, Cuckoo algorithm, prediction model

INTRODUCTION

Rolling force is a very important factor in the rough rolling process. The rolling force prediction model is used to calculate the rolling force of each pass of horizontal and vertical rolling mills. Therefore, the accuracy of the prediction model of the rolling force setting of the roughing mill determines the quality of the sheet.

To improve the prediction accuracy of the rolling force setting model, scholars have completed a large number of related studies. In recent years, it has been proposed to combine the artificial intelligence method with the theoretical model of rolling force [1]. Reference [2] optimized the rolling force model through self-learning, thus improving the prediction accuracy. This paper proposes a cuckoo-random forest algorithm to predict the setting value of rolling force in roughing. The experimental results show that the model can effectively improve the accuracy of the online prediction setting model, and provide an effective way to improve the accuracy of the rolling force model.

METHODOLOGY

Random forest[3] is an ensemble learning algorithm that integrates multiple decision trees to complete predictions. The random forest algorithm is a classic Bagging model. To ensure the model generalization ability, the two principles of "data random" and "feature random" are followed when establishing each tree. First, if the size of the training set is N, N training samples are randomly selected from the training set as the training set of each tree. The training set of each tree is different and contains repeated training samples. Second, if there are M features, m feature dimensions are randomly selected from M when each node splits, and the best features in this feature dimension are used to segment nodes. During training, Bootstrap sampling is used to form a training set for each decision tree. When training nodes of the decision tree, the features used are also part of the features extracted from the whole feature vector. The model variance can be reduced by integrating multiple decision trees and training each decision tree using the sampled samples and feature components.

To improve the prediction accuracy of the random forest algorithm, the cuckoo search algorithm was used to optimize the parameters such as the number of trees and the minimum number of leaves in the random forest model to improve the accuracy of the prediction model. The cuckoo search algorithm[4] is a heuristic intelligent algorithm that optimizes model parameters by simulating the behavior process of cuckoo parasitic brooding in nature. Because cuckoos do not raise their own young, they place their eggs in the nests of other birds and foster their own young in them. Therefore, every parasitic nest is a candidate solution.

The traditional cuckoo algorithm has three idealized assumptions: (1) Cuckoo lays only one egg at a time and randomly chooses a nest to incubate it. (2) In a group of nests selected at random, the best nests will be retained for the next generation. (3) The number of optional parasitic nests N is fixed, and the owner of each parasitic nest has a certain probability p of finding his nest parasitized. After the discovery, the cuckoo will randomly choose a new nest as its own parasitic nest.

In the Cuckoo search algorithm, *Levy* flight random walk and preference random walk are two important strategies, responsible for global and local search. *Levy* flight is used in the algorithm to update the nest position X, which is defined as follows:

In an N-dimensional solution space, the position of each nest can be defined as shown in Equation (1).

$$X = \left(x_1, x_2, \cdots, x_n\right) \tag{1}$$

B. Wang (wangboustl@126.com), School of Applied Technology, University of Science and Technology Liaoning, Anshan 114051, China.

In the $t + 1$ generation, the cuckoo will obtain the position of the new parasitic nest according to the parasitic nest position of the t generation, combined with *Levy* flight, and the flight formula is shown in Equation (2).

$$x_{i,n}^{t+1} = x_{i,n}^{t} + \delta \times \lambda \times Levy(\alpha) \times \left(x_{best,n}^{t} - x_{i,n}^{t}\right) \qquad (2)$$

Where $x_{i,d}^{t}$ represents the location of the n-dimensional parasitic nest chosen by the cuckoo i at the generation t. $\delta > 0$ is the step length of the *Levy*'s flight. λ is a uniform random number between [0,1]. *Levy*(α) represents a random distribution that obeys the current iteration number, and its probability distribution is shown in Equation (3). The length s of *Levy* flight can be calculated from Equation (4).

