

Research Article

Rolling Force Prediction of Hot Rolling Based on GA-MELM

Jingyi Liu ¹, Xinxin Liu,² and Ba Tuan Le ^{2,3}

¹College of Sciences, Northeastern University, Shenyang 110819, China

²School of Information Science and Engineering, Northeastern University, Shenyang 110819, China

³NTT Hi-Tech Institute, Nguyen Tat Thanh University, Ho Chi Minh City 700000, Vietnam

Correspondence should be addressed to Ba Tuan Le; lebatuan@qq.com

Received 18 January 2019; Revised 9 May 2019; Accepted 4 June 2019; Published 24 June 2019

Academic Editor: Mahardhika Pratama

Copyright © 2019 Jingyi Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the hot continuous rolling process, the main factor affecting the actual thickness of strip is the rolling force. The precision of rolling force calculation is the key to realize accurate on-line control. However, because of the complexity and nonlinearity of the rolling process, as well as many influencing factors, the theoretical analysis of the traditional rolling force prediction model often needs to be simplified and hypothesized. This leads to the incompleteness of the mathematical model and the deviation between the calculated results and the actual working conditions. In this paper, a rolling force prediction method based on genetic algorithm (GA), particle swarm optimization algorithm (PSO), and multiple hidden layer extreme learning machine (MELM) is proposed, namely, PSO-GA-MELM algorithm, which takes MELM as the basic model for rolling force prediction. In the modeling process, GA is used to determine the optimal number of hidden layers and the optimal number of hidden nodes, and PSO is used to search for the optimal input weights and biases. This method avoids the influence of human intervention on the model and saves the modeling time. This paper takes the actual production data of BaoSteel 2050 production line as experimental data, and the experimental results indicate that the algorithm can be effectively used to determine the optimal network structure of MELM. The rolling force prediction model trained by the algorithm has excellent performance in prediction accuracy, computational stability, and the number of hidden nodes and is applicable to the prediction of rolling force in hot continuous rolling process.

1. Introduction

In the hot continuous rolling production process, the main factor affecting the actual strip thickness is the rolling force, and the accurate calculation of the rolling force is the key to achieve accurate online control [1]. The traditional calculation of rolling force [2] is carried out by mathematical model, because the establishment process of mathematical model itself ignores and simplifies many practical factors of rolling production site, the error of calculating rolling force solely by mathematical model is large, which cannot meet the increasingly accurate rolling requirements. Unlike traditional methods, the artificial intelligence approach [3] avoids endless exploration of the deep laws of the rolling process and instead simulates the human brain to process what actually happens. It does not start from the basic principle, but takes the fact and the data as the basis, and reveals the parameter change rule in the rolling process.

With the development of artificial intelligence, the method of using artificial neural network to predict rolling force starts to rise [4–7], and a wave of intellectualized rolling technology is emerging. A number of examples of applying the intellectualized methods such as expert system, neural network and fuzzy logic to rolling process are presented [8–12]. In previous studies, most rolling force prediction methods based on neural network adopt BP neural network, which is based on gradient descent algorithm. All network parameters need to be updated iteratively, so the learning speed is slow and it is easy to fall into local optimization. Moreover, the whole network is also sensitive to the selection of learning efficiency. These inherent shortcomings have become the main bottleneck restricting its development.

Extreme learning machine (ELM) [13] is a novel learning algorithm for the single hidden layer feedforward neural network (SLFNs). This algorithm is simple and easy to implement, with fast learning speed and good generalization

performance. Many experimental studies have shown that ELM tends to select more hidden layer nodes due to the random selection strategy of input layer parameters. Therefore, many researchers are committed to the optimization of ELM model structure [14–16]. With the emergence of deep learning theory, a novel neural network based the algorithm named MELM [17] is presented that, by making the actual hidden layers output approach the expected hidden layers output, both the average training and testing performance are improved to a significant degree. Feng [18] et al. proposed an EM-ELM algorithm based on error minimization, which can add new hidden nodes individually or in batches. Lan [19] et al. further improved it. Rong [20] et al. proposed a pruning algorithm P-ELM to perform pruning of hidden nodes by using statistics and information gain. Miche [21] et al. also proposed another pruning algorithm, OP-ELM. In addition, Lan [22] et al. proposed a two-stage model construction method, TS-ELM, for regression problems. Due to the need of some practical application problems, the research on online learning and incremental learning algorithm is of great significance [23–25]. The online sequence learning algorithm (OS-ELM) proposed by Liang [26] et al. can realize the batch (or single) learning of samples in the training process, which is suitable for solving some online learning problems. Literature [27] studied the incremental learning algorithm (I-ELM) of ELM, in which hidden nodes can be added into the model one by one in the process of training. While ELM has been deeply studied and greatly improved in algorithm, it has been successfully applied in more and more fields due to its advantages of easy implementation, fast learning speed, and high precision [28]. At present, algorithms have been widely used in pattern recognition, regression estimation, and other aspects [29–31]. Specific application problems include the following: face recognition [32], time series prediction [33], soft sensing [34], medical diagnosis [35], communication technology [36], image processing [37], text classification [38], economic analysis [39], remote sensing [40], and so on. The above research work not only solves some important problems in the algorithm, but also improves its performance and broadens its research scope.

In many complex practical application problems, due to the complexity of calculation, we often hope to achieve the desired effect with fewer hidden nodes, which is the advantage of MELM network. Obviously, reasonable selection of MELM network structure is a guarantee to effectively avoid overfitting and improve generalization performance. The essence of network structure selection lies in the establishment of network structure selection criteria. For the MELM algorithm, it is necessary to study effective network structure selection criteria. Stability is also a very important performance index of a learning algorithm, which has a significant influence on the applicability of an algorithm in practical problems. In the MELM algorithm, the input weights and biases are selected randomly, which is crucial for simplifying the complexity of the model, but this randomness will inevitably affect the stability of the model and affect its application in many problems requiring high modeling accuracy and stability. Therefore, how to improve the stability of the model is undoubtedly a very important research topic.

