Control Engineering Practice 105 (2020) 104633

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Robust neural networks with random weights based on generalized M-estimation and PLS for imperfect industrial data modeling

Ping Zhou [a,*], Jin Xie [a], Wenpeng Li [a], Hong Wang [b], Tianyou Chai [a]

[a] *State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, PR China*
[b] *Control System Center, Manchester University, Manchester, M60, 1QD, UK*

ARTICLE INFO

Keywords:
Robust neural networks
Neural networks with random weights (NNRW)
Imperfect data modeling
Generalized M-estimation (GM-estimation)
Input and output outliers
Partial least squares (PLS)
Multicollinearity
Blast furnace ironmaking process

ABSTRACT

Actual industrial data inevitably contain a variety of outliers for various reasons. Even a single outlier may have a large distortion effect on modeling performance with conventional algorithms, not to mention the complicated process modeling by the imperfect industrial data existing various outliers both in input direction and output direction. Therefore, the robustness of the algorithm must be fully considered in modeling of complicated industrial processes. Aiming at this, the robust neural network with random weights based on generalized M-estimation and PLS (GM-R-NNRW) is proposed for data modeling of complicated industrial process, whose samples coexist input and output outliers and have multicollinearity problem. Firstly, the input weights and biases of the proposed GM-R-NNRW are randomly assigned within their respective given ranges. Secondly, the GM-R-NNRW determines the weights of the sample by the residual size of the model and the distance information of the input vector in the high-dimensional space according to the generalized M-estimation. Then these weights were combined to determine the final model contribution of each sample, solving the problem that the samples exist both the input direction and the output direction outliers. Moreover, the improved PLS is used to solve the multicollinearity problem existing in data samples. Finally, both data experiment and actual industrial application have showed that the general approximation performance of the algorithm is greatly improved, and an easy-to-use model with better accuracy and robust performance can be obtained.

1. Introduction

The process industry is an indispensable part of the manufacturing industry which acts as an important pillar industry for national economic and social development. Modern process industry requires environment-friendly, high-efficiency, low-consumption and low-cost industrial production, which makes optimization and control of complicated industrial processes quite critical (Chai, 2009). In fact, neither optimization nor control of complicated industrial processes is easy to achieve as the prerequisite for achieving these goals is to obtain precise measurements of each variable. However, the actual industrial process operating environment and internal mechanism are very complicated, containing various physical and chemical reactions, which make the real-time measurement of each variable is difficult or even impossible to realize. Therefore, it is quite necessary to efficiently obtain the accurate models of the complicated industrial processes (Chen & Ge, 2020; Zhou, Chai, & Wang, 2009). The traditional mechanism models are based on a deep understanding of the process mechanism and ideal simplified assumptions. While the traditional inference models rely entirely on the limited expert knowledge of system integration (Browne & Sun, 2002; Chai, 2009). As most modern industrial processes have extremely complicated nonlinear time-varying dynamics with long time delay, which make the industrial processes are commonly too complicated to fully describe by the traditional mechanism models and the inference models. In addition, even if the mechanism models or the inference models are obtained, the prediction accuracy of the obtained models is not high as they are based on the idealized industrial process (Chai, 2009; Zhou et al., 2009).

Over decades of continuous development of computer technology, the data-driven models have attracted increasing attention and have been applied to many different real-world applications. The data-driven modeling method belongs to black box modeling. It does not need to have a deep understanding of the process mechanism like mechanism modeling. Instead, the process model is directly established by collecting the rich data of industrial process to learn the mapping relationship between input variables and output variables (Chai, 2009; Zhou et al., 2009). A large number of practical applications have proved that this kind of modeling method is simple and highly accurate. Therefore, it is quite a practical way to apply the data-driven models into the process

[*] Corresponding author.
E-mail address: zhouping@mail.neu.edu.cn (P. Zhou).

https://doi.org/10.1016/j.conengprac.2020.104633
Received 30 March 2020; Received in revised form 7 September 2020; Accepted 12 September 2020
Available online xxxx

industry for solving the problems of variable measurement and quality prediction (Muller & Craig, 2015; Zhou, Guo, & Chai, 2018).

