

Research Article

Prediction of Bending Force in the Hot Strip Rolling Process Using Multilayer Extreme Learning Machine

Yan Wu ¹, Hongchao Ni,² Xu Li ³, Feng Luan ² and Yaodong He³

¹School of Metallurgy, Northeastern University, Shenyang 110819, Liaoning, China

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

³The State Key Laboratory of Rolling and Automation, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Xu Li; lixu@ral.neu.edu.cn and Feng Luan; luanfeng@mail.neu.edu.cn

Received 23 October 2020; Revised 28 December 2020; Accepted 6 January 2021; Published 19 January 2021

Academic Editor: Bekir Sahin

Copyright © 2021 Yan Wu 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 strip rolling process, accurate prediction of bending force can improve the control accuracy of the strip flatness and further improve the quality of the strip. In this paper, based on the production data of 1300 pieces of strip collected from a hot rolling factory, a series of bending force prediction models based on an extreme learning machine (ELM) are proposed. To acquire the optimal model, the parameter settings of the models were investigated, including hidden layer nodes, activation function, population size, crossover probability, and hidden layer structure. Four models are established, one hidden layer ELM model, an optimized ELM model (GAELM) by genetic algorithm (GA), an optimized ELM model (SGELM) by hybrid simulated annealing (SA) and GA, and two-hidden layer optimized ELM model (SGITELM) optimized by SA and GA. The prediction performance is evaluated from the mean absolute error (MAE), root-mean-squared error (RMSE), and mean absolute percentage error (MAPE). The results show that the SGITELM has the highest prediction accuracy in the four models. The RMSE of the proposed SGITELM is 11.2678 kN, and 98.72% of the prediction data have an absolute error of less than 25 kN. This indicates that the proposed SGITELM with strong learning ability and generalization performance can be well applied to hot rolling production.

1. Introduction

In hot strip rolling (HSR) process, the product quality of the strip is mainly contributed by dimensional accuracy, mechanical properties, and surface properties. Here, the dimensional accuracy of the strip requires two important indicators: strip thickness and surface profile [1]. The surface profile of the strip is defined as the difference of thickness between the center and a point of 40 mm from the edge of the strip, or in other words, this is the difference of thickness across the width of the strip [2]. There are many factors that affect the surface profile of the strip, which are mainly related to the roller, strip, and rolling conditions in the HSR process [3].

The hydraulic bending roll control is one of the main methods for hot strip profile control. By applying a certain bending force on the work roll, the strip profile can be improved. The schematic diagram of the hydraulic roll

bending technique is shown in Figure 1. The hydraulic roll gap control device can be used as the actuator of the leveling operation. How to set the appropriate bending force, which can affect the strip head and tail end walking behavior and strip profile, is the urgent problem to be solved. Therefore, the prediction accuracy of the roll bending force directly affects the strip profile control accuracy. The prediction accuracy is high, which is beneficial to the closed-loop feedback control of bending force. In the actual production process, the bending force is calculated according to the requirements of temperature, thickness, width, rolling force, material, roll thermal expansion, roll wear, and target profile [4].

Generally, the initial value setting of bending force is very complicated and is a multivariable optimization problem. At the same time, the adjustment of bending force needs to be calculated according to the strip profile at the exit. Some rolling parameters related to the bending force

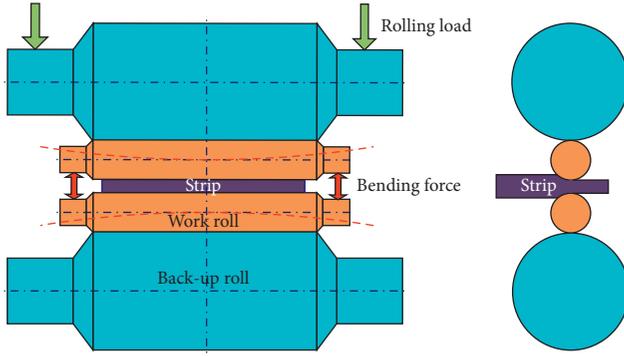


FIGURE 1: The schematic diagram of hydraulic roll bending technique.

model have the characteristics of nonlinearity, strong coupling, large detection error, etc. Therefore, the mathematical model established by the traditional theory has the disadvantages of slow response speed and low control accuracy in production practice. All these problems seriously restrict the further improvement of strip profile control accuracy. With the continuous improvement of rolling speed, the rolling process control system should be improved to adapt to the changes of the rolling process and the improvement of product requirements. Therefore, the high-precision prediction method of rolling process based on industrial big data has attracted attention [5].

Artificial neural network (ANN) is the earliest and most widely used data-driven method. ANN has been widely used in the field of metallurgy and materials because of its ability of parallel information processing, function approximation, self-learning, self-adaptation, and fault tolerance [6–8]. In the HSR process, ANN was proposed to predict the roll force [9–12], the strip profile and flatness [2, 13–15], and mechanical properties [16–18]. With the continuous development of ANN prediction in HSR, more application backgrounds have been continuously developed. For the first time, Wang et al. [4] have established an ANN model to predict the bending force of strip. To improve the performance of the model, it has also been optimized with a genetic algorithm (GA) and a GA-ANN model with good prediction performance has been obtained.

The ANN method improves the performance of prediction to a certain extent; there are still some problems, such as the slow learning speed of the algorithm for training models and the difficulty of adjusting numerous parameters. Therefore, the new machine learning algorithm is applied to various prediction problems in the rolling field. Huang et al. [19] proposed a new learning algorithm called extreme learning machine (ELM), the single-hidden layer feedforward neural networks (SLFNs), which randomly chose the input weights and determined the output weights of SLFNs by least-squares solution. Compared with the traditional gradient-based approaches, the ELM has a significantly faster learning speed and presents better generalization

performance. ELM has been widely used in metallurgy and metal processing fields, such as the mechanical properties of hot rolling products [20], rolling force prediction in HSR [21], silicon content prediction [22], alumina concentration detection [23], gas utilization ratio prediction in blast furnace [24], and tool fault diagnosis in numerical control machines [25].

In this paper, ELM is first applied to predict the bending force of the strip in HSR. Furthermore, three improved ELM-based models are developed to realize the high precision and high stability prediction of bending force. Section 2 introduces the theory and optimization strategies of the ELM model. Section 3 shows the experimental data, prediction results, and discussion. Finally, the conclusion is drawn in Section 4.