When using the MELM model to predict the rolling force in the hot continuous rolling process, in order to realize the effective design of the MELM network structure, this paper proposes a multiple hidden layer extreme learning machine method based on genetic algorithm and particle swarm optimization algorithm, namely, POS-GA-MELM. In this method, MELM is used as the basic model for rolling force prediction, and GA is used to determine the optimal number of hidden layers and the corresponding optimal number of hidden nodes in the MELM network, so as to reasonably select the network structure of the model. PSO is used to determine the optimal input weights and biases to enhance the stability of the model. In this paper, the actual production data of BaoSteel 2050 production line is used as training and test data to establish a POS-GA-MELM based rolling force prediction model. The validity of the model will be verified by a rolling force prediction example in the hot continuous rolling process.

The rest of this paper is organized as follows: Section 2 presents a brief review of the basic concepts and related work of the original ELM and the multiple hidden layers ELM, Section 3 describes the proposed PSO-GA-MELM technique, Section 4 reports and analyzes the experimental results, and, finally, Section 5 summarizes key conclusions of the present study.

2. Brief Review of ELM and MELM

In this section, we shortly introduce the original ELM and MELM, respectively. In the following, the structure of SLFNs and the theory of ELM are briefly reviewed.

2.1. Extreme Learning Machine. Extreme learning machine is an easy to utilize and effective algorithm for the single hidden layer feedforward neural network; it can adaptively set the number of hidden nodes and randomly assign for the input weights and biases, the output weights obtained by the least square method, and the whole training process completed through one mathematical change without iteration and generates a unique optimal solution, with the advantages of fast learning speed and generalization performance. Compared with traditional BP algorithm based on gradient descent, the learning speed of ELM has significantly improved. Specifically, the ELM learning algorithm mainly has the following steps:

Algorithm 1 (ELM algorithm). Given Q training samples $\{X, T\} = \{x_i, t_i\}$ ($i = 1, 2, \dots, Q$) and L hidden nodes with activation function $g(x)$:

- (1) Determine the number of hidden nodes, and randomly set the connection weights between the input layer W and the hidden layer and the biases of the hidden nodes B .
- (2) Select an infinitely differentiable function as the activation function of hidden nodes, and then calculate the hidden layer output matrix $H = g(WX + B)$.
- (3) Calculate the weights between the hidden layer and the output layer using the least-square method $\beta = H^+T$.

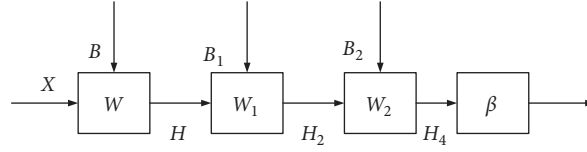


FIGURE 1: The workflow of the MELM approach.

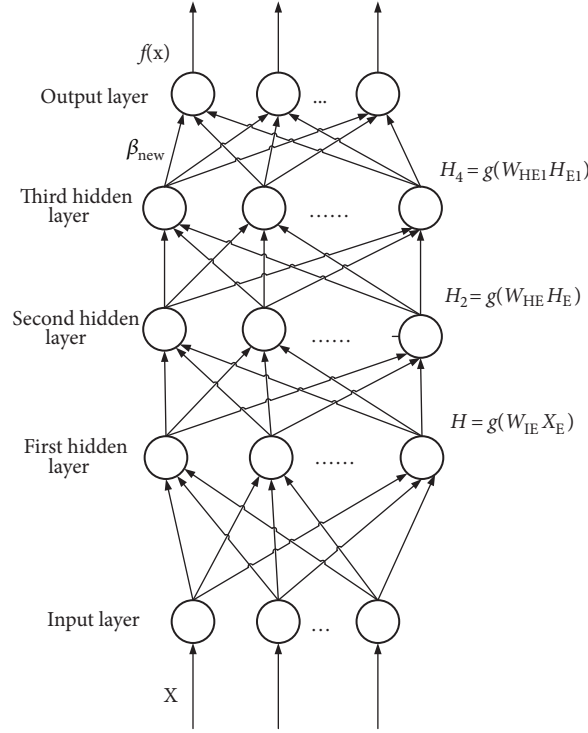


FIGURE 2: The structure of the MELM approach.

2.2. Multiple Hidden Layer Extreme Learning Machine. The structure of MELM neural network is composed of input layer, multiple hidden layers (the number of hidden layers is greater than 3), and output layer. Meanwhile, MELM network inherits the theory of ELM network randomly initializing the weights matrix between the input layer and the first hidden layer and the biases vector of the first hidden layer. By making the actual hidden output approach the expected hidden layer output, the parameters of the remaining hidden layers (the weights matrix and the biases vector) are obtained, and a new network with multiple hidden layers is constructed. In the following, we take the ELM network with three hidden layers as an example to introduce the MELM network. The workflow of the MELM architecture is illustrated in Figure 1, and the MELM network structure is depicted in Figure 2. The implementation of the MELM proceeds according to the following steps.

Algorithm 2 (MELM algorithm). For a given training data set $\{X, T\} = \{x_i, t_i\}$ ($i = 1, 2, \dots, Q$), the number of hidden nodes L , and activation function $g(x)$:

(1) Assign the connection weights matrix between the input layer and the first hidden layer W and the biases matrix

of the first hidden layer B randomly. For simplicity, defined the augmented matrix $W_{IE} = [B \ W]$, $X_E = [1 \ X]^T$.

(2) Calculate the first hidden layer output matrix $H = g(W_{IE} X_E)$.

(3) Calculate the output weights matrix between the second hidden layer and the output layer β .