So far, there have been many data-driven methods, such as multivariate statistical analysis (Zhao & Sun, 2013; Zhao, Wang, & Zhang, 2009), neural network (Chen & Ge, 2020; Zhang et al., 2020; Zhou et al., 2009; Zhou, Lv, Wang, & Chai, 2017), support vector machine (Zhou, Guo, & Chai, 2018; Zhou, Guo, Wang, & Chai, 2018) and so on. Even some of them have been widely used in many productions process modeling, there are still some shortcomings such as low model learning efficiency, weak generalization, and poor robustness. To overcome these problems, in the 90 s of the last century, the neural networks with random weights (NNRW) based on single-hidden layer feed-ward network (SLFN) was proposed in Pao and Takefuji (1992). The main characteristics of it is that the input weights between the input layer and the hidden layer are randomly generated within a certain range, and the output weights between the hidden layer and the output layer are obtained by least squares estimation. It improves many problems existing in most neural network modeling, such as slow convergence, weak generalization, and poor practicability (Igelnik & Pao, 1995; Pao & Takefuji, 1992; Schmidt, Kraaijveld, & Duin, 1992).

However, in data modeling of the actual complicated industrial processes, the model estimates of NNRW as well as other data-driven modeling algorithms are distorted or difficult to estimate accurately in many cases. This is mainly because that the actual industrial data is usually imperfect and does not meet some statistical assumptions, such as normal distribution, statistical independence and so on. Specifically, the actual industrial data usually have the following two problems:

(1) ***Multicollinearity problem***: This means that precise correlations or highly correlated relationships exist between the actual industrial production data. The occurrence of multicollinearity leads to an increase in the variance of the estimated parameters, and the instability of estimation property. Moreover, the significance test of the variables will be meaningless, and the important variables may be excluded, which is not conducive to prediction and statistical analysis. The solutions for the multicollinearity problem explored by scholars are mainly three methods, that is, principle component regression, ridge regression and partial least squares (PLS) regression (Aguilera, Escabias, & Valderrama, 2006; Gou & Fyfe, 2004; Katrutsa & Strijov, 2017). Among these methods, the PLS is a commonly used technique for high-dimensional data modeling, in which the LS is used in the calculation of weights, loads, scores and review coefficient. It can find potential variables by simultaneously decomposing the input space and output space while maintaining the orthogonality constraint and the correlation between variables (Bastien, Vinzi, & Tenenhaus, 2005; Westerhuis, Kourti, & Macgregor, 1998; Zhao et al., 2009), which can effectively solve the adverse effects of multicollinearity on modeling.

(2) ***Outliers problem, especially the outliers coexisting in both process inputs and outputs***: Another more common practical problem that needs much more special attention is that in the actual industrial production, due to the failure of devices such as measuring instruments and transmitters and other abnormal disturbances, a small amount of data is far away from most of the data, known as *outliers* (Chen & Ge, 2020; Filzmoser & Todorov, 2013; Zhou, Guo, & Chai, 2018; Zhou, Guo, Wang et al., 2018; Zhou et al., 2017). These outliers seriously affect the performance of data modeling using the conventional algorithms (Filzmoser & Todorov, 2013). Actually, even a single output direction outlier will have a large distortion effect on classical statistical modeling methods that are optimal under normal or linear assumptions, not to mention the complicated process modeling by the actual industrial data coexisting various outliers in both input direction and output direction. Therefore, the robustness of the algorithms must be fully considered in modeling of complicated industrial processes, avoiding the influence of outliers on modeling accuracy. For the specific NNRW algorithm, the output weights of it are obtained by LS method which is susceptible to outliers and thus has poor robustness. In order to apply the NNRW to the actual complicated industrial process, in addition to the above improvement for multicollinearity problem, further improvement must be made to guarantee the robustness of the algorithm. In recent robust modeling studies like Yu and Zhao (2020), a quite representative autoencoder method is proposed. While it is suitable and efficient for robust monitoring and fault isolation, this paper will mainly focus on robust modeling method for quality prediction in the actual complicated industrial process. In Wakelinc and Macfie (1992) and Daszykowski, Heyden, and Walczak (2007), the robust regression methods are proposed to replace the LS method used in PLS regression. However, these methods have high calculation requirements, and the efficiency of the robust step is low. Among the existing robust modeling methods, the M-estimation is the most commonly used method and can be re-weighted by iteration (Filzmoser & Todorov, 2013; Frahm, Nordhausen, & Oja, 2020; Huber & Ronchetti, 2009; Zhou, Guo, & Chai, 2018; Zhou, Guo, Wang et al., 2018; Zhou et al., 2017). However, the M-estimation only determines the modeling weight of the sample according to the distribution of the normalized residual, thereby realizing the weight reduction processing for the output outlier samples (Y-direction). This means that the abnormality of the input sample (X-direction) is not considered, that is, the M-estimation is a robust estimation method from the Y-direction of the sample under the assumption that the sample X-direction is normal. In most cases, the abnormality of X-direction and Y-direction coexists in the outlier samples and will have great impact on the modeling performances. Unfortunately, to our best knowledge, there is no literature about robust modeling by actual industrial data with outliers in both input and output directions.