2. ELM-Based Methods

2.1. ELM. The basic theory of the ELM model states that, for N arbitrary distinct samples (x_i, t_i) , where $x_i = [x_{i1}, x_{i2}, \dots, x_{im}]^T \in R_m$ and $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in R_m$ are the model input and the target data, standard SLFNs with \tilde{N} hidden layer nodes and an activation function $g(x)$ are mathematically described as follows:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = o_j, \quad j = 1, \dots, N, \quad (1)$$

where $w_i = [w_{i1}, w_{i2}, \dots, w_{im}]^T$ is the weight vector connecting the i th hidden node and the input nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ is the weight vector connecting the i th hidden node and the output nodes, $o_i = [o_{i1}, o_{i2}, \dots, o_{im}]^T$ is the model output, and b_i are the bias parameters of the i th hidden node. $w_i \cdot x_j$ denotes the inner product of w_i and x_j . The w_i and b_i parameters can be randomly assigned if the activation function is infinitely differentiable, so only the β parameters need to be optimized when minimizing the mean squared error between the model output and the target data. In detail, the standard SLFNs with \tilde{N} hidden nodes with activation function $g(x)$ can approximate these N samples with zero error means that $\sum_{i=1}^{\tilde{N}} \|o_i - t_i\| = 0$. Therefore, in the ELM approach, training an SLFN is equivalent to simply finding the least-squares solution of the linear system, which can be written compactly as follows:

$$H\beta = T, \quad (2)$$

where the hidden layer output matrix H is

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \dots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}, \quad (3)$$

the output weight matrix β and the target data matrix T are

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix}_{N \times m}, \quad (4)$$

$$T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}.$$

Algebraically, the linear system for β is solved via the Moore–Penrose generalized inverse H^{-1} .

$$\hat{\beta} = H^{-1}T. \quad (5)$$

The principle which distinguishes ELM from the traditional neural network methodology is that the parameters of the feedforward networks (input weights and hidden layer biases) are not required to be tuned in the former. The studies of Tamura and Tateishi [26] and Huang et al. [27] showed in their works that the SLFNs with randomly chosen input weights efficiently learn distinct training examples with a minimum error. After randomly choosing the input weights and the hidden layer biases, SLFNs can be simply considered as a linear system. The output weights which link the hidden layer to the output layer of this linear system can now be analytically determined through a simple generalized inverse operation of the hidden layer output matrices. This simplified approach makes the ELM model many times faster than that of ANN [27]. Figure 2 shows the basic schematic topological structure of an ELM network.

2.2. ELM Optimization by GA (GAELM). Because the initialization of the weights and biases is randomly assigned in the ELM algorithm, it will not lead to the optimal state of the network during the training process. Therefore, in this study, we introduce the GA algorithm to optimize the weights and biases of the ELM network. GA is a parallel random search optimization method based on the natural genetic mechanism and natural selection theory in the biological world, and it is very suitable for complex nonlinear optimization problems [28]. The GA selects, crosses, and mutates the operator according to a random initial set of solutions and iteratively generates new solutions. After a certain number of algebras, a global optimal solution is obtained.

The select operator means selecting individuals with a strong vitality in the group. The roulette wheel selection method is adopted, and which formula is as follows:

$$P_i = \frac{F_i}{\sum_{i=1}^N F_i}, \quad (6)$$

where F_i is the fitness value of i th individual and N is the number of population.

The cross operation in GA is that two pairs of chromosomes (individuals) are exchanged with each other in some way. The cross operation of the node A_k of the k th chromosome and the node A_l of the l th chromosome at the j position is as follows:

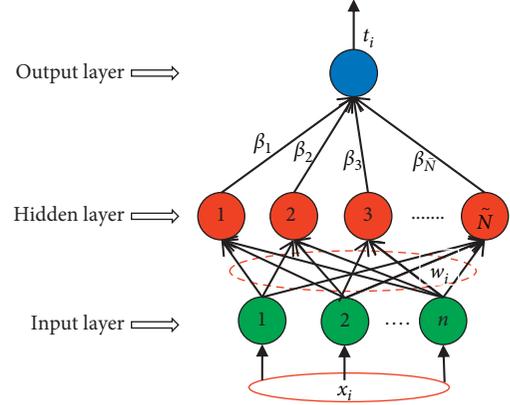


FIGURE 2: The basic schematic topological structure of an ELM network.

$$A_{kj} = A_{kj}(1 - \eta) + A_{lj}\eta, \quad (7)$$

$$A_{lj} = A_{lj}(1 - \eta) + A_{kj}\eta,$$

where η is a random number between 0 and 1. Cross operation is shown in Figure 3.

The mutate operation is achieved by flopping the randomly selected bits (see Figure 4), and the mutate probability p_m is usually small. The selection of an individual in a population is carried out by the evaluation of its fitness, and it can remain in the new generation if a certain threshold of fitness is reached. The Individuals with higher fitness are more likely to reproduce. Mutation operations are as follows:

$$A_{ij} = \{A_{ij} + (A_{ij} - A_{\max}) * f(g), \quad r_1 \geq 0.5\},$$

$$A_{ij} = \{A_{ij} + (A_{\min} - A_{ij}) * f(g), \quad r_1 < 0.5\}, \quad (8)$$

$$f(g) = r_2 * \left(1 - \frac{G}{G_{\max}}\right)^2,$$

where A_{\max} is the upper bound of A_{ij} ; A_{\min} is the lower bound of A_{ij} ; G is the current iteration time; G_{\max} is the maximum time of evolution; and r_1 and r_2 are the random number between 0 and 1. The overall algorithm process for optimizing ELM by using GA is shown in Figure 5.

2.3. GA with Simulated Annealing (SAGA). In the early stage of traditional GA, the individual difference is large. When the classic roulette method was used, the number of new individuals is proportional to the fitness of the original individuals. New individuals are easily flooded to the whole population, which causes early maturity. In the later stage, the fitness tends to be consistent, and the superior individuals do not have obvious advantages when they produce new individuals, which stops the evolution of the whole population [29]. Therefore, it is necessary to properly stretch the fitness. The simulated annealing (SA) algorithm proposed by Metropolis can realize the stretching of the fitness function [30]. In this study, SA was used to optimize the

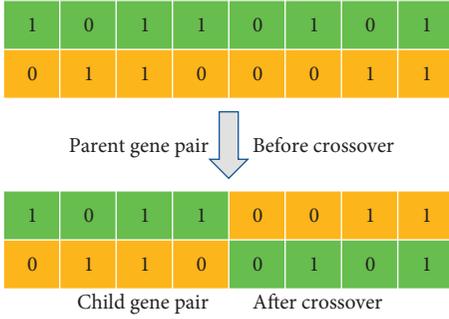


FIGURE 3: Diagram of crossover at a random crossover point in GA.

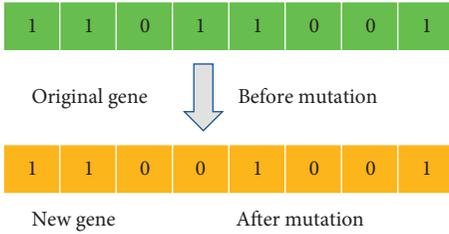


FIGURE 4: Schematic representation of mutation at a single site by flipping a randomly selected bit.

selection process of GA. The SA algorithm mainly includes the Metropolis criterion, that is, the probability of accepting the new solution of SA. As for the optimization problem of taking the minimum value for the objective function, the probability that the SA accepts the new solution is as follows:

$$p(x \Rightarrow x') = \begin{cases} 1, & f(x') < f(x), \\ \exp\left[-\frac{f(x') - f(x)}{T}\right], & f(x') \geq f(x), \end{cases} \quad (9)$$

where x represents the current solution; x' represents the new solution; $f(x)$ represents the objective function value of the current solution; $f(x')$ represents the objective function value of the new solution; and T represents temperature.

