IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 2, 2021, accepted March 14, 2021, date of publication March 19, 2021, date of current version April 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3067363

Flatness Prediction of Cold Rolled Strip Based on EM-TELM

JINGYI LIU[1], LUSHAN WAN[2], AND DONG XIAO[2], (Member, IEEE)
[1]College of Sciences, Northeastern University, Shenyang 110819, China
[2]College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

Corresponding author: Dong Xiao (xiaodong@ise.neu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0304100, in part by the Fundamental Research Funds for Central University under Grant N2005011, and in part by the Scientific Research Funds of Educational Department of Liaoning Province under Grant LT2020008.

ABSTRACT Flatness of cold rolled strip is an extremely important indicator of quality, and flatness control is the key technology of the modern high-accuracy rolling mill. The establishment of an efficient and accurate flatness prediction model is conducive to improving the flatness accuracy and realizing the effective control of flatness. Inspired by the error minimization principle, error minimized extreme learning machine with two hidden layers (EM-TELM) used to automatically determine the optimum hidden nodes is proposed in the paper, which is applied to establish the flatness prediction model of cold rolled strip. EM-TELM uses the block matrices to solve the output matrix of the second hidden layer. EM-TELM randomly adds one or a group of hidden nodes to the current network every time. During the increasing process of the network structure, the weights matrix connecting the hidden layer and the output layer are updated incrementally. Since EM-TELM is a no analytic method, it can be used in a kind of prediction problem for complex and difficult modeling systems. The experimental results indicated that the accuracy of EM-TELM is higher than that of EM-ELM, and EM-TELM reduces the computational complexity and training time compared to TELM which recalculates the parameters between different hidden layers when the network structure changes.

INDEX TERMS Block matrices, cold rolled strip, error minimization, extreme learning machine, flatness prediction, two-hidden-layer.

I. INTRODUCTION

As the most important steel product in the world, Plate and strip are applied to the most extensive rolling products in the national economic departments and are used in all aspects of the national economy. Such as food packaging, household appliances, precision instruments, automobile manufacturing, aviation, shipbuilding, civil construction, and other industries. It plays an important role in the modernization of national defense and the construction of the national economy. Its production level is an important indicator to measure the development level of a country's steel industry. With the rapid development of the economies of the world, the demand for the strip is increasing. At the same time, the rapid development of the iron and steel industry has led to more and more fierce competition in the strip market. Driven by the market, the requirements of customers on the quality, type, and performance of strip materials have been gradually improved. To meet the requirements of users, improve the competitiveness of enterprises, transform and improve the equipment and technology of each strip rolling production line, and increase the investment in new technology and new process has become an important task. Therefore, the strip rolling technology can move toward a rapid development path of high precision, high speed, and automation [1]–[3].

Flatness [4], [5] refers to the degree of buckling of the plate belt, including the dimension indexes of longitudinal and transverse dimensions of the plate belt. Transverse aspect refers to the section flatness (thickness distribution in the width direction of the plate), that is, the convexity of the plate. In the longitudinal direction, it refers to the flatness of the length of the strip, that is, the straightness, commonly known as the wave shape. Flatness control is the core part of the strip cold rolling production process. In the process of strip rolling deformation, the setting and calculation of flatness are closely related to rolling force and roll bending force. By studying the

The associate editor coordinating the review of this manuscript and approving it for publication was Giambattista Gruosso.

factors influencing the shape of the exit plate and using the existing data to establish a model, the shape of the exit plate can be effectively controlled. With the progressing of computer application technology, the strip production process has been equipped with a complete sensor measurement device, which can obtain a large amount of process data online, such as bending force, rolling force, tension, and other measured values. These process data contain useful information about the running state of the production process, which can be used to predict the quality of the final flatness. However, due to the lack of effective data processing and information extraction methods, the traditional flatness prediction methods do not effectively use a large amount of readily available measurement data. In recent years, big data and machine learning technologies have emerged, and in many fields such as agriculture, medicine, science, and industry [6], there are many cases in which neural networks are modeled by a large amount of data, and both have achieved high accuracy. The neural network can also be introduced into the steel industry, and the existing data can be used for modeling to achieve the prediction effect, which is conducive to decision-making.

Therefore, the paper combines a large number of flatness data in the cold rolling process with neural network technology, and explores the influence of the variation of the bending force of the work rolls and the intermediate rolls on the final exit flatness during the cold rolling process, and establishes a prediction model to effectively predict the flatness. The neural network model established by a large amount of data can hide the whole process in the hidden unit in the model. The established neural network model can learn autonomously through data, learn many hidden and complex knowledge and patterns. At the same time, the value of the large amount of data accumulated on the production line is utilized, and data is used to drive production. Do a good job of forecasting before the start of the production process, and adjust the value of each control means in advance according to the target flatness. That is, by controlling the input and changing the input, the output flatness can be close to or reach the target flatness, which also reduces the adjustment in the production process and saves costs. In recent years, many scholars have established a flatness prediction model based on intelligent methods. However, in the actual test process, it is found that the traditional back propagation (BP) network flatness prediction model has a long training process and is easy to fall into the local minimum problem. The radial basis function (RBF) flatness prediction model often fails to work when the data is insufficient.

Extreme learning machine (ELM) [7] is proposed by Huang *et al.* for a single hidden layer feed forward neural network to overcome the disadvantages of gradient-based algorithms. ELM randomly generates the connection weights matrix between the input layer and the hidden layer and the bias vector of the hidden layer, and no adjustment is needed in the training process [8], [9]. ELM has been favored by many scholars because of its fast learning speed, good generalization performance, and other advantages [10]–[12], and it has been applied to the rolling field in recent years. Wang *et al.* [13] applied it to the rolling force prediction of hot rolled sheets, and the experimental results show that ELM has a significant improvement on the modeling accuracy and the generalization ability of the model compared with the traditional modeling methods such as BP and RBF. Li *et al.* [14] applied ELM to flatness prediction, and the results show that ELM has higher prediction accuracy regardless of sample size and also solves the problem that traditional artificial neural network is easy to fall into a local optimal solution. Although ELM has shown its superior performance in many aspects, how to determine the number of hidden nodes and further improve the prediction accuracy of the model is still an urgent problem to be solved. Feng *et al.* [15] proposed error minimized extreme learning machine (EM-ELM) to dynamically determine the number of hidden nodes, and update the output weights incrementally. A large number of simulation results show that the algorithm can reduce the computational complexity of ELM. In order to further improve the prediction accuracy of the model, Qu *et al.* [16] proposed a two-hidden-layer extreme learning machine (TELM) and Xiao *et al.* [17] proposed a multiple hidden layers extreme learning machine (MELM). The experimental results show that the average accuracy and generalization performance of TELM and MELM are greatly improved compared with ELM.