In order to solve the above practical engineering modeling problems, this paper proposes a robust modeling strategy based on generalized M-Estimation (GM-estimation) and PLS for the existence of multiple outliers and multicollinearity in the data. That is, the robust neural networks with random weights based on GM-estimation and PLS (GM-R-NNRW). Fig. 1 shows the global strategy structure of the proposed GM-R-NNRW. Firstly, the GM-R-NNRW determines the weights of the sample by the residual size of the model output vector and the distance information of the input vector in the high-dimensional space according to the GM-estimation. Combining these weights to determine the final model contribution of each sample, so that it can solve the problem that the samples exist both the input direction and the output direction outliers, greatly improving the robustness and prediction accuracy of the model. At the same time, the improved PLS is presented to solve the multicollinearity problem exist in the data. Therefore, the general approximation performance of the algorithm is greatly improved, and a simple model with better accuracy and robust performance is obtained.

2. PLS-NNRW

2.1. The basic neural networks with random weights (NNRW)

Different from the traditional SLFN using the backpropagation iteration of the error to find the network parameters, the NNRW randomly assigns the input weights and biases of the network within a certain range, the output weights of the network are obtained by using the least squares (LS) method. While ensuring good universal approximation performance, this random learning network solves the problems that the traditional SLFN converges slowly, easy to fall into the local minimum and sensitive to the learning rate.

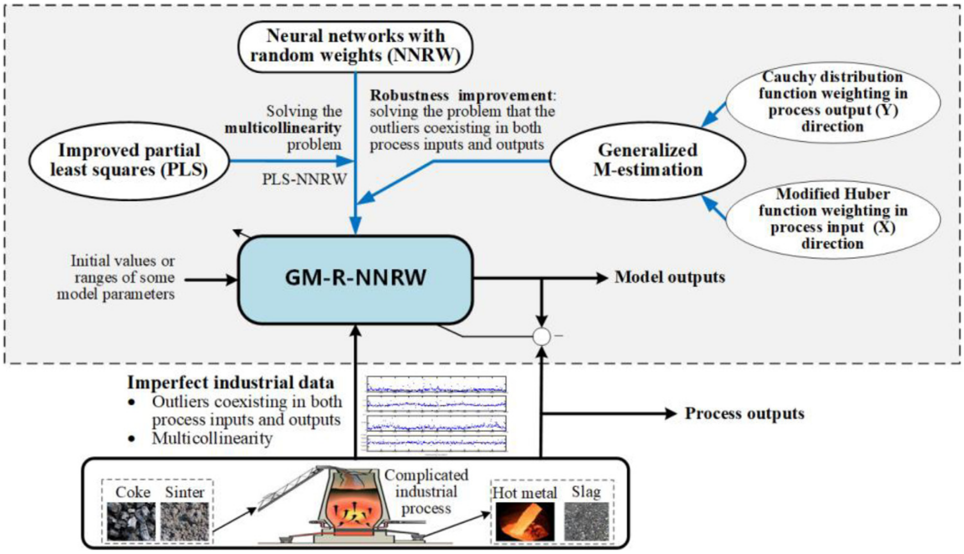

Neural networks with random weights (NNRW)
Solving the multicollinearity problem
Robustness improvement: solving the problem that the outliers coexisting in both process inputs and outputs
Cauchy distribution function weighting in process output (Y) direction
Improved partial least squares (PLS)
PLS-NNRW
Generalized M-estimation
Modified Huber function weighting in process input (X) direction
Initial values or ranges of some model parameters
GM-R-NNRW
Model outputs
−
Imperfect industrial data
• Outliers coexisting in both process inputs and outputs
• Multicollinearity
Process outputs
Coke
Sinter
Complicated industrial process
Hot metal
Slag

Fig. 1. Strategy structure of the proposed GM-R-NNRW.