2.4. An Improved Multilayer ELM. Already in 1997, the results of Tamura and Tateishi showed that with the increase of network layer, the prediction accuracy will become higher and higher, and the number of hidden layer nodes will decrease greatly [26]. Qu et al. [31] proposed a two-layer ELM (TELM) neural network. The structure of the TELM is composed of an input layer, two-hidden layers, and an output layer, and the neurons between the layers are all connected. The TELM still retains some of the advantages of the ELM, such as strong generalization ability, fast operation speed, and very little chance of falling into overfitting. The principle of the TELM is as follows:

Randomly generate the first hidden layer input matrix weights W and biases B . The first hidden layer input parameter matrix is defined as $W_{IE} = [BW]$. And the augmented matrix

of the input matrix is defined as $X_E = [1X]^T$. Calculating the output matrix of the first hidden layer as

$$H_1 = g(W_{IE}X_E). \quad (10)$$

The output weights matrix between the first hidden layer and the final output layer according to the traditional ELM are calculated as

$$\beta = H_1^{-1}T. \quad (11)$$

The expected output matrix of the second hidden layer is calculated as

$$H_2 = T\beta^{-1}. \quad (12)$$

The augmented matrix of the first hidden layer output matrix $H_E = [1H_1]$ is constructed and the second hidden layer input parameters matrix is calculated as

$$W_{12} = g^{-1}(H_2)H_E^{-1}. \quad (13)$$

The output matrix of the second hidden layer is calculated as

$$H_2 = g(W_{12}H_E). \quad (14)$$

The output weight matrix between the second hidden layer and the final output layer is calculated as

$$\beta_{\text{new}} = H_2^{-1}T. \quad (15)$$

The final output of the network is

$$f(x) = g([W_{12}[1g(W_{IE}X_E)]])\beta_{\text{new}}, \quad (16)$$

where the superscript -1 represents the Moore–Penrose generalized inverse operation; g^{-1} represents the inverse function of g . The basic schematic topological structure of a TELM network is shown in Figure 6.

To get a more stable prediction output and better generalization performance, we calculate the output weight matrix β of TELM by considering the following three cases:

If the number of training samples is more than the number of hidden layer nodes, then

$$\beta = \left(\frac{I}{\lambda} + H^T H\right)^{-1} H^T T. \quad (17)$$

If the number of training samples is less than the number of hidden layer nodes, then

$$\beta = H^T \left(\frac{I}{\lambda} + H H^T\right)^{-1} T. \quad (18)$$

If the number of training samples is equal to the number of hidden layer nodes, then

$$\beta = H^{-1}T, \quad (19)$$

where λ is a uniformly distributed random number between 0 and 1 [32]. And this method is named as improved TELM (ITELM).

Based on these tasks, we optimized the ITELm network further by using the SAGA algorithm. We call it the

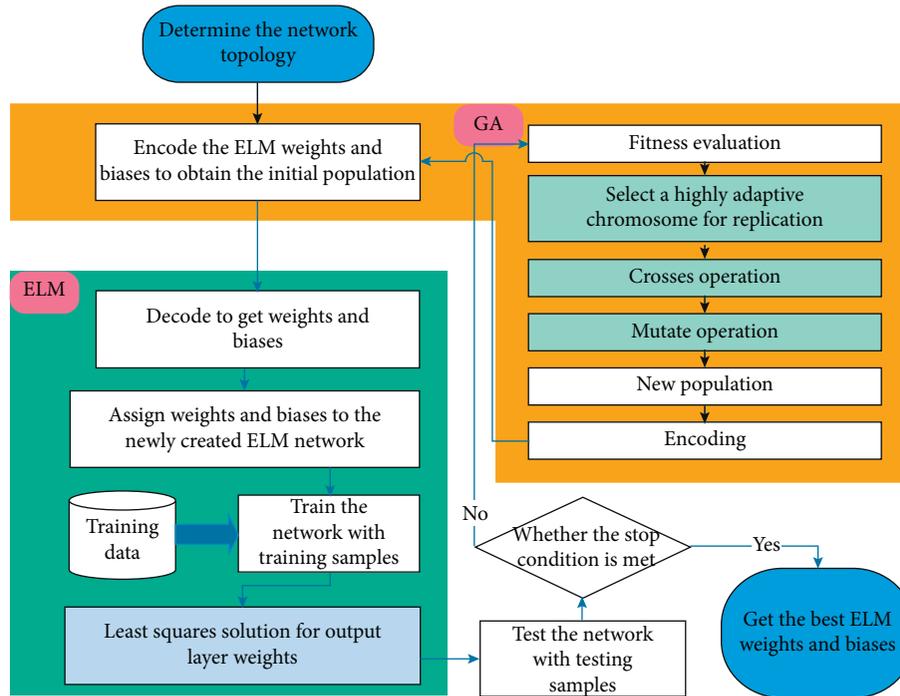


FIGURE 5: The flow chart for optimizing ELM by using GA.

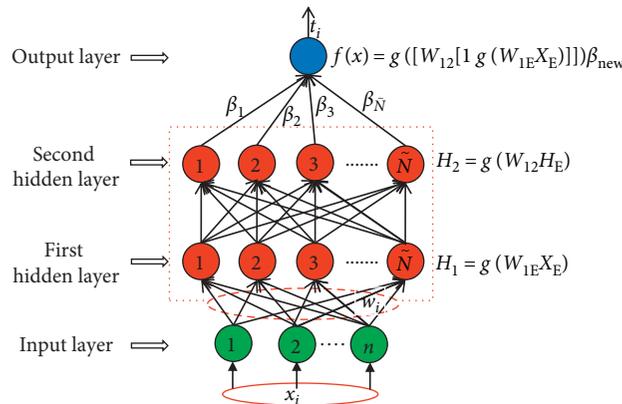


FIGURE 6: The basic schematic topological structure of a TELM network.

SGITELM algorithm. The flow chart of SGITELM algorithm is shown in Figure 7.

3. Experimental Results and Analysis

3.1. Data and Processing. In this paper, the final stand rolling data of a 1580 mm HSR process in a steel factory are collected for experiments. The input variables used for the proposed prediction model of bending force are entrance temperature ($^{\circ}\text{C}$), entrance thickness (mm), exit thickness (mm), strip width (mm), rolling force (kN), rolling speed (m/s), roll shifting (mm), yield strength (MPa), and target

profile (μm). The output variable of the model is the bending force (kN). A total of 1300 pieces of steel data are employed in the experiments, and the dataset is divided into the following two subsets: training set (70%) and testing set (30%). The training dataset is used in determining the model structure and selecting training parameters. In this paper, the testing dataset is used as an unseen validation dataset to verify the model performance. The fractal dimension visualization diagram of the collected dataset is shown in Figure 8. Clearly, the input data vary considerably in different dimensions. Table 1 shows the data distributions for each input variable. To eliminate the difference between the numbers of different dimensional data, avoid the prediction