(a) If the number of training samples is greater than the number of hidden nodes, we can obtain the output weights matrix $\beta = (I/\lambda + H^T H)^{-1} H^T T$;

(b) If the number of training samples is less than the number of hidden nodes, we can obtain the output weights matrix $\beta = H^T (I/\lambda + H H^T)^{-1} T$;

(c) If the number of training samples is equal to the number of hidden nodes, we can obtain the output weights matrix $\beta = H^+ T$.

(4) Calculate the expected output matrix of the second hidden layer $H_{1*} = T \beta^+$.

(5) Determine the parameters of the second hidden layer W_{HE} , that is the connection weights matrix between the first and second hidden layer W_1 and the biases of the second hidden layer B_1 , where $W_{HE} = [B_1 \ W_1] = g^{-1}(H_{1*}) H_E^+$,

the matrix $H_E = [1 \ H]^T$, the notation $g^{-1}(x)$ represent the inverse of the activation function $g(x)$.

(6) Obtain the actual output of the second hidden layer $H_2 = g(W_{HE}H_E)$.

(7) Recalculate the weights matrix between the second hidden layer and the output layer β_{new} .

(a) If the number of training samples is greater than the number of hidden nodes, we can obtain the output weights matrix $\beta_{new} = (I/\lambda + H_2^T H_2)^{-1} H_2^T T$;

(b) If the number of training samples is less than the number of hidden nodes, we can obtain the output weights matrix $\beta_{new} = H_2^T (I/\lambda + H_2 H_2^T)^{-1} T$;

(c) If the number of training samples is equal to the number of hidden nodes, we can obtain the output weights matrix $\beta_{new} = H_2^+ T$.

(8) Calculate the expected output matrix of the third hidden layer $H_{3*} = T \beta_{new}^+$.

(9) Determine the parameters of the third hidden layer W_{HE1} that is the connection weights matrix between the second and third hidden layer W_2 and the biases of the third hidden layer B_2 , where $W_{HE1} = [B_2 \ W_2] = g^{-1}(H_{3*})H_{E1}^+$, the matrix $H_{E1} = [1 \ H_2]^T$, the notation $g^{-1}(x)$ denote the inverse of the activation function $g(x)$.

(10) Obtain the actual output of the third hidden layer $H_4 = g(W_{HE1}H_{E1})$.

(11) Recalculate the weights matrix between the third hidden layer and the output layer β_{new1} .

(a) If the number of training samples is greater than the number of hidden nodes, we can obtain the output weights matrix $\beta_{new1} = (I/\lambda + H_4^T H_4)^{-1} H_4^T T$;

(b) If the number of training samples is less than the number of hidden nodes, we can obtain the output weights matrix $\beta_{new1} = H_4^T (I/\lambda + H_4 H_4^T)^{-1} T$;

(c) If the number of training samples is equal to the number of hidden nodes, we can obtain the output weights matrix $\beta_{new1} = H_4^+ T$

(12) Obtain the final output of MELM network $f(x) = H_4 \beta_{new1}$.

3. Proposed PSO-GA-MELM

In the MELM network, the number of hidden layers and the number of hidden layer nodes need to be predetermined artificially. The selection of appropriate network structure has a direct impact on the performance of the model. Too many hidden layers and hidden nodes will increase the structural complexity of the model, and too few hidden layers and hidden nodes will not reach the optimal accuracy of the model. Up to now, there is no established theory on how to select the number of hidden layers and the number of hidden nodes. In most cases, it is based on the experience of scholars themselves or through a large number of experiments to determine the appropriate network structure. However, for the MELM model, if the appropriate network structure is still determined through experiments, it will be very time-consuming. In addition, in order to simplify the model, the input weights and biases of the first hidden layer in the MELM algorithm are selected randomly, which will greatly affect the

stability of the model. Therefore, this paper introduces GA and PSO algorithm into the framework of MELM and puts forward a novel learning algorithm named PSO-GA-MELM. The workflow of the proposed PSO-GA-MELM technique is demonstrated in Figure 3. In the modeling process of PSO-GA-MELM, GA is used to determine the optimal number of hidden layers and the corresponding optimal number of hidden nodes, and PSO is used to determine the optimal input weights and biases of the first hidden layer, so as to select the model network structure reasonably and enhance the stability of the model.

Genetic algorithm [41] is a parallel stochastic search optimization algorithm that simulates the genetic mechanism and biological evolution in nature. It introduces the biological evolution principle of survival of the fittest in nature into the coding tandem population formed by optimization parameters and screens the individuals according to the selected fitness function and through the selection, crossover, and variation in heredity, so that the individuals with good fitness are retained and those with poor fitness are eliminated. The new population not only inherited the information of the previous generation, but also is superior to the previous generation. In this way, the cycle is repeated until the conditions are met. GA is characterized by efficient heuristic search and parallel computation and has been widely used in function optimization, combination optimization, and production scheduling.

PSO is a swarm intelligence optimization algorithm [42], in which each particle represents a potential solution. The velocity of particles determines the direction and distance of particle movement, and the velocity is dynamically adjusted according to the movement experience of particle itself and other particles, so as to achieve optimization in the solvable space. In order to ensure the prediction accuracy of the model, after determining the structure of MELM with GA, PSO is used to optimize the input weights and biases. According to the above discussion, the proposed PSO-GA-MELM can be summarized as follows.

Algorithm 3 (proposed algorithm PSO-GA-MELM). (1) Determine the optimal input weights matrix and biases vector of the ELM model with different hidden nodes.

(2) Initialize the GA algorithm, determine the individual length, the size of the population, the number of iterations, the crossover probability and the mutation probability, and randomly generate the binary population.

(3) Convert the binary population to the decimal system for ELM modeling. The input weights and biases are no longer generated randomly, but are directly called from the results saved in step 1 according to the number of hidden nodes.

(4) Calculate the fitness value of each individual to find the optimal individual. In the step, the fitness value of the individual is the mean absolute error of the prediction model, and the individual with the smallest error is the optimal individual.