Given N different sets of training samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)|(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathbb{R}^d \times \mathbb{R}^m, i = 1, 2, \ldots, N\}$, where $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,d}]^T$ is the d-dimensional input vector, and $\boldsymbol{y}_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,m}]^T$ is the m-dimensional output vector. For a SLFN-based NNRW, the network output can be expressed as

$$\boldsymbol{f}_{L,i}(\boldsymbol{x}_i) = \sum_{j=1}^{L} \boldsymbol{\beta}_j g(\boldsymbol{w}_j \cdot \boldsymbol{x}_i + b_j), i = 1, 2, \ldots, N \tag{1}$$

where L is the number of hidden neurons, $\boldsymbol{w}_j$ and b_j respectively denote the input weights and biases between the input layer and the jth hidden neuron, $\boldsymbol{\beta}_j$ is the output weights between the output layer and the jth hidden layer neuron, and $g(\cdot)$ is the activation function of the hidden neurons.

In order to complete the establishment of NNRW, the output of the network should be infinitely approximated to the actual sample output of the training set with 0 errors, that is $\boldsymbol{E}_L = \sum_{i=1}^{M} \|\boldsymbol{f}_L - \boldsymbol{y}_i\| = 0$. Therefore, according to Eq. (1), the establishment process of NNRW can be transformed into finding the corresponding $\boldsymbol{w}_j, b_j$ and $\boldsymbol{\beta}_j$ satisfying

$$\sum_{j=1}^{L} \boldsymbol{\beta}_j g(\boldsymbol{w}_j \cdot \boldsymbol{x}_i + b_j) = \boldsymbol{y}_i, i = 1, 2, \ldots, N \tag{2}$$

which can be abbreviated as

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{Y} \tag{3}$$

where $\boldsymbol{H}$ is the hidden output matrix, $\boldsymbol{\beta}$ is the output weight matrix which needs to be obtained, and $\boldsymbol{Y}$ represents the real output matrix of the sample.

$$\begin{aligned} &\boldsymbol{H}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_L, \boldsymbol{x}_1, \ldots \boldsymbol{x}_N, b_1, \ldots, b_L) \\ &= \begin{bmatrix} g(\boldsymbol{w}_1 \cdot \boldsymbol{x}_1 + b_1) & \cdots & g(\boldsymbol{w}_L \cdot \boldsymbol{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{w}_1 \cdot \boldsymbol{x}_N + b_1) & \cdots & g(\boldsymbol{w}_L \cdot \boldsymbol{x}_N + b_L) \end{bmatrix}_{N \times L}, \\ &\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix}_{L \times m}, \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1^T \\ \vdots \\ \boldsymbol{y}_N^T \end{bmatrix}_{N \times m} \end{aligned} \tag{4}$$

Since $\boldsymbol{w}_j$ and b_j of NNRW are randomly assigned within a certain range, the establishment of NNRW can be completed after obtain the corresponding $\boldsymbol{\beta}$ matrix by applying the LS method of the following Eq. (5).

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{H}\boldsymbol{\beta} - \boldsymbol{Y}\|^2 = \boldsymbol{H}^{\dagger}\boldsymbol{Y} \tag{5}$$

where $\boldsymbol{H}^{\dagger} = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T$ is the *Moore–Penrose* generalized inverse of the hidden layer output matrix $\boldsymbol{H}$.

2.2. The proposed PLS-NNRW

The basic NNRW completes the establishment of the network by using the LS method to obtain the output weights of the network. It is generally accepted that under the premise of Gauss–Markov theorem, the classic LS estimation has the advantages of unbiased, uniform and minimum variance, and enjoys a high status in the research field. But the LS estimation has the problems of multicollinearity and poor robustness. In this section, the solution of multicollinearity problem is mainly discussed.

Remark 1. Multicollinearity (also collinearity) is the occurrence of high intercorrelations among independent variables in a multiple regression model, which can lead to the model estimation distorted or difficult to estimate accurately. In practical industrial modeling applications, due to factors such as measurement lag or sample collection range limitations, there are often correlations between process variables. Therefore, directly obtaining the output weights of the network by LS method has serious multicollinearity problem, which seriously affects the modeling result. □

This paper introduces an improved partial least squares (PLS) into NNRW to propose the PLS-NNRW, solving the above multicollinearity problem. Conventional PLS is a commonly used statistical analysis technique that projects information from a high-dimensional data space into a low-dimensional space defined by a few principal components and integrates multiple regression and principal component analysis. The main idea of PLS is to extract the principal variables with the largest variance from the input variable set and the output variable set, ensuring that the covariance between the principal elements of the input and output variables is the largest under the condition of satisfying certain orthogonality and normalized constraint. The components and residual matrices of improved PLS regression have many excellent properties, one of which is that the components are mutually orthogonal, which eliminates the multiple linear correlation to a certain extent (Bastien et al., 2005; Westerhuis et al., 1998).