Based on the above problems, the paper will establish a cold rolling flatness prediction model based on error minimized extreme learning machine with two hidden layers (EM-TELM). EM-TELM uses the block matrices to solve the output matrix of the second hidden layer, which is different from the way that TELM uses the generalized inverse to solve the output matrix of the second hidden layer. Then, EM-TELM adds a hidden layer than EM-ELM, which improves calculation accuracy. EM-TELM adds hidden layer nodes one by one or group by group while keeping the structural parameters of the original hidden layer nodes unchanged, and updates the connection weights between the first hidden layer and the second hidden layer and the bias vector of the second hidden layer incrementally. Finally, EM-TELM is validated by the strip steel production data of the cold rolling mill. The experimental results show that EM-TELM has a higher accuracy of flatness prediction than EM-ELM, and EM-TELM reduces computational complexity and reduces training time compared with TELM.

II. BRIEF REVIEW OF ELM AND TELM

A. ELM

Extreme learning machine (ELM) randomly generates the weights between the input layer and the hidden layer and the bias of the hidden nodes. And ELM only needs to set the number of nodes in the hidden layer during the training process to obtain the unique optimal solution. The advantage of ELM is that it improves the generalization performance of the network and avoids time-consuming iterative training steps and Local minimum.

Suppose there are N independent samples $(x_i, t_i)(i = 1, 2, \ldots, N)$ consisting of the input $X = [x_1, x_2, \ldots, x_N]^T$ and the expected output $T = [t_1, t_2, \ldots, t_N]^T$, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$, $t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in R^m$. And transpose of vector or matrix represented by superscript T in the paper. Assuming that the number of nodes in the hidden layer is L and the activation function of the hidden layer is $g(x)$. And ELM randomly selects the weight matrix $W = [W_1, W_2, \ldots, W_L]^T \in R^{L\times n}$ connecting the input layer and the hidden layer and the bias vector $B = [b_1, b_2, \ldots, b_L]^T \in R^{L\times N}$ of the hidden nodes. After determining W and B, their values will not be changed in the training stage. The next steps make the nonlinear system to be transformed into the linear system whose mathematical description is

$$H\beta = T \tag{1}$$

where $\beta = [\beta_1, \beta_2, \ldots, \beta_L]^T \in R^{L\times m}$ is the weights matrix connecting the hidden layer and the output layer and its vector element $\beta_j = [\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm}]^T (j = 1, 2, \ldots, L)$ represents the connection weights between the jth hidden node and the mth output nodes, and $H = g(WX + B) \in R^{N\times L}$ is the output matrix of the hidden layer and its expression is

$$H(w_1, \ldots, w_L, b_1, \ldots, b_L, x_1, \ldots, x_N) = \begin{bmatrix} g(w_1, b_1, x_1) & \cdots & g(w_L, b_L, x_1) \\ \vdots & \cdots & \vdots \\ g(w_1, b_1, x_N) & \cdots & g(w_L, b_L, x_N) \end{bmatrix}_{N\times L} \tag{2}$$

where $h_{ij} = g(W_j x_i + b_j)(i = 1, 2, \cdots, N, j = 1, 2, \cdots, L)$ represents the output of the jth node corresponding to x_i, $W_j = [W_{j1}, W_{j2}, \cdots, W_{jn}]^T$ represents the connection weight between nth input node and jth hidden node, b_j is the bias of jth hidden node, and $W_j x_i$ represents the inner product between W_j and x_i.

Then ELM uses the least square method to obtain the output matrix β.

$$\beta = H^+ T \tag{3}$$

where H^+ is the Moore-Penrose generalized inverse [18] of the matrix H, which can be calculated by the orthogonal projection method. In other words $H^+ = (H^T H)^{-1} H^T$ if $H^T H$ is nonsingular, and $H^+ = H^T (HH^T)^{-1}$ if HH^T is nonsingular.

B. TELM

Tamura and Tateishi [19] pointed out that the advantage of the two-hidden-layer feedforward networks (TLFNs) is that fewer hidden nodes can be used to achieve the desired performance, and it can achieve arbitrarily small errors by using TLFNs with $(N/2 + 3)$ hidden layer nodes to learn N samples. Huang [20] further proved that the number of nodes in the hidden layer can be $2\sqrt{(m + 3)N}$. Therefore, Qu *et al.* proposed TELM [16] to bring superiority of double-layer structure into conventional ELM algorithm. The work flow chart and the network structure of TELM are shown in Figures 1 to 2 respectively.

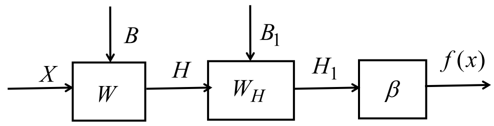


FIGURE 1. The work flow chart of TELM. The parameters of the first hidden layer are B and W and the output is H, and the parameters of the second hidden layer are B_1 and W_H and the output is H_1. The actual output of TELM is $f(x)$.

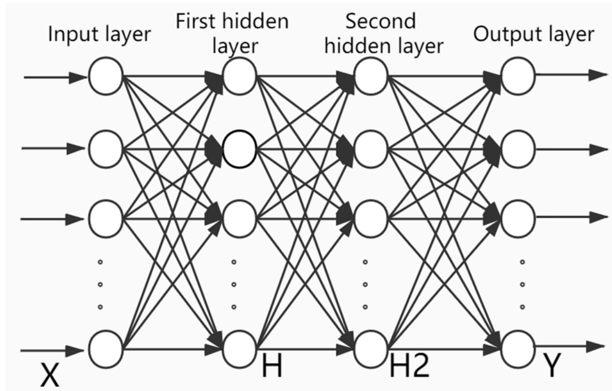


FIGURE 2. The network structure of TELM. The actual output of TELM is $Y = f(x)$ and the actual output of the second hidden layer is $H_2 = H2$. TELM contains two hidden layers with the same activation function ($g(x)$) and the same number of nodes (L).

Suppose there are N samples (x_i, t_i). First, the weights matrix connecting the input layer and the first hidden layer and the bias vector of the first hidden layer are initialized by using the randomly generated values. Then the two hidden layers are regarded as one hidden layer, and the connection matrix β between the second hidden layer and the output layer is calculated according to formula (3). In this way, we can obtain the expected output of the second hidden layer as follows:

$$H_1^* = T\beta^+ \tag{4}$$

where β^+ is the Moore-Penrose generalized inverse of the matrix β, the definition and its calculation method of β^+ are the same as that of H^+. In other words $\beta^+ = (\beta^T\beta)^{-1}\beta^T$ if $\beta^T\beta$ is nonsingular, and $\beta^+ = \beta^T(\beta\beta^T)^{-1}$ if $\beta\beta^T$ is nonsingular.