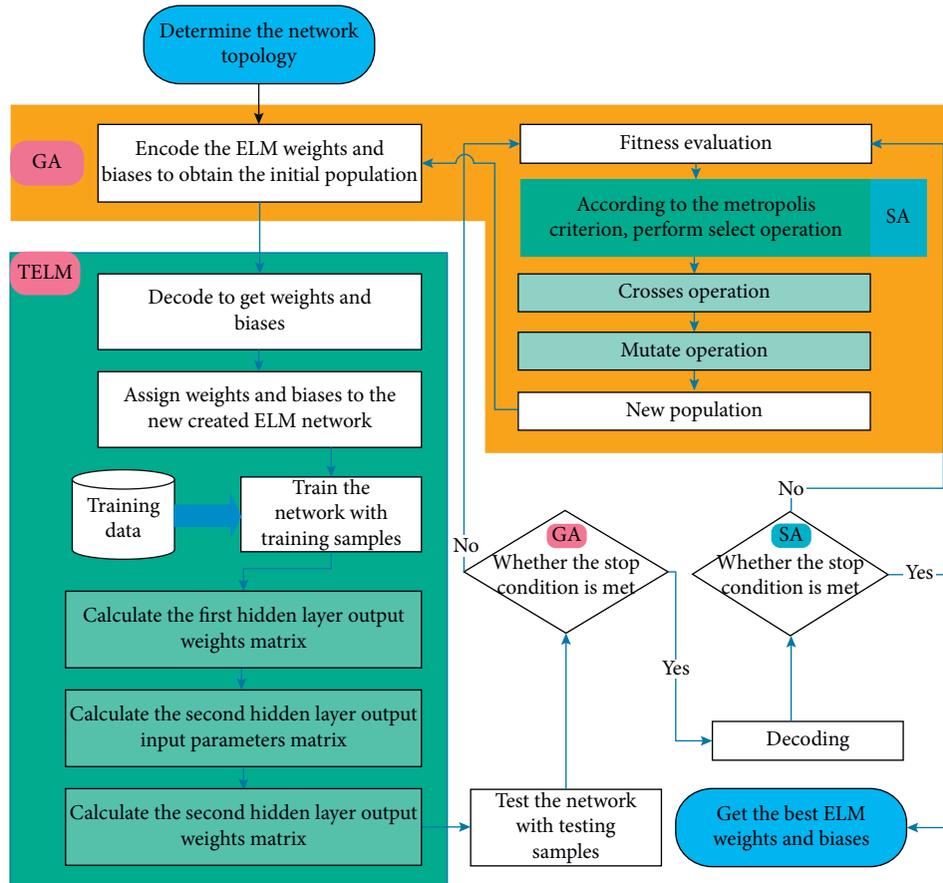


FIGURE 7: The flow chart of SGITELM.

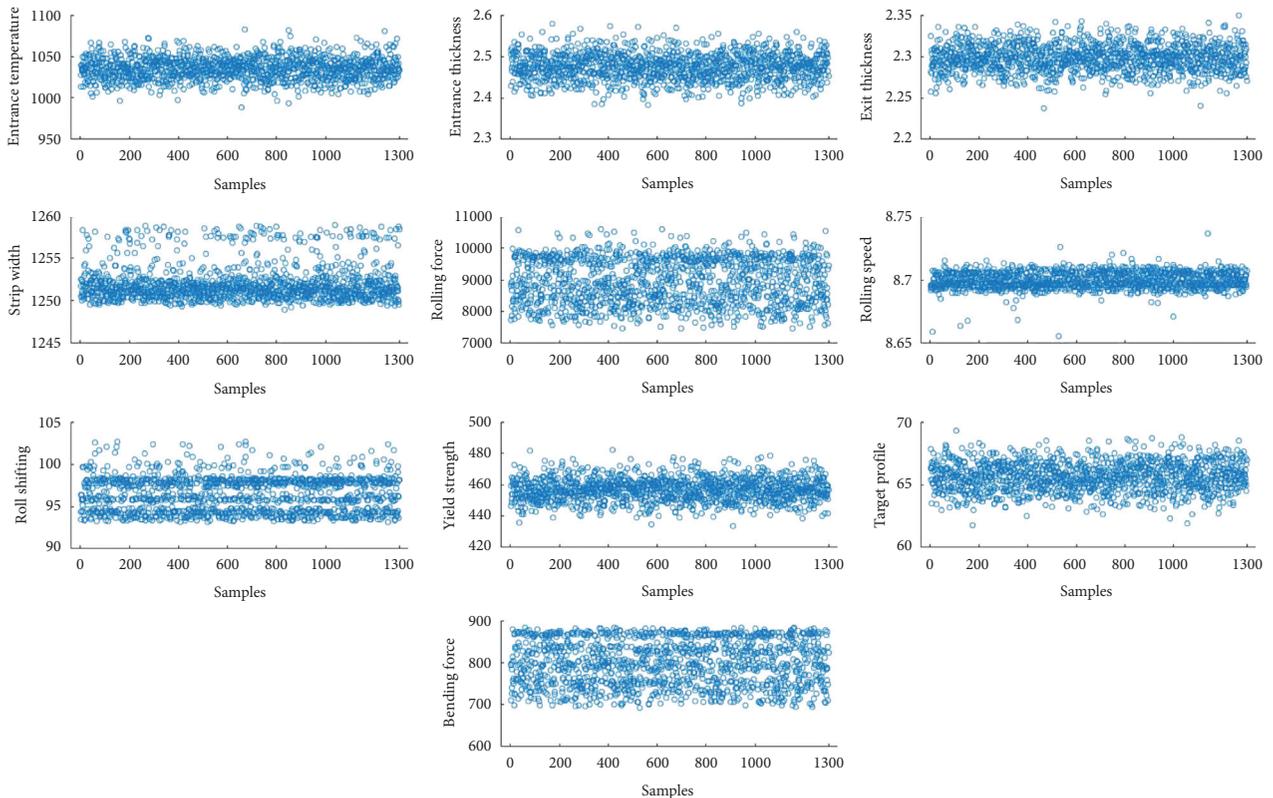


FIGURE 8: The fractal dimension visualization diagram of the data.

TABLE 1: The input parameters and values.

Parameter	Unit	Mean	Range of value
Entrance temperature	°C	1035.2	988.0505–1082.8000
Entrance thickness	mm	2.4751	2.3827–2.5782
Exit thickness	mm	2.2979	2.2374–2.3494
Strip width	mm	1252	1248.9–1259.0
Rolling force	kN	8941.623	7465.211–10590
Rolling speed	m/s	8.6995	8.6558–8.7362
Roll shifting	mm	96.42	93.13–102.63
Yield strength	MPa	456.7339	433.7365–482.0241
Target profile	μm	65.6	61.7–69.3

error increase because of the big difference between input and output data and update the weights and biases conveniently in the modeling process. It is necessary to scale the data to a small interval in a certain proportion. Normalization is required prior to data entry into the model [33]. The following formula is used to normalize the data:

$$y'_i = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}}, \quad (20)$$

where y'_i , y_i , y_{\min} , and y_{\max} are the normalized data, original data, maximal data, and minimal data, respectively.