Again, given N different sets of training samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)|(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathbb{R}^d$
$\times \mathbb{R}^m, i = 1, 2, \dots, N\}$, for a PLS-NNRW with L hidden neurons can be expressed as

$$\sum_{j=1}^{L} \beta_j g(\boldsymbol{w}_j \cdot \boldsymbol{x}_i + b_j) = \boldsymbol{o}_j, i = 1, 2, \dots, N \tag{6}$$

where $\boldsymbol{o}_j$ is the estimated output of the network.

The network hidden output matrix $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_L]$ and the network output matrix $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_m]$ can also be represented by Eq. (4). After getting the hidden output matrix and the model output matrix, the following decomposition is performed.

$$\begin{cases} \boldsymbol{H} = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E} = \sum_{i=1}^{A} \boldsymbol{t}_i \boldsymbol{p}_i^T + \boldsymbol{E} \\ \boldsymbol{Y} = \boldsymbol{U}\boldsymbol{Q}^T + \boldsymbol{F} = \sum_{i=1}^{A} \boldsymbol{u}_i \boldsymbol{q}_i^T + \boldsymbol{F} \end{cases} \tag{7}$$

where A is the number of the main elements, $\boldsymbol{P} \in \mathbb{R}^{L\times A}$ is the load matrix of the hidden output matrix, $\boldsymbol{u}_i$ is the principal element of the output matrix, and $\boldsymbol{Q} \in \mathbb{R}^{m\times A}$ is the load matrix of the network output matrix. Moreover, $\boldsymbol{E}$ and $\boldsymbol{F}$ respectively represent the residual of the hidden output matrix and the network output matrix, and $\boldsymbol{t}_i$ is a latent variable sequentially extracted from the data matrix according to the maximum covariance value of $\boldsymbol{t}_i$ and $\boldsymbol{u}_i$ under the constraint of orthogonality and normalization, i.e.

$$\max\{\mathrm{Cov}(\boldsymbol{t}_i, \boldsymbol{u}_i)\} \\ \text{s.t.} \begin{cases} \boldsymbol{t}_i = \boldsymbol{E}_{i-1}\boldsymbol{k}_i \\ \boldsymbol{u}_i = \boldsymbol{F}_{i-1}\boldsymbol{c}_i \\ \|\boldsymbol{k}_i\| = 1 \\ \|\boldsymbol{c}_i\| = 1 \end{cases} \tag{8}$$

where $\boldsymbol{E}_i$ and $\boldsymbol{F}_i$ respectively represent the residual of the hidden output matrix and the network output matrix after extracting $i-1$ principal elements, $\boldsymbol{t}_i$ and $\boldsymbol{k}_i$ respectively represent the score vector of the hidden output matrix after extracting $i-1$ principal elements and their corresponding load vectors. Moreover, $\boldsymbol{u}_i$ and $\boldsymbol{c}_i$ respectively represent the score vector of the network output matrix after extracting $i-1$ principal elements and their corresponding load vectors, so the PLS regression model between the final hidden output matrix and the network output can be expressed as

$$\begin{cases} \boldsymbol{Y} = \boldsymbol{H}\boldsymbol{\beta} + \boldsymbol{F} \\ \boldsymbol{\beta} = \boldsymbol{K}(\boldsymbol{P}^T\boldsymbol{K})\boldsymbol{Q}^T \end{cases} \tag{9}$$

where $\boldsymbol{K} \in \boldsymbol{R}^{n\times A}, \boldsymbol{P} \in \boldsymbol{R}^{n\times A}$ and $\boldsymbol{Q} \in \boldsymbol{R}^{m\times A}$ are the load matrices corresponding to each load vector, and the output weight matrix $\boldsymbol{\beta}$ is obtained by nonlinear iterative PLS algorithm (Wold, 1973).

3. The proposed GM-R-NNRW

In the actual industrial production, due to the failure of devices and other abnormal disturbances, a small amount of data is generally far away from most of the data, known as *outliers*. These outliers seriously affect the performance of data modeling using the conventional algorithms. Therefore, the robustness of the PLS-NNRW algorithm must be fully considered in modeling of complicated industrial processes, avoiding the influence of outliers on modeling accuracy.