The theoretical output of the second hidden layer is

$$H_1 = g(W_H H + B_1) \tag{5}$$

where W_H is the connection matrix during the first and second hidden layers, B_1 is the bias vector of the second hidden layer, and H is the output of the first hidden layer.

Let $W_{HE} = [\, B_1 \;\; W_H \,]$,

$$W_{HE} = g^{-1}(H_1^*) H_E^+ \tag{6}$$

where H_E^+ is the Moore-Penrose generalized inverse of $H_E = [\, 1 \;\; H \,]^T$, and $g^{-1}(x)$ is the inverse function of $g(x)$. Therefore, the paper also get the actual output of the second hidden layer.

$$H_2 = g(W_{HE} H_E) \tag{7}$$

Finally, the output weight of the network is updated and

$$\beta_{new} = H_2^{+} T \tag{8}$$

where H_2^{+} is the Moore-Penrose generalized inverse of H_2. Then the output of TELM is

$$f(x) = H_2 \beta_{new} \tag{9}$$

III. PROPOSED LEARNING ALGORITHM

The number of nodes in the hidden layer of ELM has always been an important research issue. It has been proved in [15] and [21] that the prediction error of ELM will be smaller and smaller as the number of hidden nodes increases. And EM-ELM [15] dynamically determines the network structure based on the principle of error minimization and allows the nodes of the hidden layer to be added to the network one by one or a group of groups. EM-ELM solves the generalized inverse in the calculation process by block matrices. Therefore, the paper proposes error minimized extreme learning machine with two hidden layers (EM-TELM) after integrating the advantages of EM-ELM and TELM. And EM-TELM uses the block matrices method to determine the parameter of the second hidden layer.

A. CONVERGENCE ANALYSIS OF TELM

Before introducing EM-TELM formally, let's briefly introduce a lemma to prove the convergence of TELM. The derivation process of the convergence of TELM is similar to that of the convergence of ELM.

Lemma 1(Convergence Lemma): A TELM network is given. Let $H_{1,1} = H(a_1, \cdots, a_{L_0}, b_1, \cdots, b_{L_0}, x_1, \cdots, x_N)$ and $H_{1,2}$ denote the output matrix of the first hidden layer and the output matrix of the second hidden layer, respectively. Each hidden layer of TELM contains L_0 nodes $\{(a_i, b_i)\}_{i=1}^{L_0}$. If each hidden layer of TELM adds $L_1 - L_0$ new nodes $\{(a_i, b_i)\}_{i=L_0+1}^{L_0}$, the new output matrix of the first hidden layer and the second hidden layer becomes $H_{2,1} = H(a_1, \cdots, a_{L_1}, b_1, \cdots, b_{L_1}, x_1, \cdots, x_N)$ and $H_{2,2}$, separately. Then

$$\begin{aligned} E(H_{2,2}) &= \min \left\| H_{2,2}\beta_{2,2} - T \right\| \le E(H_{1,2}) \\ &= \min \left\| H_{1,2}\beta_{1,2} - T \right\|. \end{aligned}$$

Proof: Since $H_{1,1}$ and $H_{2,1}$ are the output of the hidden layer before and after adding $\delta L_0 = L_1 - L_0$ new nodes respectively and $H_{E1} = [\,1 \;\; H_{1,1}\,]^T$,

$$H_{E2} = [\,1 \;\; H_{2,1}\,]^T = [\,1 \;\; H_{1,1} \;\; \delta H_{1,1}\,]^T = [\,H_{E1}^T \;\; \delta H_{1,1}\,]^T \tag{10}$$

where the output corresponding to the new node in the first hidden layer is

$$\delta H_{1,1} = \begin{bmatrix} g(a_{L_0+1}, b_{L_0+1}, x_1) & \cdots & g(a_{L_1}, b_{L_1}, x_1) \\ \vdots & \cdots & \vdots \\ g(a_{L_0+1}, b_{L_0+1}, x_N) & \cdots & g(a_{L_1}, b_{L_1}, x_N) \end{bmatrix} \tag{11}$$

Since $\beta_{2,2}$ is the least square solution of $\min \left\| H_{2,2}\beta - T \right\|$, according to formula (7),

$$\begin{aligned} E(H_{1,2}) &= \min \left\| H_{1,2}\beta_{1,2} - T \right\| \\ &= \min \left\| g(W_{HE1}H_{E1})\beta_{1,2} - T \right\| \\ &= \min \left\| g(g^{-1}(T\beta_{1,1}^{+})H_{E1}^{+}H_{E1})\beta_{1,2} - T \right\| \\ &= \min \left\| g(g^{-1}(T\beta_{1,1}^{+})(H_{E1}^{T}H_{E1})^{-1}H_{E1}^{T}H_{E1}) \right. \\ &\qquad \left. \times \beta_{1,2} - T \right\| \\ &= \min \left\| T(\beta_{1,1}^{T}\beta_{1,1})^{-1}\beta_{1,1}^{T}\beta_{1,2} - T \right\| \end{aligned} \tag{12}$$

$$\begin{aligned} E(H_{2,2}) &= \min \left\| H_{2,2}\beta_{2,2} - T \right\| \\ &= \min \left\| g(W_{HE2}H1_{E2})\beta_{2,2} - T \right\| \\ &= \min \left\| g(g^{-1}(T\beta_{1,2}^{+})H_{E2}^{+}H_{E2})\beta_{2,2} - T \right\| \\ &= \min \left\| g(g^{-1}(T\beta_{1,2}^{+})([\,H_{E1}^{T} \;\; \delta H_{1,1}\,] \begin{bmatrix} H_{E1} \\ \delta H_{1,1}^{T} \end{bmatrix})^{-1} \right. \\ &\qquad \left. \times [\,H_{E1}^{T} \;\; \delta H_{1,1}\,] \begin{bmatrix} H_{E1} \\ \delta H_{1,1}^{T} \end{bmatrix})\beta_{2,2} - T \right\| \\ &\le \min \left\| g(g^{-1}(T([\,\beta_{1,1}^{T} \;\; 0^{T}\,] \begin{bmatrix} \beta_{1,1} \\ 0 \end{bmatrix})^{-1}[\,\beta_{1,1}^{T} \;\; 0^{T}\,])) \right. \\ &\qquad \left. \times \begin{bmatrix} \beta_{1,2} \\ 0 \end{bmatrix} - T \right\| \\ &= \min \left\| T\left[\,(\beta_{1,1}^{T}\beta_{1,1})^{-1} \;\; \beta_{1,1}^{T} \;\; 0\,\right] \begin{bmatrix} \beta_{1,2} \\ 0 \end{bmatrix} - T \right\| \\ &= \min \left\| T(\beta_{1,1}^{T}\beta_{1,1})^{-1}\beta_{1,1}^{T}\beta_{1,2} - T \right\| \\ &= \min \left\| H_{1,2}\beta_{1,2} - T \right\| \\ &= E(H_{1,2}) \end{aligned} \tag{13}$$