$$Levy(\alpha) \sim u = t^{-\alpha}, 1 \le \alpha \le 3 \qquad (3)$$

$$s = \frac{u}{|v|^{\frac{1}{\beta}}} \qquad (4)$$

Where, $v \in [0,1]$ and $u \in [0, \sigma_u^2]$ are random numbers with normal distribution. $\alpha = 1 + \beta$, $\beta = 1.5$. σ_u^2 can be calculated by Equation (5). The $\Gamma(x)$ in Equation (5) can be calculated using Equation (6).

$$\sigma_u^2 = \left\{ \frac{\Gamma(1+\beta) \times sin\left(\frac{\beta\pi}{2}\right)}{\beta \times \Gamma\left(\frac{1+\beta}{2}\right) \times 2^{\frac{(\beta-1)}{2}}} \right\}^{\frac{1}{\beta}} \qquad (5)$$

$$\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt \qquad (6)$$

The global search is carried out by *Levy* flight, and then the partial solution will perform another local search to update the position, and the new position is shown in Equation (7). Finally, keep the best set of solutions.

$$x_{i,n}^{t+1} = x_{i,n}^{t} + \rho \times \left(x_{j,n}^{t} - x_{k,n}^{t}\right) \qquad (7)$$

Where, ρ is a random number subject to the uniform distribution, and $x_{j,n}^{t}$ and $x_{k,n}^{t}$ represent the positions j and k randomly generated in generation t.

In summary, the steps of Cuckoo search algorithm are as follows:

Step 1: Initialize the cuckoo location, set the population size, maximum discovery probability, and maximum number of iterations.

Step 2: Cuckoo will compare the nest found by levy flight with the previous parasitic nest and select the better parasitic nest as the next parasitic nest. The nest position of the next generation is updated and the fitness value of the objective function is calculated, and the fitness function is shown in Equation (8).

$$f = MSE(\Xi) + MSE(\Theta) \qquad (8)$$

Where Ξ represents the training set, Θ represents the testing set, and *MSE* represents the mean square error. If the value is better than the objective function value of the previous generation, the nest position is updated, otherwise the original position is kept unchanged.

Step 3: After the position is updated, the random number $\varepsilon \in [0,1]$ is compared with p. If $\varepsilon > p$ is used, the position is randomly changed; otherwise keep the position unchanged.

Step 4: Calculate the fitness value of the updated nest and compare it with the fitness value of the better nest to obtain the best set of nest positions and fitness values.

Step 5: Determine whether the algorithm meets the set maximum number of iterations. If yes, end the iterative optimization and output the optimal solution; Otherwise, go to Step 2.

The Cuckoo search algorithm is used to improve the traditional random forest algorithm to improve prediction accuracy. The steps of the Cuckoo Search Random Forest (CS-RF) prediction algorithm proposed in this paper are as follows:

Step 1: preprocess the input rolling force history record and filter out the noise value.

Step 2: Initialize the Cuckoo algorithm. The optimal parameters such as the number of trees and the minimum number of leaves in the random forest model are found by the algorithm, and then the optimal parameters are taken as the initial value of the random forest algorithm.

Step 3: Train the generated random forest model using historical data.

Step 4: The trained model is used to predict the rolling force of roughing mill.

EXPERIMENT AND ANALYSIS

The validity of the prediction algorithm is verified by experiments. The algorithms for experimental comparison of prediction results were Cuckoo Search-Support Vector Machines(CS-SVM)[5], Hidden Markov Model(HMM)[6], and Random Forest(RF)[7], and the main parameters were set as shown in Table 1.