Remark 2. For outliers, the conventional and simple processing method is to remove outliers by data preprocessing, so as to improve the accuracy of data modeling. However, actual industrial measurement data is often extraordinarily complex, and it is difficult to find all outliers using traditional manual deletion of outliers. In addition, some abnormal data detected in actual industrial data are not necessarily outliers, there may be useful information contained in, that is, information that can correctly reflect the characteristics of data distribution and can contribute to improving modeling accuracy. If these available information are artificially excluded, it will lead to poor data modeling. □

Remark 3. Robust estimation is simpler than the traditional method of eliminating outliers. Robust estimation can play a smoothing role in rejecting or accepting a data and can retain some data that is not of good quality but can still be used. In addition, it is quite difficult to accurately determine the proportion of valid and harmful information in the data beforehand, and to determine in which observation samples such harmful information is specifically contained. □

The M-estimation proposed by Huber in 1964 is a widely used robust estimation algorithm due to its high efficiency and easy implementation (Huber & Ronchetti, 2009). The M-estimation and its various improved versions construct the even function of the residual by improving the LS optimization objective function, so that can treat different samples differently, and reduce the influence of outlier samples on the modeling results, thus the robust performance of estimation is enhanced.

3.1. M-Estimation

For the NNRW, the M-estimation optimization objective function is defined as follows

$$J = \sum_{i=1}^{N} \rho(\boldsymbol{r}_i) = \sum_{i=1}^{N} \rho(\boldsymbol{y}_i - \boldsymbol{h}_i^T\boldsymbol{\beta}) \tag{10}$$

where $\rho(\cdot)$ is the influence function for the M-estimation and $\boldsymbol{r}$ is the residual term.

In order to ensure the scale equivariance of the M-estimation result, that is, the regression coefficient is independent from the unit of the dependent variable. The robust scale estimation $\hat{\sigma}$ from robust estimation theory is introduced, which is generally taken as Median Absolute Deviation (MAD) (Rousseeuw & Croux, 1993; Zhou et al., 2017) divided by the value 0.6745, namely:

$$\hat{\sigma} = \frac{\mathrm{MAD}}{0.6745} = \frac{\mathrm{median}_i\,|\boldsymbol{r}_i - \mathrm{median}(\boldsymbol{r}_i)|}{0.6745} \tag{11}$$

where median$(\cdot)$ is the median function.

Remark 4. In Eq. (11), MAD is more suitable for the outliers in the dataset than the standard deviation, and the value 0.6745 is the MAD value for standard normal distribution. When the residual obeys the Gaussian distribution, dividing the MAD by 0.6745 ensures the consistency of the parameter estimates. □

The residual r_i is divided by the robust scale estimate $\hat{\sigma}$ to obtain a standardized residual $r_i/\hat{\sigma}$. The solution of $\boldsymbol{\beta}$ at this point becomes:

$$\hat{\boldsymbol{\beta}}_M = \arg\min_{\beta} \sum_{i=1}^{N} \rho(\frac{\boldsymbol{y}_i - \boldsymbol{h}_i^T\boldsymbol{\beta}}{\hat{\sigma}}) = \arg\min_{\beta} \sum_{i=1}^{N} \rho(\frac{\boldsymbol{r}_i(\boldsymbol{\beta})}{\hat{\sigma}}) \tag{12}$$

Taking the derivative of $\boldsymbol{\beta}$, and set it be 0, obtaining

$$\sum_{i=1}^{N} \varphi(\frac{\boldsymbol{r}_i(\boldsymbol{\beta})}{\hat{\sigma}})\boldsymbol{h}_i^T = 0 \tag{13}$$

where $\varphi = \rho'$ is a score function, let $d(r) = \varphi(r)/r$, so that Eq. (13) can be converted to

$$\sum_{i=1}^{N} d(\frac{\boldsymbol{r}_i(\boldsymbol{\beta})}{\hat{\sigma}})(\boldsymbol{y}_i - \boldsymbol{h}_i\boldsymbol{\beta})\boldsymbol{h}_i^T = 0 \tag{14}$$

where the value of $d(\cdot)$ is the modeling weight, and Eq. (14) can be converted to

$$\boldsymbol{H}^T\boldsymbol{D}(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{\beta}) = 0 \tag{15}$$

Therefore, for M-estimation, the regression coefficient $\hat{\boldsymbol{\beta}}_M$ can be iteratively calculated by the following formula

$$\hat{\boldsymbol{\beta}}_M = (\boldsymbol{H}^T \boldsymbol{D} \boldsymbol{H})^{-1} \boldsymbol{H}^T \boldsymbol{D} \boldsymbol{Y} \tag{16}$$

where the modeling weight matrix $\boldsymbol{D}$ is a diagonal matrix whose elements on the diagonal are $d_i, i = 1, 2, \dots, N$.