3.2. Evaluation Criteria. Complete assessment of model performance is carried out by calculating the mean absolute error (MAE), root-mean-squared error (RMSE), and mean absolute percentage error (MAPE) on the testing dataset. The formulas of the three evaluation criteria are described as follows:

$$\begin{aligned} \text{MAE} &= \frac{\sum_{i=1}^n |x_i - y_i|}{n}, \\ \text{RMSE} &= \sqrt{\frac{\sum_{i=1}^n (x_i - y_i)^2}{n}}, \\ \text{MAPE} &= \frac{\sum_{i=1}^n (|x_i - y_i|/y_i)}{n} * 100, \end{aligned} \quad (21)$$

where n denotes the number of sample data and y_i and x_i are the measured value and the predicted value of the i th sample.

3.3. Determining the Best Configuration for ELM. ELM can fit nonlinear complex functions well, mainly because the activation function is used in the hidden layer. The activation function plays an important role in learning the model and understanding the very complex and nonlinear relationships. It can learn complex arbitrary function mappings that represent nonlinearities between input and output. For the commonly used activation functions “Radbas,” “sin,” “sigmoid,” “Hardlim,” and “Tribas,” this paper tests their impact on the performance of the prediction model. The formulas for these activation functions are as follows:

$$\text{Radbas}(x) = e^{-(x/2)},$$

$$\text{sin}(x) = \frac{e^{ix} - e^{-ix}}{2i},$$

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}, \quad (22)$$

$$\text{Hardlim}(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0, \end{cases}$$

$$\text{Tribas}(x) = \begin{cases} 1 - |x|, & -1 \leq x \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

Besides the activation function, the number of hidden layer nodes also plays a crucial role in the accuracy and generalization ability of the model. If there are too many hidden layer nodes, it will inevitably cause some initial node units to be invalid or redundant, which will greatly affect the generalization ability of the model. If the node is set too small, the accuracy of the model will be affected. We must reasonably control the number of nodes to reduce the generation of redundant nodes and ensure the prediction accuracy of the model. The results in Table 2 fully reflect the RMSE of activation functions and the number of hidden layer nodes of the ELM algorithm. Table 2 shows that the RMSE value corresponding to the “sigmoid” activation function is always the smallest in the case of the same number of hidden layer nodes. Table 2 also shows that when “sigmoid” is used as the activation function, the RMSE decreases first and then increases with the increase of the number of hidden layer nodes, which is consistent with the previous description. Although the RMSE of activation functions “Radbas,” “Hardlim,” and “Tribas” are still decreasing with the increase of the number of nodes, they are always larger than the RMSE of activation function “Sigmoid” when the number of hidden layer nodes is 90. For the three activation functions, when the number of hidden layer nodes is large enough, there may be a relatively small RMSE, but the increase of the number of hidden layer nodes means the increase of model complexity. Therefore, “sigmoid” is finally proposed as the activation function and 90 is the number of hidden layer nodes.

TABLE 2: The RMSE of different activation functions and node number combinations of ELM.

Activation function	Number of hidden layer node						
	10	40	80	90	100	140	180
Radbas	55.2617	19.0126	16.1132	15.8309	15.6568	15.2251	15.1812
Sin	55.2617	14.4886	13.9327	13.7969	13.9768	14.0701	14.4754
Sigmoid	55.2617	13.8739	13.7326	13.2417	13.7331	14.0202	14.3876
Hardlim	55.2617	23.6476	19.4240	18.5305	18.6029	17.5579	17.3333
Tribas	55.2617	24.0597	18.6536	18.1056	17.7518	17.1867	16.6234

3.4. *ELM Optimized by SAGA.* Among the SAGA parameters that must be determined, population size and crossover probability are discussed in this section. Large population size may lead to slow convergence speed, while a small population size can involve a local optimum point. Crossover probability is employed to determine if two individuals must be crossed or not. With the results shown in Table 3, the population size of 10 generates the lowest RMSE of 12.3072. The effect of crossover probability from 0.4 to 0.9 at intervals of 0.1 is listed in Table 4, and the optimal crossover probability is 0.7. Based on the results of the extensive and complex experiments carried out above, a convincing ELM model optimized by SAGA (SGELM) is finally established, and the parameters of each variable are listed in Table 5.

Figure 9 shows the optimization procedures of the algorithms including GAELM and SGELM. It can be observed that the fitness curve of GAELM completely converges after the 60th iteration, while the fitness curve of SGELM just starts to converge from the 86th iteration. However, it can also be observed that the final fitness value of SGELM is better than that of GAELM. It indicates that, although the convergence rate of SGELM is slow, the solution quality of SGELM is better than that of GAELM.

3.5. *Determining the Best Configuration for TELM.* The number of nodes in each hidden layer plays an important role in the capacity of TELM, and there is no accepted theory to determine it. In this study, the activation function with “sigmoid” is determined first and the number of nodes in each of the two hidden layers is set to be the same [32]. Based on the results shown in Table 6, a TELM with the hidden layer structure 40-40 has the lowest RMSE of 12.9550.

3.6. *Comparison with Different ELM-Based Methods.* The prediction results of ELM, GAELM, SGELM, SGITELM, and real bending force values are shown in Figure 10. For better visualization, only the predicted values for the 40 samples in the testing dataset are shown. As can be seen from Figure 10, the bending force values predicted by the four ELM-based models are relatively close to the measured bending force values. It shows that the four ELM-based models can be applied to the prediction of bending force in HSR. In addition, among the four models, ELM has relatively poor prediction performance, because its prediction value is farthest from the measured value.

The prediction performance of the ELM, GAELM, SGELM, and SGITELM is represented by the scatter plot, as shown in Figure 11. The color scale is used to grade the

TABLE 3: Effect of population size of SAGA.

Population size	RMSE
5	12.5720
8	12.3347
10	12.3072
13	12.3791
17	12.3343
25	12.4006
35	12.3805

TABLE 4: Effect of crossover probability of SAGA.

Crossover probability	RMSE
0.4	12.4726
0.5	12.5295
0.6	12.2501
0.7	12.0482
0.8	12.4107
0.9	12.5453

TABLE 5: Final parameters setting of SGELM.

Parameters of SGELM	Value
Number of input layer nodes	9
Number of hidden layer nodes	90
Number of output layer nodes	1
Activation function	Sigmoid
Population size	10
Selection probability	0.7
Crossover probability	0.7
Mutation probability	0.01
Cooling rate	0.8
Initial temperature	100
Final temperature	1

absolute error; as the color goes from red to blue, the absolute error increases from 0 to 25 kN and the pink spot symbols indicate an error of over 25 kN. For higher production requirements, the absolute error between the predicted and measured bending force is expected to be less than 25 kN. Therefore, the more pink spots the model has, the worse the prediction performance the model has. In Figure 11, the predicted values are evenly distributed on both sides of the diagonal line symmetrically. Among them, ELM has the largest number of pink spots and the SGITELM has the least number of pink spots; 98.72% of the absolute errors of SGITELM prediction results are less than 25 kN. It also

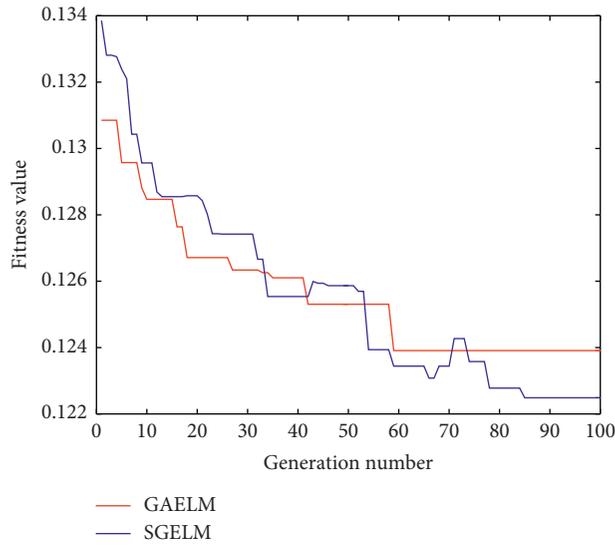


FIGURE 9: The curve of average fitness change.