(5) Implement selection operation, crossover operation, and mutation operation successively.

(6) Determine whether the maximum number of iterations is reached. If the maximum number of iterations is not

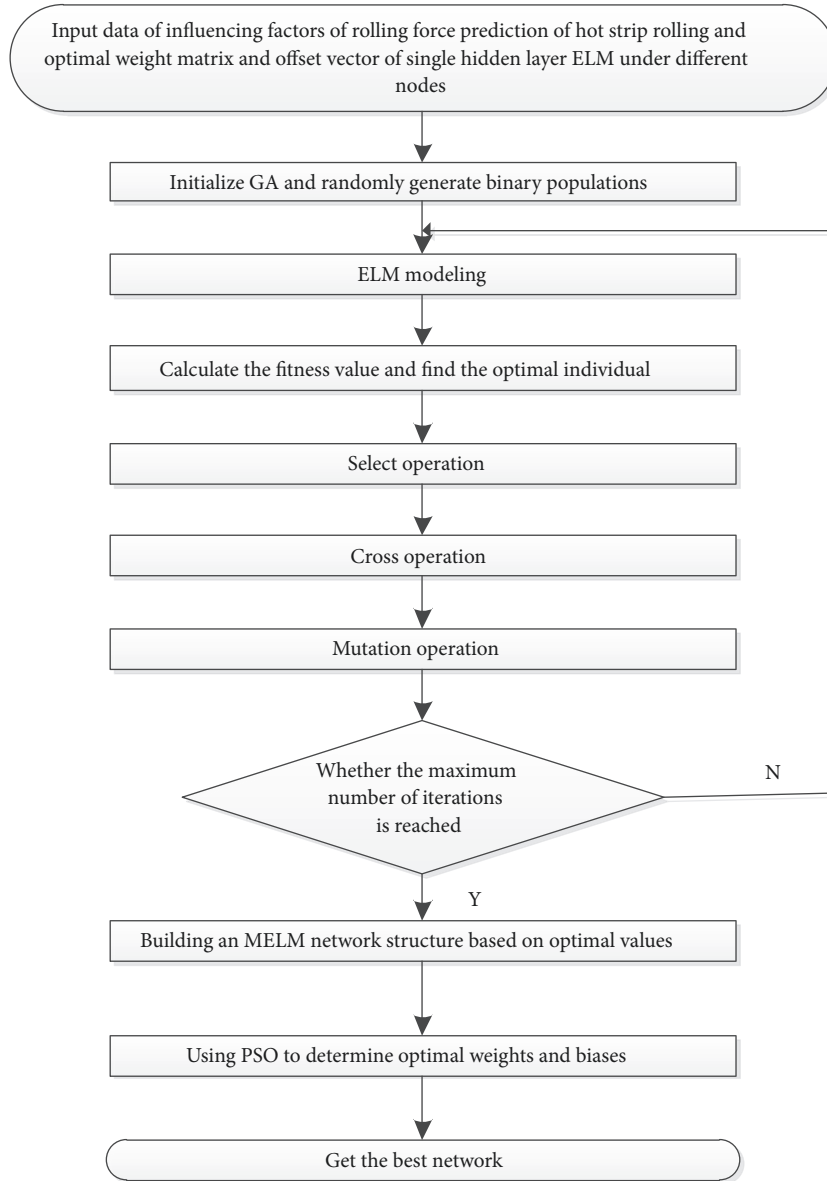


FIGURE 3: The workflow of the proposed PSO-GA-MELM approach.

reached, return to step (3); otherwise, the optimal MELM network structure is obtained.

(7) Determine the number of parameters to be optimized for PSO.

(8) Assign and initialize the basic parameters, including the number of population iterations, the population size, the maximum and minimum of the individual, and the maximum and minimum of the speed.

(9) Calculate the fitness value of the particles and search for the individual extremum and the population extremum. In this step, the fitness value of the particles is the reciprocal of the rolling force prediction error. The higher the fitness value is, the better the particles are.

(10) Update the velocity and position of the particle according to the following formula. The velocity update formula is $V_{id}^{k+1} = 0.8V_{id}^k + 1.49445r_1(P_{id}^k - X_{id}^k) + 1.49445r_2(P_{gd}^k -$

$X_{id}^k)$, the position update formula is $X_{id}^{k+1} = X_{id}^k + V_{id}^{k+1}$, where k is the number of current iterations, V_{id} is the particle velocity, r_1 and r_2 are random numbers distributed between $[0, 1]$.

(11) Calculate the fitness value of the particle, update the individual extremum and the population extremum according to the fitness value of the particle.

(12) Determine whether the maximum number of iterations is reached. If the maximum number of iterations is not reached, return to step (10); otherwise, the optimization process is completed, and the optimal input weights and biases are obtained.

4. Experiments and Results

Extensive experiments were conducted to evaluate the prediction accuracy of the proposed PSO-GA-MELM, which are

further compared with PSO-ELM and deep belief networks (DBN) [43].

4.1. Acquisition of Training Samples. This paper utilizes the actual production data of BaoSteel 2050 production line as the experimental data. The production line of BaoSteel 2050 is mainly used to roll the slabs after rough rolling into strips with required thickness. First of all, we briefly introduce the technological process of the hot rolling mill.

The slabs delivered by the blooming mill and continuous casting are heated by a step-type continuous heating furnace. After removing phosphorus with high pressure water and controlling the width and shape of the slab with strong roller, the slab is sent to no.1-4 roughing mill to be rolled into intermediate billet. Thus, forming a three-quarters hot continuous rolling mode with the finishing mill behind. The roughing mill is numbered as 1-4 according to the order in which the slabs enter. Among them, no. 1 roughing mill is two-high reversible mill, no. 2 mill is four-high reversible mill, and nos. 3 and 4 roughing mill are four-high irreversible continuous mill. The thickness of the intermediate blank is 38-65mm. After the cutting head by crop shear, it is sent to seven sets of continuous four-high finishing mill to be rolled into finished strip steel. Though laminar flow cooling and temperature control, the strip is coiled into steel coils by bottom coiler, and the finished steel coils are used for different purposes according to the quality.