3.2. Generalized M-estimation and the proposed GM-R-NNRW

It can be seen from the above that the M-estimation can only realize robust estimation by weight reduction processing from the output (Y) direction of the sample under the assumption that the X-direction is normal. However, in most actual industrial cases, the abnormality of X-direction and Y-direction coexists in the samples and will have great impact on the modeling results of the model. Therefore, in order to simultaneously consider these two kinds of outlier samples, this paper introduces Generalized M-estimation (GM-estimation) into the PLS-NNRW network to further improve the robustness of the algorithm. The GM-estimation is an improved estimation of the conventional M-estimation, it can enhance the robustness of the outliers with abnormality in input direction and output direction at the same time.

In order to realize the weight reduction processing for the X-direction outliers of the independent variable, the above M-estimation Eq. (13) is rewritten into its Mallows form, i.e.

$$\sum_{i=1}^{N} \varphi\left(\frac{\boldsymbol{r}_i(\boldsymbol{\beta})}{\hat{\sigma}}\right) \cdot v(\boldsymbol{h}_i^T)\boldsymbol{h}_i^T = 0 \tag{17}$$

where $v(\boldsymbol{h}_i^T)$ denotes the modeling weight of $\boldsymbol{h}_i^T$ determined by the position of each sample point in the high-dimensional space of the hidden output matrix (X-direction). If the sample point is abnormal in the X-direction, then its corresponding $\boldsymbol{h}_i^T$ will deviate from most data in the high dimensional space, and the value of $v(\boldsymbol{h}_i^T)$ will be smaller, even 0. That is, the sample modeling weight is determined by the degree of X-direction anomaly in the hidden layer output space. At the same time, the GM-estimation retains φ function in the M-estimation and inherits many of the advantages of the M-estimation, with higher estimation efficiency and crash point.

According to the above derivation, it can obtain

$$\sum_{i=1}^{N} d\left(\boldsymbol{r}_i(\boldsymbol{\beta})/\hat{\sigma}\right) \times (\boldsymbol{y}_i - \boldsymbol{h}_i\boldsymbol{\beta})\boldsymbol{h}_i^T \cdot v(\boldsymbol{h}_i^T) = 0 \tag{18}$$

where $d(\cdot)$ is the modeling weight for each output sample, $v(\cdot)$ is the modeling weight corresponding to each input sample, and the above formula can be simplified to

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{V}\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{V}\boldsymbol{Y} \tag{19}$$

Therefore, the network output weight $\boldsymbol{\beta}$ of GM-estimation NNRW can be obtained by the following iterative expression

$$\hat{\boldsymbol{\beta}}_{\mathrm{GM}} = \left(\boldsymbol{H}^T \boldsymbol{D} \boldsymbol{V} \boldsymbol{H}\right)^{-1} \boldsymbol{H}^T \boldsymbol{D} \boldsymbol{V} \boldsymbol{Y} \tag{20}$$

where $\boldsymbol{D} = \mathrm{diag}\{d_1, d_2, \dots, d_N\}$ and $\boldsymbol{V} = \mathrm{diag}\{v_1, v_2, \dots, v_N\}$ respectively denote the modeling weight matrix corresponding to the X-direction and the Y-direction of the sample.

Remark 5. The optimization objective function of the basic LS based NNRW, the M-estimation based robust NNRW, and the GM-estimation based robust NNRW can be summarized as shown in Eqs. (21)∼(23). It can be seen that both the M-estimation and the GM-estimation weight the sample to determine the contribution of the sample to the modeling. Automatically determine valid data, suspicious data, and abnormal data through robust algorithms, and retain useful data information, reduce suspicious data information, perform zero weight or near zero weight processing of abnormal data information. Therefore, the robust modeling method is more accurate and convenient than the modeling method that uses the anomaly detection algorithm to identify the outliers and then manually replace the values.

$$\hat{\boldsymbol{\beta}}_{\text{LS-NNRW}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} (\boldsymbol{y}_i - \boldsymbol{h}_i\boldsymbol{\beta})^2 \tag{21}$$

$$\hat{\boldsymbol{\beta}}_{\text{M−NNRW}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} d_i(\boldsymbol{y}_i - \boldsymbol{h}_i\boldsymbol{\beta})^2 \tag{22}$$

$$\hat{\boldsymbol{\beta}}_{\text{GM-NNRW}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} d_i v_i(\boldsymbol{y}_i - \boldsymbol{h}_i\boldsymbol{\beta})^2 \tag{23}$$

In order to achieve the optimization of algorithm, the sample X-direction weight matrix $\boldsymbol{V}$ and the sample Y-direction weight matrix $\boldsymbol{D}$ need to be accurately obtained to achieve accurate weighting of each modeling data. Consequently, the selection of the weight functions $d_i(\cdot)$ and $v_i(\cdot)$ is critical to the computational efficiency of the algorithm and the ultimate robustness of the model.