TABLE 6: Effect of hidden layer structure of TELM.

Structure of hidden layers	RMSE
10-10	15.0758
30-30	13.0411
40-40	12.9550
50-50	12.9851
80-80	13.0522
120-120	13.2239
180-180	13.7523

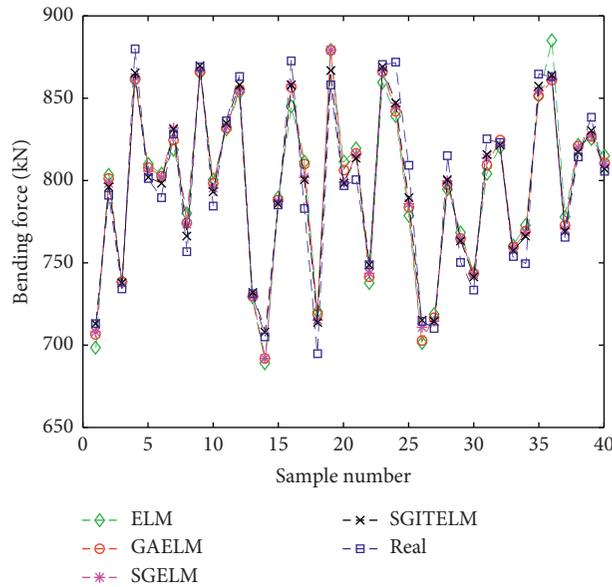


FIGURE 10: Comparison of predicted values of the proposed methods with the real measured values.

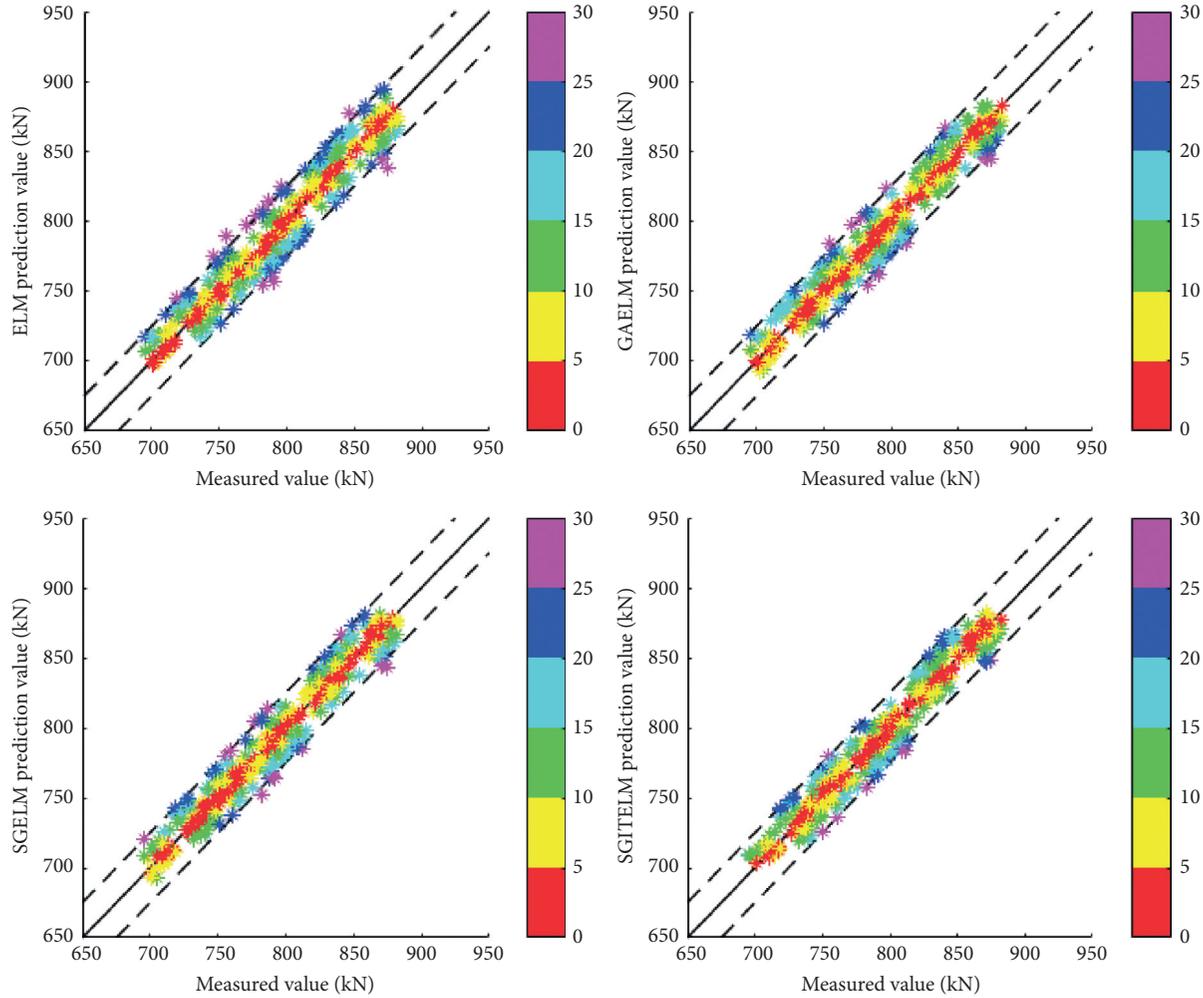


FIGURE 11: Scatter plot of prediction values for the proposed methods.

shows that the prediction performance is improved after the model is optimized.

Reasonable error distributions of the model are of great significance to analyze the feasibility of the model. Figure 12 is the histograms and distribution curves of the errors from the ELM, GAELM, SGELM, and SGITELM. The error distribution curves have a bell shape of normal distribution, which indicates that the prediction errors of all models are normal distribution. SGITELM performs relatively well, and their normal distribution curves are higher and narrower with the smallest standard deviation σ , which indicates that more prediction values with smaller errors are obtained. These results further prove the superiority of the prediction performance of the SGITELM algorithm.

Figure 13 shows the boxplot for the relative error results of the proposed methods. In general, the relative error reflects the deviation between the predicted value and the measured value, which better reflects the reliability of the

prediction performance than the absolute error. The boxplot represents the degree of spread for the relative error with its respective quartile. SGITELM has the least outliers and the smallest quartile, which indicates that SGITELM has advantages over other bending force prediction methods.