In the experiment, there are 3 000 samples in the history of the set value of vertical roller mill rolling force in the first rough rolling, and the rolling force is between (588, 24500) KN. The model was trained and predicted by the method of 10-fold cross-validation, and the average result was given. After the training, the

Table 1 **Simulation setting**

Name	Parameters
CS-SVM	*p* = 0,25, *δ* = 1, *N* = 30, *M* =30 kernel="rbf",, degree=3, gamma="scale", tol=1e-3, C=1,0, epsilon=0,1
HMM	covariance_type='diag', min_covar=1e-3, ovars_prior=1e-2, n_hidden_states=4
RF	n_estimators = 1 000, random_state = 42, criterion="gini", max_depth=5
CS-RF	*p* = 0,25, *δ* = 1, *N* = 30, *M* =30 n_estimators = 1 000, random_state = 42, criterion="gini", max_depth=5

rolling force of rough rolling is predicted, and then the prediction results of the above algorithm are evaluated. The evaluation indicators are Mean Squared Error(MSE), Root Mean Square Error(RMSE), Mean Absolute Error(MAE), and R-squared. The experimental results are shown in Table 2.

Table 2 **Comparative experimental evaluation results**

Name	MSE	RMSE	MAE	R-squared
CS-SVM	105938,732683	325,482308	175,982367	0,533876
HMM	317,760838	17,825847	13,112037	0,997876
RF	7,927064	2,815504	0,814690	0,999966
CS-RF	3,216941	1,793583	0,661239	0,999987

It can be seen from the experimental results that the richer the historical data, the closer the predicted results are to the actual values. Cuckoo Random Forest (CS-RF) model predicted better results than other comparison algorithms. The average accuracy of the CS-RF prediction algorithm can reach 94,9 %, the average accuracy of the RF prediction algorithm is 92,88 %, the average accuracy of HMM prediction algorithm is 89,91 %, and the average accuracy of the CS-SVM prediction algorithm is 68,31 %. In particular, for roughing rolling force set values with significant historical data changes, or for new varieties just put into production, there is a difference between the predicted and the actual value, but this difference can be gradually narrowed after several rounds of learning. At the same time, the prediction algorithm proposed in this paper has better performance than other comparison algorithms in four evaluation indicators. Therefore, the prediction model established in this paper can predict the actual rolling force setting value more accurately.

CONCLUSION

In this paper, the random forest algorithm is used to predict the rolling force of roughing mill by using historical data, and the cuckoo algorithm is used to dynamically adjust the parameters of the prediction model, to improve the accuracy of the model. The experimental results show that the proposed model has good performance.

Acknowledgemets

The work was financially supported by Research Projects of Colleges and Universities of the Liaoning Provincial Department of Education in China (No. LJKZ0311).

REFERENCES

[1] W.G Li, C. Liu, B. Liu, et al. Modeling friction coefficient for roll force calculation during hot strip rolling. The international Journal of Advanced Manufacturing Technology 92(2017)1-4, 597-604.

[2] F.W. Zhu, X. L. Hu, Z. Zhao, et al. High Precision Short-term Self-learning of Rolling Force in Plate Rolling Process, Journal of Northeastern University, 29 (2008) 7, 980-983.

[3] L. Breiman, Random forests[J]. Machine learning, 45 (2001) 1, 5-32.

[4] X.S. Yang, S. Deb, Cuckoo search via Lévy flights[C]. Proceedings of the 2009 World congress on nature & biologically inspired computing (NaBIC), Coimbatore, India, 2009, 210-214.

[5] H. N. Shah, Prediction of Stock Market Using Artificial Intelligence[C]. Proceedings of the 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, 2019, 1-6.

[6] A. Sacco, F. Esposito, G. Marchetto, Restoring Application Traffic of Latency-Sensitive Networked Systems Using Adversarial Autoencoders[J]. IEEE Transactions on Network and Service Management, 19(2022)3, 2521-2535.

[7] M. Adhikari, M. Ambigavathi, V. G. Menon, et al. Random Forest for Data Aggregation to Monitor and Predict COVID-19 Using Edge Networks[J]. IEEE Internet of Things Magazine, 4 (2021) 2, 40-44.

Note: The responsible translator for English is Bo Wang, University of Science and Technology Liaoning, Anshan, China