In finishing rolling process, motor and hydraulic press are used for thickness control, and seven CVC working rollers and hydraulic bending rollers are used for shape control. The seven racks are identical in structure, but the work rolls are different in diameter. In the experimental data selected in this paper, the diameters of the seven rollers are, respectively, 810mm, 837mm, 838mm, 714mm, 691mm, 741 mm, and 758 mm. The frame is denoted as F1-F7 according to the entry sequence of the strip steel, the height of the F1-F3 engine base is higher than that of the F4-F7 engine base, the F1-F3 adopt Cr, Ni semi-steel roll, the F4-F7 adopt infinite chilled cast iron roll, the F1-F5 is driven by the main motor through the reducer, gear machine, and universal joint shaft, and the F6-f7 is driven by motor through gear base and universal joint shaft. The rolling temperature decreases successively and the rolling speed increases gradually.

In this experiment, a total of 134 sets of data were obtained, which were respectively from 134 rolls of strip steel, belonging to batch data. Since the rolling time of a coil of strip is about 60 seconds to 80 seconds, the time interval for obtaining each group of data is 60 seconds to 80 seconds. We get the value of influencing factors through the sensor on the rolling mill, among which the first 100 groups were selected as training data and the last 34 groups as test data. The factors that affect the prediction of rolling force are the width of incoming material, the thickness of incoming material, the entry and exit thickness of each roller, the entry and exit tension, the rolling temperature, and the rolling speed. Combined with the experimental data, it is determined that the input of the model is 30 dimensions and the output is 7 dimensions; that is, there are 7 rolling forces to be predicted.

4.2. Evaluation and Performance Analysis. When GA is used to determine the network structure of MELM, because the number of hidden layers and the number of hidden layer nodes must be positive integers, the genetic algorithm adopts binary coding. The parameters of GA are set as follows: the population iteration number is 100, the population size is 20, the crossover probability is set at 0.4, and the mutation probability is set at 0.2. The method selected in the selection operation is the roulette method. According to the data used in this experiment, we determined that the number of hidden layers of the model should be selected between 1-15, and the number of hidden nodes should be selected between 1-31. Therefore, the binary encoding length is 9, and the absolute value of the average prediction error of the model is selected as the evaluation index. The parameters of PSO are set as follows: the number of population iterations is 200, the population size is 20, the maximum and minimum of individuals are 1 and -1, and the maximum and minimum values of speed are 0.3 and -0.3, respectively.

In this section, several experiments in rolling force prediction are carried out to show the better performance do the proposed PSO-GA-MELM. Those experiments are design from the aspects of the accuracy and stability of the proposed PSO-GA-MELM compared with PSO-ELM and DBN. DBN is a deep network model with multiple hidden layers. Each layer of DBN is composed of a Restricted Boltzmann Machines (RBM). This method further solves the problem of gradient disappearance in deep training, which has been troubled for many years, and leads the development boom of deep learning. In this experiment, the data input has 30 features, which are trained by the DBN network, and the final output is 10 features.

All the simulations in this experiment were conducted under the environment of MATLAB 2016b. The activation function adopted by MELM is the hyperbolic tangent function. Before using genetic algorithm to determine the MELM network structure, we first tested the prediction error of some MELM models with fixed network structure through a large number of experiments. The experimental results are shown in Table 1.

The data in the Table 1 represent the prediction error of the MELM model with fixed network structure, that is, the difference between the measured rolling force and the predicted rolling force. It can be seen from Table 1 that when the number of hidden layers is between 3 and 10, and the number of hidden nodes is between 10 and 15, the prediction error of the model is small, and the errors in other ranges are large. Therefore, we determined that the optimal number of hidden layers is between 3 and 10, and the optimal number of hidden nodes is between 10 and 15. On this basis, further experiments are conducted, and the experimental results are shown in Table 2.

By analyzing the data in Table 2, we can conclude that when the number of hidden layers is between 3 and 5 and the number of hidden nodes is between 10 and 12, the prediction error of the model is small. Therefore, for the experimental data in this paper, the optimal number of hidden layers in the MELM based rolling force prediction model is between 3 and 5, and the optimal number of hidden nodes is between 10

TABLE 1: The prediction error of MELM models with fixed network structure.

The number of hidden layers	The number of hidden nodes				
	10	15	20	25	30
3	360.24	433.33	400.78	360.77	418.95
5	250.65	282.39	328.18	424.44	623.37
8	357.71	1116.80	1116.80	1116.80	1116.80
10	307.03	1116.80	1116.80	1116.80	1116.80
13	608.08	1116.80	1116.80	1116.80	1116.80
15	1116.80	1116.80	1116.80	1116.80	1116.80

TABLE 2: The prediction error of MELM models with fixed network structure.

The number of hidden layers	The number of hidden nodes					
	10	11	12	13	14	15
3	360.24	325.78	276.04	369.20	330.46	433.33
4	368.18	299.67	288.20	349.05	320.38	314.27
5	250.65	291.97	288.29	292.69	335.23	282.39
6	406.11	353.54	420.48	318.03	261.71	362.73
7	370.04	295.40	351.66	380.54	478.04	481.86
8	357.71	266.36	374.05	550.01	2156.40	1116.80
9	390.99	458.34	587.47	1116.80	1116.80	1116.80
10	307.03	562.14	1116.80	1116.80	1116.80	1116.80

TABLE 3: The prediction error for the algorithms PSO-ELM, PSO-GA-MELM and DBN.

Model	Training error	Testing error
PSO-ELM	599.91	341.74
PSO-GA-MELM	328.73	229.76
DBN	379.75	362.38

and 12. The network structure of the rolling force prediction model based on PSO-GA-MELM algorithm is as follows: the optimal number of hidden layers is 5, and the optimal number of hidden nodes is 11.