B. EM-ELM

The paper first introduces EM-ELM [15] and then extends to EM-TELM. First, the paper gives the initial number L_0 and the maximum number $L_{\max}$ of hidden nodes in each hidden layer, the expected prediction error is ε, and $H_1 = H(w_1, \cdots, w_{L_0}, b_1, \cdots, b_{L_0}, x_1, \cdots, x_N)$ is the output of the hidden layer with L_0 hidden nodes. Suppose the paper adds $\delta L_0 = L_1 - L_0$ hidden nodes $\{(w_i, b_i)\}_{i=L_0+1}^{L_1}$ to the network, a new output is $H_2 = H(w_1, \cdots, w_{L_1}, b_1, \cdots, b_{L_1}, x_1, \cdots, x_N)$. According to formula (2), $H_2 = [\,H_1 \;\; \delta H_1\,]$, where the output corresponding to the new node in the hidden layer is

$$\delta H_1 = \begin{bmatrix} g(w_{L_0+1}, b_{L_0+1}, x_1) & \cdots & g(w_{L_1}, b_{L_1}, x_1) \\ \vdots & \cdots & \vdots \\ g(w_{L_0+1}, b_{L_0+1}, x_N) & \cdots & g(w_{L_1}, b_{L_1}, x_N) \end{bmatrix}_{N \times (L_1 - L_0)} \tag{14}$$

Let $E(H) = \min \|H\beta - T\|$, and $E(H)$ is the prediction error of the network. If $E(H_1) = \min \|H_1\beta_1 - T\| < \varepsilon$, it is no need to add hidden nodes to the network, and the training step is completed. According to the introduction of the extreme learning machine,

$$H_2^+ = (H_2^T H_2)^{-1} H_2^T = (\begin{bmatrix} H_1^T \\ \delta H_1^T \end{bmatrix} [H_1 \ \delta H_1])^{-1} \begin{bmatrix} H_1^T \\ \delta H_1^T \end{bmatrix} \quad (15)$$

Let

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = (\begin{bmatrix} H_1^T \\ \delta H_1^T \end{bmatrix} [H_1 \quad \delta H_1])^{-1} \quad (16)$$

Then

$$H_2^+ = \begin{bmatrix} U \\ D \end{bmatrix} = \begin{bmatrix} A_{11}H_1^T + A_{12}\delta H_1^T \\ A_{21}H_1^T + A_{22}\delta H_1^T \end{bmatrix} \quad (17)$$

As described in [5], H_1 is a full rank matrix when $N \geq L_1$. Then the Schur complement M of $H_1^T H_1$ is necessarily invertible. And $M = \delta H_1^T \delta H_1 - \delta H_1^T H_1 (H_1^T H_1)^{-1} H_1^T \delta H_1$. The matrix A is invertible when $H_1^T H_1$ is nonsingular matrix. According to the method for computing the inverses of 2×2 block matrices,

$$\begin{cases} A_{11} = (H_1^T H_1)^{-1} + (H_1^T H_1)^{-1} H_1^T \delta H_1 R^{-1} \\ \delta H_1^T H_1 (H_1^T H_1)^{-1} \\ A_{12} = -(H_1^T H_1)^{-1} H_1^T \delta H_1 R^{-1} \\ A_{21} = -R^{-1} \delta H_1 H_1 (H_1^T H_1)^{-1} \\ A_{22} = R^{-1} \end{cases} \quad (18)$$

where $R = \delta H_1^T \delta H_1 - \delta H_1^T H_1 (H_1^T H_1)^{-1} H_1^T \delta H_1^T$.

Since $H_1^+ = (H_1^T H_1)^{-1} H_1^T$, $R = \delta H_1^T \delta H_1 - \delta H_1^T H_1 H_1^+ \delta H_1^T$. Then the paper calculates D from formulas (17) and (18).

$$\begin{aligned} D &= R^{-1} \delta H_1^T - R^{-1} \delta H_1^T H_1 H_1^+ \\ &= (\delta H_1^T \delta H_1 - \delta H_1^T H_1 H_1^+ \delta H_1)^{-1} \delta H_1^T (I - H_1 H_1^+) \\ &= (\delta H_1^T (I - H_1 H_1^+) \delta H_1)^{-1} \delta H_1^T (I - H_1 H_1^+) \end{aligned} \quad (19)$$

Because $I - H_1 H_1^+$ has the characteristics of symmetry and orthogonal projection,

$$D = ((I - H_1 H_1^+) \delta H_1)^+ \quad (20)$$

In the same way,

$$U = H_1^+ - H_1^+ \delta H_1^T D \quad (21)$$

So far, the H_2 of EM-ELM can be obtained. Then EM-ELM solves the parameter β in the same way as ELM.

C. EM-TELM

EM-ELM can be transformed into EM-TELM by adding a new hidden layer. And EM-TELM uses the block matrices to solve the output matrix of the second hidden layer, which is similar to the process of EM-ELM solving the output matrix of the hidden layer. Unlike TELM using generalized inverse to solve the output matrix of the hidden layer, EM-TELM can solve this problem by using the block matrices. In addition, EM-TELM gradually increases the nodes of the hidden layer, and the other settings are the same as TELM. Specifically, EM-TELM uses the error minimized theory and successively increases the number of nodes in the hidden layer, which can make the error between the actual output and the expected output of the model smaller and smaller. And EM-TELM uses the block matrices to replace the generalized inverse but does not include the generalized inverse of β, which reduces the computational complexity and improves the operating efficiency of the model. It is worth noting that the parameters between the original nodes are unchanged during the process of adding nodes, which reduces the computational complexity. All in all, the biggest feature of EM-TELM is that it uses the block matrices to solve the output matrix of the second hidden layer, and the network updates faster and the computational complexity is low after adding hidden layer nodes.