To evaluate the prediction accuracy of these ELM-based methods more intuitively, the results of the three evaluation criteria are given in Figure 14. The results show that the accuracy ranking results of the four models under MAE, MAPE, and RMSE are the same; the evaluation criteria in ascending order is $ELM > GAELM > SGELM > SGITELM$. The proposed SGITELM has the best prediction performance, and the MAE, MAPE, and RMSE of SGITELM are 9.0893, 1.1433%, and 11.2678, respectively. These results fully prove that the proposed SGITELM is more suitable for bending force prediction of the hot strip than traditional ELM methods because of its higher prediction accuracy and better generalization performance.

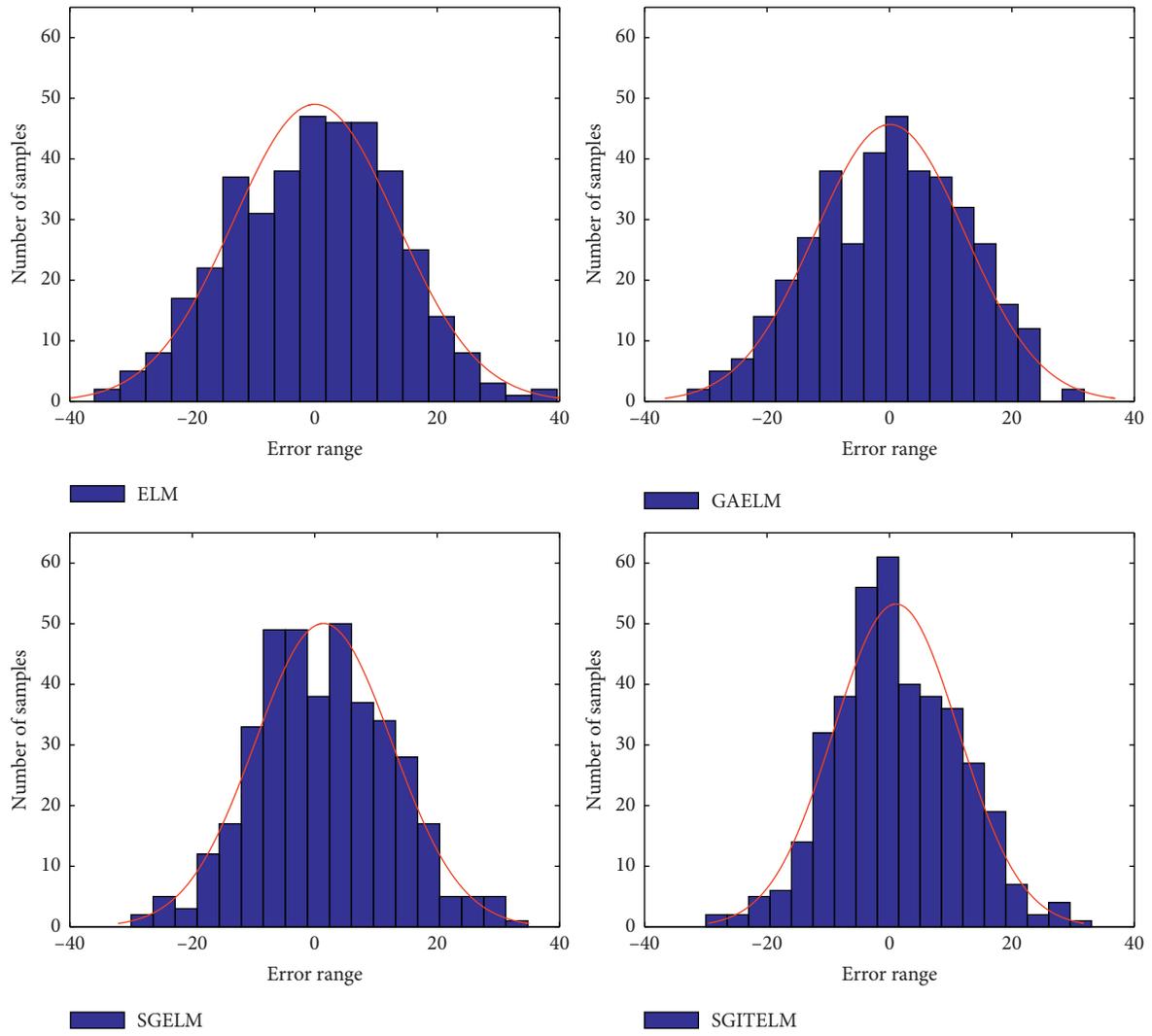


FIGURE 12: Error histograms for prediction of the proposed methods.

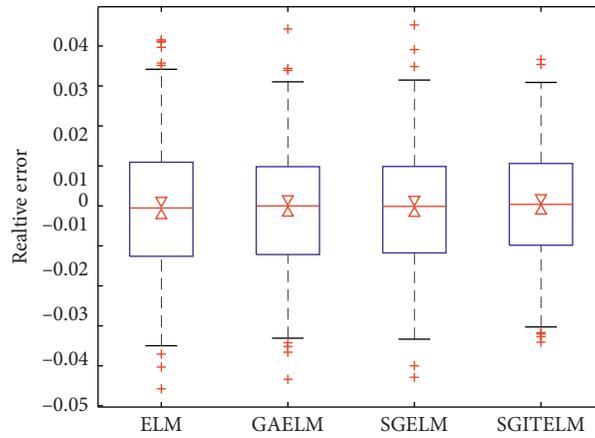


FIGURE 13: Boxplot for relative error of proposed methods.

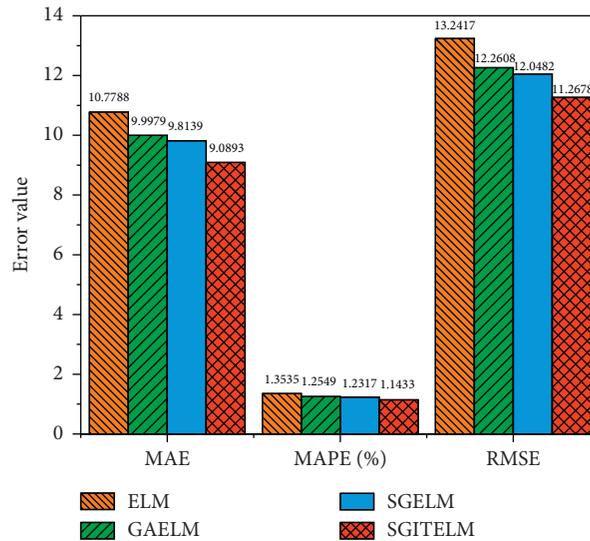


FIGURE 14: Results of prediction accuracy for proposed methods.

4. Conclusion

In this paper, four ELM-based methods to predict the bending force were proposed. A total of 1300 pieces of steel data were collected to train and test the models. Values of hidden layer nodes, activation functions, and hidden layer structure of ELM were separately tested to determine the structure of ELM. The prediction performance of ELM, GAELM, SGELM, and SGITELM was evaluated, and the prediction accuracy was compared with the three criteria of MAE, MAPE, and RMSE. The prediction accuracy of the ELM-based methods can be significantly influenced by different hidden nodes, activation functions, parameter settings of SAGA, and the hidden structure of the TELM. The improved SGITELM is the most recommended method, which has the highest prediction accuracy and the best generalization performance and can be recommended for bending force prediction in hot strip rolling. ELM-based methods can work well for bending force prediction in hot strip rolling. It is also recognized that if ELM-based methods are introduced into other predictions in the hot rolling industry, more production benefits and economic benefits may be obtained.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Key R&D Program of China (no. 2017YFB0304100), the National Natural Science Foundation of China (no. U20A20187), and

the Fundamental Research Funds of the Central Universities (nos. N180708009 and N2007006).