In order to investigate the improvement of learning accuracy of the proposed PSO-GA-MELM approach, the PSO-ELM based rolling force prediction model and the DBN based rolling force prediction model are also evaluated, and the number of hidden nodes in the PSO-ELM model is 17. The prediction errors for the algorithms PSO-ELM, PSO-GA-MELM, and DBN are shown in Table 3.

As observed from Table 3, the proposed PSO-GA-MELM algorithm achieves a lower prediction error for rolling force prediction in hot continuous rolling process, relative to the PSO-ELM and DBN techniques. PSO-GA-MELM adds hidden layer and sets hidden layer parameters; by making the actual hidden layer output approach the expected hidden layer output, a better mapping method between input signal and output signal is finally found, which improves the prediction accuracy of the model.

Prediction results of rolling force prediction model based on PSO-ELM are shown in Figure 4. Prediction results of

rolling force prediction model based on PSO-GA-MELM are shown in Figure 5. Prediction results of rolling force prediction model based on DBN are shown in Figure 6. The average testing error for the algorithm PSO-ELM, PSO-GA-MELM, and DBN is shown in Figure 7.

The rolling force prediction model is established to predict the rolling force through the rolling conditions, and the change of rolling force also reflects the change of rolling conditions, such as rolling temperature, incoming material thickness, and other factors. From the above experimental results, we can be concluded that both the prediction accuracy and stability of the PSO-GA-MELM algorithm are dramatically superior to those of the PSO-ELM and DBN algorithms when using fewer hidden nodes. It is therefore inferred that the proposed PSO-GA-MELM algorithm achieves a superior performance under the conditions where there is relatively small number of hidden nodes. These results further indicate the effectiveness of adopting the novel PSO-GA-MELM approach for the rolling force prediction in hot continuous rolling process.

5. Conclusion

As an improved model of MELM that can automatically determine the optimal network structure, PSO-GA-MELM uses GA to determine the optimal number of hidden layers and the number of nodes matching the optimal hidden layers. Meanwhile, by introducing PSO to search for the optimal input weights and biases of MELM, PSO-GA-MELM algorithm has significantly improved its generalization ability and calculation stability compared with the traditional neural

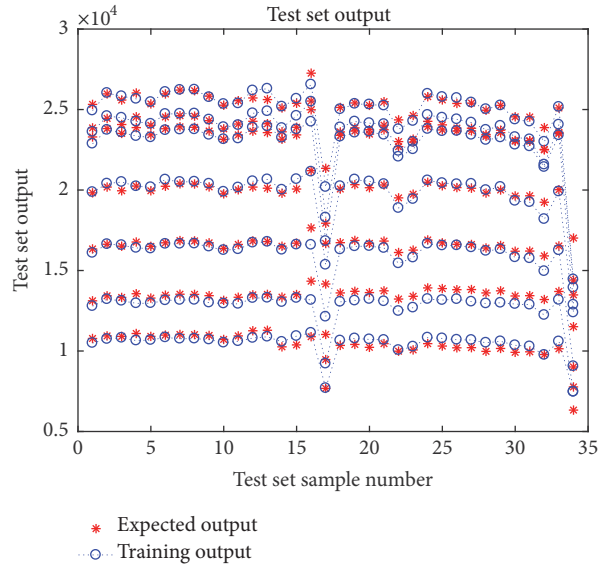


FIGURE 4: Prediction results of PSO-ELM based rolling force prediction model.

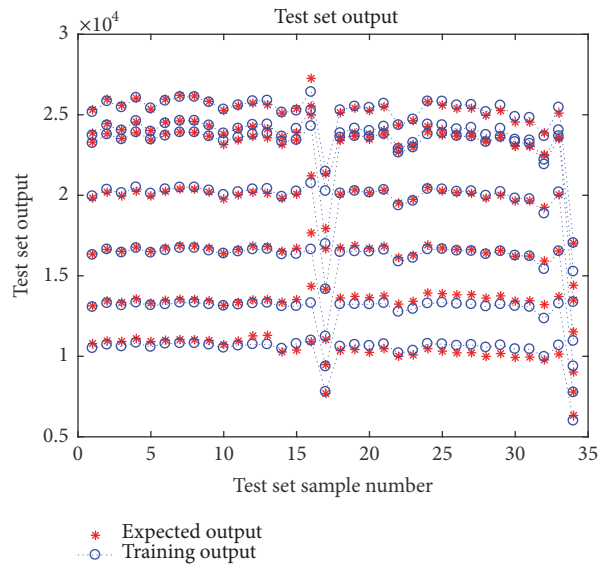


FIGURE 5: Prediction results of PSO-GA-MELM based rolling force prediction model.

network model. The influence of human intervention training process on the prediction accuracy and calculation stability of the model is effectively avoided.

Experiments on rolling force prediction in hot continuous rolling demonstrate that PSO-GA-MELM can effectively determine the optimal network structure of MELM and has the advantages of high prediction accuracy and computational stability, which can provide a novel and efficient solution for rolling force prediction in hot continuous rolling. The average testing error percentage of the proposed PSO-GA-MELM algorithm is distinctly lower than those of the PSO-ELM and DBN techniques when using fewer hidden

nodes. Therefore, this technique is a particularly attractive alternative for solving complex practical application problems in the presence of limited computational storage resources. In particular, the PSO-GA-MELM approach can able to deliver improved accuracy in applications where the number of hidden nodes that can be assigned is limited by hardware limitations.