The parameter settings of EM-TELM and EM-ELM in Part B are the same. And the derivation process of EM-TELM is the same as TELM, but the difference lies in the way of solving the output matrix of the hidden layer. When EM-TELM has a single hidden layer, the derivation process of EM-TELM is the same as EM-ELM. After finding the output H_2 of the first hidden layer, then

$$\beta_{1,1} = H_2^+ T \quad (22)$$

Then, the expected output of the second hidden layer is $H_{1,2}^* = T\beta_{1,1}^+$. According to formula (5), the actual output of the second hidden layer is $H_{1,2} = g(W_H H_2 + B_1)$. According to formula (6), the parameters of the second hidden layer are $W_{HE2} = g^{-1}(H_{1,2}^*) H_{E1}^+$, where

$$H_{E1} = [\,1 \ \ H_2\,]^T = [\,1 \ \ H_1 \ \ \delta H_1\,]^T = [\,H_{E0}^T \ \ \delta H_1\,]^T = R^T \quad (23)$$

According to the introduction of the extreme learning machine,

$$R^+ = (R^T R)^{-1} R^T \quad (24)$$

Let

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = (\begin{bmatrix} H_{E0} \\ \delta H_1^T \end{bmatrix} [H_{E0}^T \ \ \delta H_1])^{-1} \quad (25)$$

Then the paper uses the block matrices method to obtain

$$R^+ = \begin{bmatrix} U_1 \\ D_1 \end{bmatrix} = \begin{bmatrix} (H_{E0}^T)^+ (I - \delta H_1 D_1) \\ ((I - H_{E0}^T (H_{E0}^T)^+) \delta H_1)^+ \end{bmatrix} \quad (26)$$

Then

$$H_{E1}^T = [\,U_1^T \quad D_1^T\,] \quad (27)$$

Therefore, W_{HE2} and $H_{1,2}$ can be obtained, and β_{new} can be obtained by formulas (8) and (9).

The specific steps of the proposed EM-TELM algorithm are as follows.

1) The parameters of the first hidden layer in EM-TELM with $2L_0$ hidden nodes are randomly initialized and each hidden layer has L_0 hidden nodes. And L_0 is a small positive integer given by a person. At the same time, let $k = 0$.

2) The two hidden layers are regarded as one hidden layer, and the connection weight matrix β between the second hidden layer and the output layer is obtained from formula (3).

3) The connection weight and bias between the second hidden layer and the first hidden layer can be obtained from formulas (4) to (6).

4) The output matrix H_2 of the second hidden layer is calculated by formula (7), and the prediction error $E(H_2) = \left\| H_2 H_2^+ T - T \right\|$ of the model is calculated.

5) $k = k + 1$.

6) EM-TELM randomly initialize δL_{k-1} hidden nodes newly added to the first hidden layer and both hidden layers are added with δL_{k-1} hidden nodes. Thus the number of nodes for each hidden layer in the existing network is $L_k = L_{k-1} - \delta L_{k-1}$. Meanwhile, the output matrix of the first hidden layer in the $(k+1)$th iteration is $H_{k+1} = [\, H_k \;\; \delta H_k \,]$, and the increment matrix δH_k can be concretely expressed as

$$\delta H_k = \begin{bmatrix} g(w_{L_{k-1}}, b_{L_{k-1}}, x_1) & \cdots & g(w_{L_k}, b_{L_k}, x_1) \\ \vdots & \cdots & \vdots \\ g(w_{L_{k-1}}, b_{L_{k-1}}, x_N) & \cdots & g(w_{L_k}, b_{L_k}, x_N) \end{bmatrix}_{N \times \delta L_{k-1}} \quad (28)$$

7) The two hidden layers are regarded as one hidden layer. According to formula (3), the output weight matrix between the second hidden layer and the output layer are updated in a fast recursive way as

$$D_k = ((I - H_k H_k^+)\delta H_k)^+ \quad (29a)$$
$$U_k = H_K^+(I - \delta H_k^T D_k) \quad (29b)$$
$$\beta_{k+1} = H_{k+1}^+ T = \begin{bmatrix} U_k \\ D_k \end{bmatrix} T \quad (29c)$$

8) According to formula (6), we can get the connection weight matrix between the second hidden layer and the first hidden layer and the bias vector of the second hidden layer recursively, and

$$H_{E,k+1}^T = [\, 1 \;\; H_{k+1} \,] = [\, 1 \;\; H_k \;\; \delta H_k \,] \quad (30)$$

Let $A = [\, 1 \;\; H_k \,]$, formula (30) can be simplified to

$$H_{E,k+1}^T = [\, A \;\; \delta H_k \,] \quad (31)$$

So EM-TELM get $H_{E,k+1}^+$ in a recursive way

$$D_{k,1} = ((I - A_k A_k^+)\delta H_k)^+ \quad (32a)$$
$$U_{k,1} = A_k^+(I - \delta H_k^T D_k) \quad (32b)$$
$$(H_{E,k+1}^T)^+ = \begin{bmatrix} U_{k,1} \\ D_{k,1} \end{bmatrix} \quad (32c)$$
$$H_{E,k+1}^+ = [\, U_{k,1}^T \quad D_{k,1}^T \,] \quad (32d)$$

Then the connection matrix $W_{HE,k}$ containing the weights matrix and bias vector between the first and second hidden layers can be solved as

$$W_{HE,k} = g^{-1}(H_{1,k}^*) H_{E,k+1}^+ \quad (33)$$

where $H_{1,k}^* = H_{k+1}^+ T$.

9) The output matrix $H_{2,k+1}$ of the second hidden layer is calculated by formula (7), and then we can get the prediction error $E(H_{2,k+1}) = \left\| H_{2,k+1} H_{2,k+1}^+ T - T \right\|$ of the model.

10) If $L_k < L_{\max}$ and $E(H_{2,k+1}) > \varepsilon$, return to step 5. Otherwise, the process of training is completed.

Among them, steps 1 to 4 belong to the initialization phase, and steps 5 to 11 belong to the recursively growing phase. As a special case, the hidden node is added to the existing EM-TELM one by one, which is $\delta L_0 = \delta L_1 = \delta L_2 = \cdots = \delta L_k = 1$. And δH_k is a vector and denotes as

$$\delta h_k = [G(w_{L_{k-1}}, b_{L_{k-1}}, x_1), \cdots, G(w_{L_{k-1}}, b_{L_{k-1}}, x_N)]^T \quad (34)$$

and the calculation formulas of U_k and D_k become

$$D_k = \frac{\delta h_k^T (I - H_k H_k^+)}{\delta h_k^T (I - H_k H_k^+)\delta h_k} \quad (35a)$$
$$U_k = H_k^+(I - \delta h_k D_k) \quad (35b)$$

According to the specific steps of EM-TELM, the pseudo-code algorithm of EM-TELM can be obtained and shown below.