References

- [1] R. Takahashi, "State of the art in hot rolling process control," *Control Engineering Practice*, vol. 9, no. 9, pp. 987–993, 2001.
- [2] S. Sikdar and S. Kumari, "Neural network model of the profile of hot-rolled strip," *The International Journal of Advanced Manufacturing Technology*, vol. 42, no. 5-6, pp. 450–462, 2009.
- [3] H. Alaei, M. Salimi, and A. Nourani, "Online prediction of work roll thermal expansion in a hot rolling process by a neural network," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 5–8, p. 1769, 2015.
- [4] Z.-H. Wang, D.-Y. Gong, X. Li, G.-T. Li, and D.-H. Zhang, "Prediction of bending force in the hot strip rolling process using artificial neural network and genetic algorithm (ANN-GA)," *The International Journal of Advanced Manufacturing Technology*, vol. 93, no. 9–12, pp. 3325–3338, 2017.
- [5] Z. Y. Hu, Z. H. Wei, H. Sun, J. M. Yang, and L. X. Wei, "Optimization of metal rolling control using soft computing approaches: a review," *Archives of Computational Methods in Engineering*, In press.
- [6] F. Wang, X. Liu, G. Deng, X. Yu, H. Li, and Q. Han, "Remaining life prediction method for rolling bearing based on the long short-term memory network," *Neural Processing Letters*, vol. 50, no. 3, pp. 2437–2454, 2019.
- [7] C. W. Kan and L. J. Song, "An artificial neural network model for prediction of colour properties of knitted fabrics induced by laser engraving," *Neural Processing Letters*, vol. 44, no. 3, pp. 639–650, 2016.
- [8] S. W. Wu, X. G. Zhou, G. M. Cao, N. A. Shi, and Z. Y. Liu, "High-dimensional data-driven optimal design for hot strip rolling of microalloyed steel," *Steel Research International*, vol. 89, no. 7, 2018.
- [9] 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.

- [10] D. Lee and Y. Lee, "Application of neural-network for improving accuracy of roll-force model in hot-rolling mill," *Control Engineering Practice*, vol. 10, no. 4, pp. 473–478, 2002.
- [11] Y. Mahmoodkhani, M. A. Wells, and G. Song, "Prediction of roll force in skin pass rolling using numerical and artificial neural network methods," *Ironmaking & Steelmaking*, vol. 44, no. 4, pp. 281–286, 2017.
- [12] Y. Y. Yang, D. A. Linkens, J. Talamantes-Silva, and I. C. Howard, "Roll force and torque prediction using neural network and finite element modelling," *ISIJ International*, vol. 43, no. 12, pp. 1957–1966, 2003.
- [13] S. John, S. Sikdar, P. K. Swamy, S. Das, and B. Maity, "Hybrid neural-GA model to predict and minimise flatness value of hot rolled strips," *Journal of Materials Processing Technology*, vol. 195, no. 1–3, pp. 314–320, 2008.
- [14] Z. Wang, G. Ma, D. Gong, J. Sun, and D. Zhang, "Application of mind evolutionary algorithm and artificial neural networks for prediction of profile and flatness in hot strip rolling process," *Neural Processing Letters*, vol. 50, no. 3, pp. 2455–2479, 2019.
- [15] D. H. Kim, Y. Lee, and B. M. Kim, "Application of ANN for the dimensional accuracy of workpiece in hot rod rolling process," *Journal of Materials Processing Technology*, vol. 130–131, pp. 214–218, 2002.
- [16] P. Wang, Z.-y. Huang, M.-y. Zhang, and X.-w. Zhao, "Mechanical property prediction of strip model based on PSO-BP neural network," *Journal of Iron and Steel Research International*, vol. 15, no. 3, pp. 87–91, 2008.
- [17] M. S. Ozerdem and S. Kolukisa, "Artificial neural network approach to predict mechanical properties of hot rolled, nonresulfurized, AISI 10xx series carbon steel bars," *Journal of Materials Processing Technology*, vol. 199, no. 1–3, pp. 437–439, 2008.
- [18] H. J. Kim, M. Mahfouf, and Y. Y. Yang, "Modelling of hot strip rolling process using a hybrid neural network approach," *Journal of Materials Processing Technology*, vol. 201, no. 1–3, pp. 101–105, 2008.
- [19] 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.
- [20] X. Sui and Z. Lv, "Prediction of the mechanical properties of hot rolling products by using attribute reduction ELM," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 5–8, pp. 1395–1403, 2016.
- [21] J. Liu, X. Liu, and B. T. Le, "Rolling force prediction of hot rolling based on GA-MELM," *Complexity*, vol. 2019, Article ID 3476521, 11 pages, 2019.
- [22] H. Zhang, S. Zhang, Y. Yin, and X. Chen, "Prediction of the hot metal silicon content in blast furnace based on extreme learning machine," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 10, pp. 1697–1706, 2018.
- [23] S. Zhang, T. Zhang, Y. Yin, and W. Xiao, "Alumina concentration detection based on the kernel extreme learning machine," *Sensors*, vol. 17, no. 9, p. 2002, 2017.
- [24] Y. Li, S. Zhang, Y. Yin, W. Xiao, and J. Zhang, "A novel online sequential extreme learning machine for gas utilization ratio prediction in blast furnaces," *Sensors*, vol. 17, no. 8, p. 1847, 2017.
- [25] Y. Zhou and W. Xue, "A multisensor fusion method for tool condition monitoring in milling," *Sensors*, vol. 18, no. 11, p. 3866, 2018.
- [26] S. Tamura and M. Tateishi, "Capabilities of a four-layered feedforward neural network: four layers versus three," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 251–255, 1997.
- [27] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, pp. 513–529, 2012.
- [28] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, pp. 65–85, 1994.
- [29] J. Santos, A. Ferreira, and G. Flintsch, "An adaptive hybrid genetic algorithm for pavement management," *International Journal of Pavement Engineering*, vol. 20, no. 3, pp. 266–286, 2019.
- [30] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [31] B. Y. Qu, B. F. Lang, J. J. Liang, A. K. Qin, and O. D. Crisalle, "Two-hidden-layer extreme learning machine for regression and classification," *Neurocomputing*, vol. 175, pp. 826–834, 2016.
- [32] Y. Miche, M. Van Heeswijk, P. Bas, O. Simula, and A. Lendasse, "TROP-ELM: a double-regularized ELM using LARS and Tikhonov regularization," *Neurocomputing*, vol. 74, no. 16, pp. 2413–2421, 2011.
- [33] R. Leaman, R. Islamaj Dogan, and Z. Lu, "DNorm: disease name normalization with pairwise learning to rank," *Bioinformatics*, vol. 29, no. 22, pp. 2909–2917, 2013.