In practical application, this paper only discusses the application of extreme learning machine in the modeling of hot continuous rolling process. In fact, for all the complex industrial processes which are difficult to establish the mechanism model, the extreme learning machine should be a new

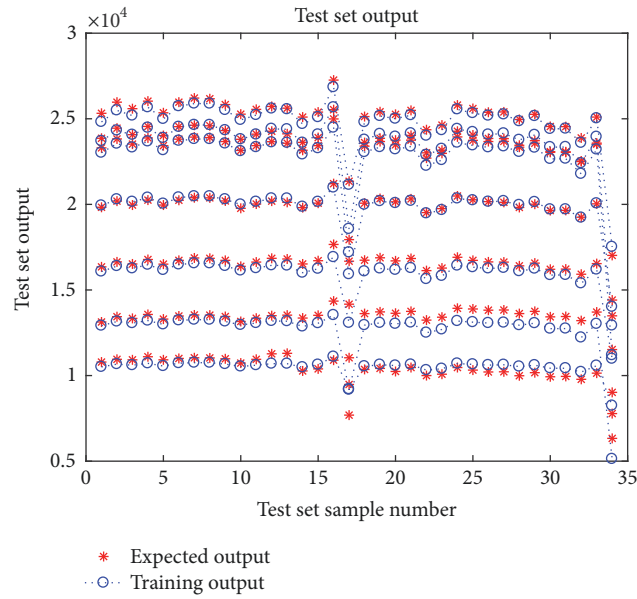


FIGURE 6: Prediction results of DBN based rolling force prediction model.

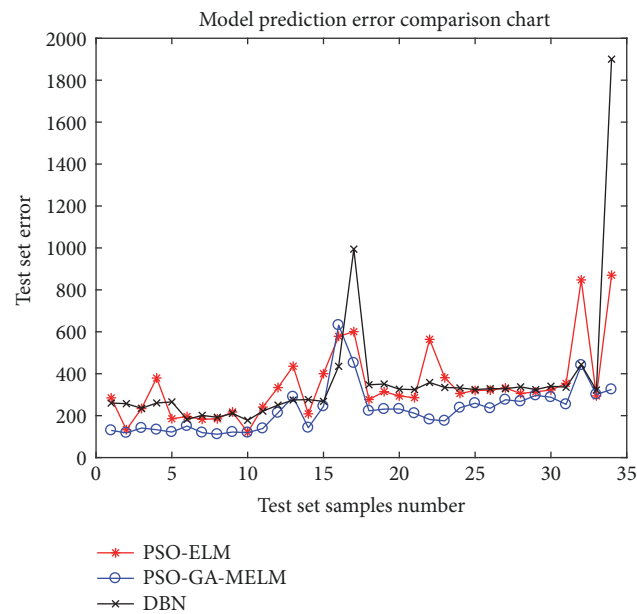


FIGURE 7: Average testing error for the algorithms PSO-ELM, PSO-GA-MELM, and DBN.

algorithm worth trying. This is helpful for further discovering and solving potential new problems in the algorithm of extreme learning machine and improving its theoretical framework. In addition, there is a very important problem to be solved in practical application, that is, how to better use the extreme learning machine, such a learning algorithm of static mapping relation, to model the dynamic industrial process which is essentially a dynamic evolutionary process.

Data Availability

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

Conflicts of Interest

All authors declare that they have no conflicts of interest.

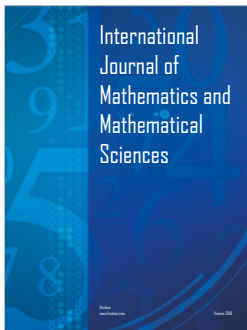
Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 71672032, 61773105, 61374147, and 61733003, in part by the Fundamental Research Funds for Central University under Grants N150402001, N180404012, and N182608003, and in part by the National Key Research and Development Program of China under Grant 2017YFB0304100.