Input: The input variable X, and the expected output T.
Output: The structure parameters and the performance indicators of EM-TELM.

Set the structural parameters of EM-TELM as W, B, W_H, B_1 and β respectively, the initial number of nodes in the hidden layer is L_0, the maximum number of nodes in the hidden layer is $L_{\max}$, the expected error of EM-TELM is ε, and $k = 0$.
The initialization phase:
    When the number of nodes in the hidden layer of EM-TELM is L_0, formulas (3) to (7) are used to solve the initial structural parameters of EM-TELM.
The recursively growing phase:
    When $L_k < L_{\max}$ and $E(H_{2,k+1}) > \varepsilon$
        $k = k + 1$;
        EM-TELM adds δL_{k-1} new hidden layer node.
        According to formula (29), the output H_{k+1} of the first hidden layer is calculated using the block matrices.
        According to formula (32), the output $H_{E,k+1}$ of the first hidden layer is calculated using the block matrices.
        According to formula (33) and step 9, the predic -tion error $E(H_{2,k+1}) = \left\| H_{2,k+1} H_{2,k+1}^+ T - T \right\|$ of the model is calculated.
    end

D. CONVERGENCE ANALYSIS OF EM-TELM

Knowing that TELM and EM-ELM have convergence [15], the paper can give and prove that EM-TELM has convergence. The derivation process of the convergence of

EM-TELM is similar to that of the convergence of EM-ELM, so the paper will not repeat the proof.

Theorem 1 (Convergence Theorem): For a given set of distinct samples $\wp = \{(x_i, t_i)|x_i \in R^n, t_i \in R^m, i = 1, \cdots, N\}$ and a given arbitrary positive value ε, there exists a positive integer k such that $E(H_{2,k}) = \min \left\| H_{2,k}\beta_{2,k} - t \right\| \leq \varepsilon$.

IV. PERFORMANCE VERIFICATION

A. EXPERIMENT OBJECT: 1740MM PRODUCTION LINE

In this section, we apply the proposed algorithm to the prediction of flatness. The dataset comes from the actual production data collected by the 1740 mm production line of the steel mill, and the exit flatness of the fifth frame of the strip steel is predicted through the proposed algorithm.

The 1740mm production line was completed in 2015. The pickling and rolling mill in the production line uses the combined pickling and rolling mill. Plate rolls are installed behind the first and fifth frames, and the fifth frame controls the shape of the plates by means of roll bending, roll shifting, and segment cooling. The specification of the raw material is (1.5-6.0)*(700-1600)mm, and the specification of the finished product is (0.2-2.5)*(700-1600)mm.

The data used are the actual measurement results of sensors of 5 racks in the first production sequence of the cold rolling mill production line. By consulting relevant literature and data, it was determined that the characteristic variables are 5 framework roll bending force, 5 frames intermediate roll bending force, 5 frames rolling force, 5 frames incoming tension, outgoing tension, first frame front tension, curl tension, and exit flatness measured in each sensor area of the first frame, namely input variables, the exit flatness measured by each sensor area of the fifth rack is taken as the target variable, that is, the output variable.

The flatness refers to the degree of warpage of the strip and is represented by elongation of each fiber [22].

$$\lambda = \frac{\Delta L}{L} \tag{36}$$

where λ represents the elongation of the longitudinal strip in the direction of the length of the strip, ΔL represents the difference between the longitudinal strip and the reference length in the direction of the length of the strip, and L which is the average length of each longitudinal strip represents the reference length of the strip.

Since the elongation calculated according to the formula (36) is a small value, the unit I is used to characterize the flatness in order to characterize the defects of the flatness visually. And the relationship between I and λ is

$$I = 10^5\lambda = 10^5\frac{\Delta L}{L} \tag{37}$$

Therefore, the unit of the industrially measured export flatness data is I. If the data value is greater than 0, the elongation of the flatness of the measuring point is positive, indicating that the elongation of the flatness is too long compared with the reference length and the plate quality is loose. If the data value is less than 0, the elongation of the flatness of the measuring point is negative, indicating that the elongation of the flatness is too short compared with the reference length and the plate quality is tight. If the data value is close to 0, it indicates that the flatness is close to the reference length and the flatness is good.

In the 1740mm production line, 32 sets of sensors are installed at the exit of the first rack to measure the exit flatness which is divided into 32 areas. The 54 sets of sensors are installed on the outlet of the fifth rack to measure the exit flatness which is divided into 54 areas. Through the analysis of the data, it can be seen that the bending force of the roll does not change in a certain period, but the flatness continues to change. Thus time is also an important factor influencing the prediction of flatness. The sample points in the dataset are generated sequentially every 0.08 seconds, so the time column is added to the input variable to reflect the change of the sample point time. Since the data used in the experiment were taken from the period from 8:56 to 9:24 on a certain day, the added time was listed as

$$time = (0.08, 0.16, \cdots, 18762.24)^T \tag{38}$$

In summary, the number of input variables is $27 + 1 + 32 = 60$, and the number of output variables is 54. Due to the flatness data is only close to 0 and not equal to 0, the data with the flatness of 0 is rejected. Therefore, the exit flatness measured from area 1 to area 9 of the first rack and the fifth rack should be excluded, which also applies to area 46 to area 54 of the fifth rack. The final number of output variables is 51, and the final number of output variables is 36.

At the same time, the paper selects 8840 groups as the training set and the remaining 9614 groups as the test set. Since the data unit difference between the rolling force and the frame tension is large, it will affect the prediction accuracy of the model. Therefore, we have standardized processing of the input data with a mean of 0 and a variance of 1 before modeling. And the experimental environment is MATLAB 2016b.

B. ANALYSIS OF ACCURACY

For parameter setting, the EM-TELM algorithm increases the hidden nodes one by one, the activation function of the hidden layer is Sigmoid, the starting node of the network is 5 (that is, each hidden layer has 5 hidden nodes), the maximum number of hidden nodes for each hidden layer is 30, and the expected prediction error of the model is 1.0. The parameter setting of EM-ELM is the same as EM-TELM.