References

- [1] M. Ataka, "Rolling technology and theory for the last 100 years: the contribution of theory to innovation in strip rolling technology," *ISIJ International*, vol. 55, no. 1, pp. 89–102, 2015.
- [2] T. Morimoto, F. Yoshida, I. Chikushi, and J. Yanagimoto, "Combined macro-micro modeling for rolling force and microstructure evolution to produce fine grain hot strip in tandem hot strip rolling," *ISIJ International*, vol. 47, no. 10, pp. 1475–1484, 2007.
- [3] X. H. Liu, Q. L. Zhao, and Z. Y. Huang, "Prospect of artificial intelligent application in rolling," *Steel Rolling*, vol. 34, no. 4, pp. 1–5, 2017.
- [4] Y.-S. Kim, B.-J. Yum, and M. Kim, "Robust design of artificial neural network for roll force prediction in hot strip mill," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN'01)*, pp. 2800–2804, Washington, DC, USA, July 2001.
- [5] M. Bagheripoor and H. Bisadi, "Application of artificial neural networks for the prediction of roll force and roll torque in hot strip rolling process," *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 4593–4607, 2013.
- [6] J. H. Liu, G. X. Wang, and Y. K. Liu, "Prediction of rolling force based on grey theory and neural network," *Forging and Stamping Technology*, vol. 40, no. 10, pp. 126–129, 2015.
- [7] B. Dou, "Prediction of rolling force and model optimization with bayes neural network," *Mental Products*, vol. 43, no. 6, pp. 42–48, 2017.
- [8] D. S. Wu, Q. Yang, and A. Z. Wang, "Rolling force prediction based on PSO optimized support vector regression," in *Proceedings of the IEEE International Conference on Natural Computation*, pp. 839–843, Shanghai, China, 2011.
- [9] X. Q. Zhao and Y. S. Wang, "BP Neural Network Based GPSA used in tandem cold rolling force prediction," in *Proceedings of the IEEE International Conference on Consumer Electronics, Communications and Networks*, pp. 4829–4832, Xianning, China, 2011.
- [10] Y. Liu, C. Tong, and F. Lin, "Rolling force prediction based on wavelet multiple-RBF neural network," in *Proceedings of the 2012 IEEE International Conference on Information and Automation, ICIA 2012*, pp. 941–944, Shenyang, China, June 2012.
- [11] Z. Chen and Z. Luo, "Rolling force prediction based on multiple support vector machines," in *Proceedings of the 32nd Chinese Control Conference, CCC 2013*, pp. 3306–3309, Xi'an, China, July 2013.
- [12] S. Zhang, W. L. Jiang, and J. H. Zhang, "SVM prediction-based rolling force setting calculation method of tandem cold rolling mill," *Chinese Hydraulics & Pneumatics*, no. 7, pp. 50–55, 2017.
- [13] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [14] B. Qu, B. Lang, J. Liang, A. Qin, and O. Crisalle, "Two-hidden-layer extreme learning machine for regression and classification," *Neurocomputing*, vol. 175, pp. 826–834, 2016.
- [15] K. Ning, M. Liu, M. Dong, C. Wu, and Z. Wu, "Two efficient twin ELM methods with prediction interval," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 2058–2071, 2015.
- [16] S. F. Ding, N. Zhang, X. Xu, L. Guo, and J. Zhang, "Deep extreme learning machine and its application in EEG classification," *Mathematical Problems in Engineering*, vol. 2015, Article ID 129021, 11 pages, 2015.
- [17] D. Xiao, B. J. Li, and Y. C. Mao, "A multiple hidden layers extreme learning machine method and its application," *Mathematical Problems in Engineering*, vol. 2017, Article ID 4670187, 10 pages, 2017.
- [18] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 20, no. 8, pp. 1352–1357, 2009.
- [19] Y. Lan, Y. C. Soh, and G. B. Huang, "Random search enhancement of error minimized extreme learning machine," in *Proceedings of the European Symposium on Artificial Neural Networks*, pp. 327–332, Bruques, Belgium, 2010.
- [20] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, no. 1–3, pp. 359–366, 2008.
- [21] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: optimally pruned extreme learning machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 21, no. 1, pp. 158–162, 2010.
- [22] Y. Lan, Y. C. Soh, and G. B. Huang, "Two-stage extreme learning machine for regression," *Neurocomputing*, vol. 73, no. 16–18, pp. 3028–3038, 2010.
- [23] G. B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16–18, pp. 3056–3062, 2007.
- [24] G. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, no. 16–18, pp. 3460–3468, 2008.
- [25] D. Xiao, B. Li, and S. Zhang, "An online sequential multiple hidden layers extreme learning machine method with forgetting mechanism," *Chemometrics and Intelligent Laboratory Systems*, vol. 176, pp. 126–133, 2018.
- [26] N. Liang, G. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 6, pp. 1411–1423, 2006.
- [27] G. Huang, L. Chen, and C. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 4, pp. 879–892, 2000.
- [28] G. Huang, G. B. Huang, S. J. Song, and K. Y. You, "Trends in extreme learning machines: a review," *Neural Networks*, vol. 61, pp. 32–48, 2015.
- [29] S. Boriratrut, S. Chiewchanwattana, K. Sunat, P. Musikawan, and P. Horata, "Improvement flower pollination extreme learning machine based on meta-learning," in *Proceedings of the 13th International Joint Conference on Computer Science and Software Engineering, JCSSE 2016*, Khon Kaen, Thailand, July 2016.
- [30] J. Tang, C. Deng, and G. B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 27, no. 4, pp. 809–821, 2016.
- [31] J. Zhang, W. Xiao, Y. Li, and S. Zhang, "Residual compensation extreme learning machine for regression," *Neurocomputing*, vol. 311, pp. 126–136, 2018.
- [32] W. Zong and G.-B. Huang, "Face recognition based on extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2541–2551, 2011.
- [33] M. Van Heeswijk, Y. Miche, T. Lindh-Knuutila et al., "Adaptive ensemble models of extreme learning machines for time series prediction," in *Lectures Notes in Computer Science*, vol. 5769, pp. 305–314, 2009.

- [34] H.-X. Tian and Z.-Z. Mao, "An ensemble ELM based on modified AdaBoost.RT algorithm for predicting the temperature of molten steel in ladle furnace," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 73–80, 2010.
- [35] Q. Yuan, W. Zhou, S. Li, and D. Cai, "Epileptic EEG classification based on extreme learning machine and nonlinear features," *Epilepsy Research*, vol. 96, no. 1-2, pp. 29–38, 2011.
- [36] V. Malathi, N. S. Marimuthu, S. Baskar, and K. Ramar, "Application of extreme learning machine for series compensated transmission line protection," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 5, pp. 880–887, 2011.
- [37] G. Feng, Z. Qian, and N. Dai, "Reversible watermarking via extreme learning machine prediction," *Neurocomputing*, vol. 82, no. 4, pp. 62–68, 2012.
- [38] Y. Liu, H. T. Loh, and S. B. Tor, "Comparison of extreme learning machine with support vector machine for text classification," in *Proceedings of the 18th International Conference on Innovations in Applied Artificial Intelligence*, vol. 3533, pp. 390–399, Bari, Italy, 2005.
- [39] F. L. Chen and T. Y. Ou, "Sales forecasting system based on Gray extreme learning machine with Taguchi method in retail industry," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1336–1345, 2011.
- [40] C.-W. T. Yeu, M.-H. Lim, G.-B. Huang, A. Agarwal, and Y.-S. Ong, "A new machine learning paradigm for terrain reconstruction," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 3, pp. 382–386, 2000.
- [41] A. Jain and N. S. Chaudhari, "An improved genetic algorithm for developing deterministic OTP key generator," *Complexity*, vol. 2017, Article ID 7436709, 17 pages, 2017.
- [42] S. Chun and T. Kim, "Simultaneous identification of model structure and the associated parameters for linear systems based on particle swarm optimization," *Complexity*, vol. 2018, Article ID 2713684, 17 pages, 2018.
- [43] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.




Hindawi

Submit your manuscripts at
www.hindawi.com