The index for evaluating the accuracy of the model selects the mean absolute deviation, which is the average for the absolute values of the deviations between the individual observation and the arithmetic mean. The mean absolute deviation can avoid the problem of mutual cancellation of errors and can accurately reflect the size of the actual forecast error. Suppose the actual output of the model is $Tsim$, the expected output of the model is $Ttest$, the number of samples

TABLE 1. **The mean absolute deviation of EM-ELM and EM-TELM with 5-30 hidden nodes.**

Hidden nodes	EM-ELM	EM-TELM
5	1.40	1.42
6	1.40	1.40
7	1.39	1.40
8	1.39	1.39
9	1.39	1.39
10	1.38	1.39
11	1.38	1.38
12	1.38	1.38
13	1.38	1.38
14	1.37	1.37
15	1.37	1.37
16	1.37	1.35
17	1.37	1.35
18	1.36	1.35
19	1.36	1.34
20	1.36	1.34
21	1.36	1.34
22	1.36	1.33
23	1.35	1.33
24	1.34	1.33
25	1.34	1.33
26	1.34	1.33
27	1.34	1.33
28	1.34	1.32
29	1.34	1.32
30	1.33	1.32

is *num*, and the mean absolute deviation is *Mean*, then

$$Mean = \frac{1}{num}\sum_{i=1}^{num}|Tsim_i - Ttest_i| \tag{39}$$

The number of hidden layers of EM-TELM is more than that of EM-ELM, but it is developed based on EM-ELM. Through the comparison of the mean absolute deviation (Table 1 and Figure 3), the accuracy of EM-TELM is better than that of EM-ELM as the number of hidden layer nodes increases. Therefore, the increase in the number of hidden layers improves the accuracy of the model. It should be pointed out that the mean absolute deviation used in the paper refers to the mean absolute deviation of the test set.

C. ANALYSIS OF MODEL TRAINING TIME

The parameter settings of TELM in the section are the same as EM-TELM. When the number of nodes in the hidden layer of TELM changes, TELM needs to recalculate the structural parameters of all nodes. When the number of nodes in the hidden layer of EM-TELM increases, EM-TELM only needs to solve the structural parameters of the newly added nodes by using the block matrices, which helps to reduce the training time. As shown in Table 2 and Figure 4, the training time of EM-TELM has been much lower than that of TELM with the

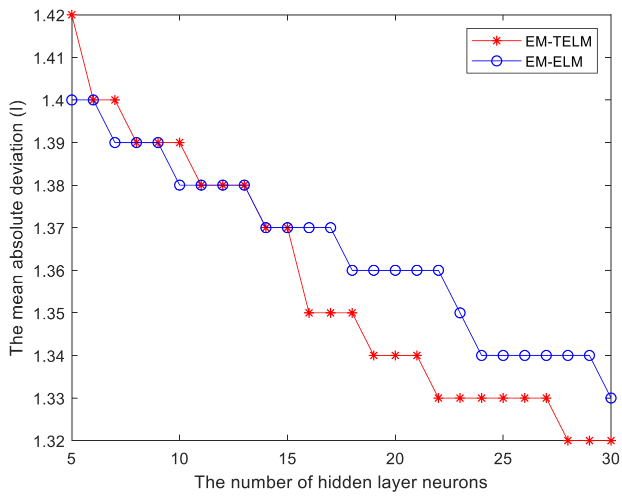


FIGURE 3. **The mean absolute deviation of EM-ELM and EM-TELM with different hidden nodes. When the number of nodes in the hidden layer is less than 16, the mean absolute deviation of EM-TELM is basically the same as EM-ELM. When the number of nodes in the hidden layer is greater than 16, the mean absolute deviation of EM-TELM is smaller than that of EM-ELM.**

TABLE 2. **The comparison of training time (10^{-3}s) of TELM and EM-TELM with 5-30 hidden nodes.**

Hidden nodes	TELM	EM-TELM
5	2.6	1.2
6	3.0	1.4
7	3.4	1.7
8	3.8	1.6
9	4.7	1.8
10	4.9	1.9
11	5.3	2.1
12	5.7	2.3
13	6.4	2.4
14	7.2	2.7
15	7.6	2.8
16	8.1	3.0
17	8.8	3.0
18	9.5	3.3
19	10.1	3.4
20	10.5	3.4
21	11.4	3.8
22	11.5	4.0
23	12.5	4.1
24	12.8	4.2
25	14.4	4.7
26	14.9	4.9
27	14.5	5.0
28	15.0	5.5
29	17.0	5.8
30	16.2	5.6

increase of hidden layer nodes. Therefore, the training time of EM-TELM is less than that of TELM because EM-TELM uses the block matrices to replace the generalized inverse and

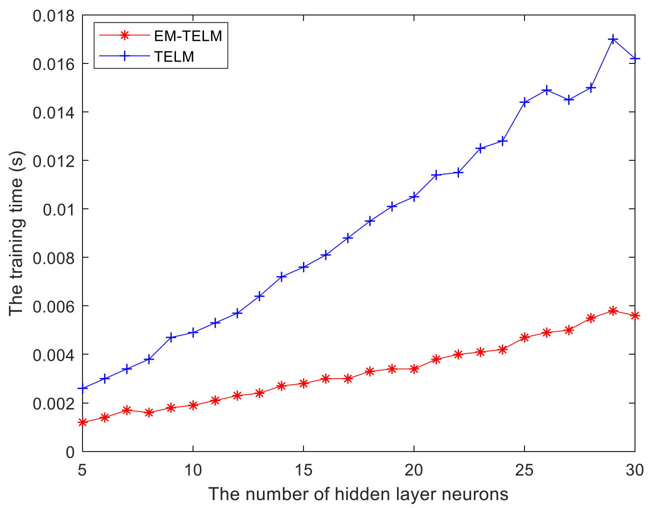


FIGURE 4. The comparison of training time of TELM and EM-TELM with different hidden nodes. The training time of EM-TELM has been much lower than that of TELM with the increase of hidden layer nodes. And the training time is gradually increasing with increase of hidden layer nodes.

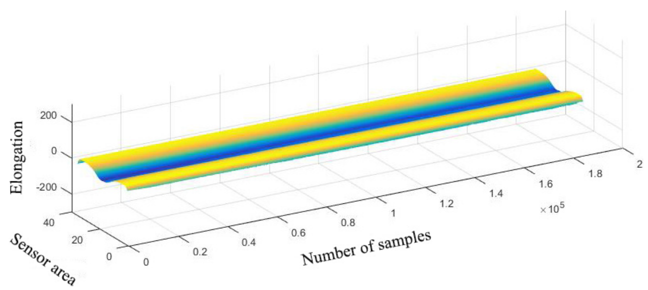


FIGURE 5. Three-dimensional distribution map of the exit flatness based on EM-TELM. The number of samples is 18454 and serves as the X-axis. The number of sensor areas is 36 and is the output variable of the model, as well as the Y-axis. The elongation is export flatness, the unit is *I*, and it is used as the Z-axis.

does not change the structural parameters of existing nodes, which also applies to the relationship between EM-ELM and ELM.

D. THREE-DIMENSIONAL DISTRIBUTION MAP OF THE EXIT FLATNESS BASED ON EM-TELM

From Figure 3, we can see that the accuracy of the flatness prediction model based on EM-TELM is higher than that based on EM-ELM. It can be seen from Figure 4 that as the number of hidden nodes grows, it is much faster to update the connection weight matrix and bias vector between hidden layers by incremental learning than the traditional TELM method. Finally, the established model is used to forecast the data collected in the time period from 8:55:32 to 9:26:57, and a three-dimensional distribution map of the exit flatness based on EM-TELM is obtained.

V. CONCLUSION

The paper proposes EM-TELM based on EM-ELM and TELM, and EM-TELM uses the block matrices to solve the output matrix of the second hidden layer. At the same time, EM-TELM can allow hidden nodes to be added to the network one by one or group by group. It can be seen from the above experimental results that the accuracy of EM-TELM is higher than that of EM-ELM. Compared with TELM which recalculates the parameters between different hidden layers based on the entire new output matrix of the first hidden layer whenever the network architecture is changed, EM-TELM reduces the computation complexity by only updating the parameters between different hidden layers incrementally each time.

REFERENCES

[1] 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 Process. Lett.*, vol. 50, no. 3, pp. 2455–2479, Dec. 2019.

[2] P. F. Wang, Z. J. Zhang, J. Sun, D. H. Zhang, H. M. Liu, and X. L. Gao, "Flatness control of cold rolled strip based on relay optimisation," *Ironmaking Steelmaking*, vol. 45, no. 2, pp. 166–175, Feb. 2018.

[3] Z.-W. Yan, B.-S. Wang, H.-N. Bu, and D.-H. Zhang, "Intelligent assignation strategy of collaborative optimization for flatness control," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 40, no. 3, pp. 1–13, Mar. 2018.

[4] M. Song, H. Liu, Y. Xu, D. Wang, and Y. Huang, "Decoupling adaptive smith prediction model of flatness closed-loop control and its application," *Processes*, vol. 8, no. 8, p. 895, Jul. 2020.

[5] Q.-L. Wang, J. Sun, X. Li, Y.-M. Liu, P.-F. Wang, and D.-H. Zhang, "Numerical and experimental analysis of strip cross-directional control and flatness prediction for UCM cold rolling mill," *J. Manuf. Processes*, vol. 34, pp. 637–649, Aug. 2018.

[6] G. Gruosso, G. S. Gajani, F. Ruiz, J. D. Valladolid, and D. Patino, "A virtual sensor for electric vehicles' state of charge estimation," *Electron.*, vol. 9, no. 2, 2020, Art. no. 278.

[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[8] J. Cao and Z. Lin, "Extreme learning machines on high dimensional and large data applications: A survey," *Math. Problems Eng.*, vol. 2015, pp. 1–13, Mar. 2015.

[9] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, Jan. 2015.

[10] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2016.

[11] J. Zhang, W. Xiao, Y. Li, and S. Zhang, "Residual compensation extreme learning machine for regression," *Neurocomputing*, vol. 311, pp. 126–136, Oct. 2018.

[12] W. Yu, F. Zhuang, Q. He, and Z. Shi, "Learning deep representations via extreme learning machines," *Neurocomputing*, vol. 149, pp. 308–315, Feb. 2015.

[13] Z. Wang, D. Zhang, D. Gong, and W. Peng, "A new data-driven roll force and roll torque model based on FEM and hybrid PSO-ELM for hot strip rolling," *ISIJ Int.*, vol. 59, no. 9, pp. 1604–1613, Sep. 2019.

[14] X. Li, Y. Fang, and L. Liu, "Kernel extreme learning machine for flatness pattern recognition in cold rolling mill based on particle swarm optimization," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 42, no. 5, p. 270, May 2020.

[15] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 8, pp. 1352–1357, Aug. 2009.

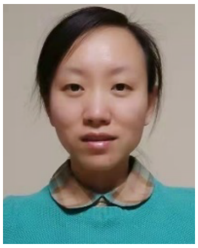
[16] B. Y. Qu, B. F. Lang, J. J. Liang, A. K. Qin, and O. D. Crisalle, "Two-hidden-layer extreme learning machine for regression and classification," *Neurocomputing*, vol. 175, pp. 826–834, Jan. 2016.

[17] D. Xiao, B. Li, and Y. Mao, "A multiple hidden layers extreme learning machine method and its application," *Math. Problems Eng.*, vol. 2017, Jan. 2017, Art. no. 4670187.

[18] S. Lu, X. Wang, G. Zhang, and X. Zhou, "Effective algorithms of the Moore-penrose inverse matrices for extreme learning machine," *Intell. Data Anal.*, vol. 19, no. 4, pp. 743–760, Jul. 2015.

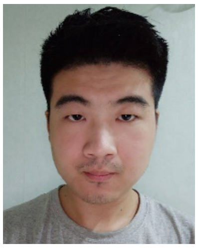
[19] S. Tamura and M. Tateishi, "Capabilities of a four-layered feedforward neural network: Four layers versus three," *IEEE Trans. Neural Netw.*, vol. 8, no. 2, pp. 251–255, Mar. 1997.

[20] G.-B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 274–281, Mar. 2003.
[21] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.
[22] J.-L. Sun, Y. Peng, and H.-M. Liu, "Dynamic characteristics of cold rolling mill and strip based on flatness and thickness control in rolling process," *J. Central South Univ.*, vol. 21, no. 2, pp. 567–576, Feb. 2014.



JINGYI LIU received the B.S. and M.S. degrees from Northeastern University, Shenyang, China, in 2003 and 2008, respectively, and the Ph.D. degree in statistics from Northeast University, in 2019.

She is currently a Lecturer with Northeastern University. Her research interests include engineering numerical calculation and applied mathematics.



LUSHAN WAN was born in Shandong, China, in 1995. He received the B.S. degree in telecommunications and information engineering from the Shenyang University of Technology, Shenyang, China, in 2018. He is currently pursuing the M.S. degree in control theory and control engineering with Northeastern University, Shenyang. His research interests include extreme learning machines and artificial intelligence algorithms.

DONG XIAO (Member, IEEE) received the Ph.D. degree in control theory and control engineering from Northeast University, Shenyang, China, in 2009.

Since 2011, he has been a Professor with the College of Information Science and Engineering, Northeast University. His research interests include neural networks, ELM algorithm, PLS algorithm, and MMMD algorithm.