(1) *Modified Huber function weighting in X-direction*: Common weight functions are Huber weight function, Hampel weight function and Tukey dual weight function (Pitselis, 2013; Zhou, Guo, & Chai, 2018; Zhou, Guo, Wang et al., 2018; Zhou et al., 2017). Among them, the Huber weight function is more suitable for data including outliers and non-strict normal distributions. The Huber weight function is combined with the adjustment parameter c and the spatial median to achieve a good balance between the robust performance and computational efficiency of the algorithm (Serneels, Croux, Filzmoser, & VanEspen, 2005). Therefore, the weighting factor $v(h_i^T)$ of the sample X-direction can be determined by the modified Huber weight function shown in Eq. (24).

$$f_{\text{Huber}}(u, c) = \begin{cases} 1, & |u - \mu| \le c \\ \dfrac{c}{|u - \mu|}, & |u - \mu| > c \end{cases} \tag{24}$$

where μ is the median of the variable μ, and the adjustment parameter c is a constant. From this, the formula for calculating the sample weight is

$$v_i = f_{\text{Huber}}(0.6745 \times \frac{\|\boldsymbol{t}_i - \mathrm{med}_{L1}(\boldsymbol{T})\|}{\mathrm{median}_i \left\|\boldsymbol{t}_i - \mathrm{med}_{L1}(\boldsymbol{T})\right\|}, c) \tag{25}$$

where $||.||$ is the Euclidean norm, $\mathrm{med}_{L1}(\boldsymbol{T})$ is the $L1$-median of data set $\{\boldsymbol{t}_1, \boldsymbol{t}_2 \cdots, \boldsymbol{t}_n\}$. It should be noted that the $L1$-median (geometric median) is the protrusion form univariate median to high latitude, which is a multivariate position robust estimate with good statistical properties. The calculation of it is that for the data set $\boldsymbol{X} = \{x_1, x_2 \cdots, x_n\}, x_i \in \mathbb{R}^P$, obtaining μ that satisfies the following conditions, that is

$$\mu(\boldsymbol{X}) = \arg\min_{\mu} \sum_{i=1}^{n} \|x_i - \mu\| \tag{26}$$

Actually, the $L1$-median represents a data point whose sum of Euclidean distances to the remaining N data points are the smallest. If the data point is non-collinear, then the solvation of Eq. (26) is unique, the point of collapse is 0.5, the scale equivariance and the position invariance will be satisfied.

(2) *Cauchy distribution function weighting in Y-direction*: The above-mentioned Huber weight function based weighting method essentially realizes the weighting of the entire sample data by the prediction error of a single sample, but the sample points with the prediction error at the center tend to better characterize the overall characteristics of the data. The Huber weighting may also perform weight reduction processing on these samples and lose some sample information in the network training process, resulting in distortion or over-fitting of the model. Simultaneous selection of the Huber weight function in the X, Y-directions will have a certain impact on the accuracy of the model. In the previous work, the weighting method based on Cauchy distribution has been applied in Zhou et al. (2017) and obtained a good modeling effect. Therefore, this paper introduces the weighting method of Cauchy

Table 1
The proposed GM-R-NNRW Algorithm.

```
Algorithm 1: GM-R-NNRW algorithm
Given the modeling input dataset $\boldsymbol{X}=[\boldsymbol{x}_1^T,\boldsymbol{x}_2^T,\cdots,\boldsymbol{x}_i^T,\cdots,\boldsymbol{x}_N^T]_{N\times d}^T, \boldsymbol{x}_i=[x_{i,1},x_{i,2},\cdots,x_{i,d}]^T\in\mathbb{R}^d$ and the modeling output data set
$\boldsymbol{Y}=[\boldsymbol{y}_1^T,\boldsymbol{y}_2^T,\cdots,\boldsymbol{y}_i^T,\cdots,\boldsymbol{y}_N^T]_{N\times m}^T, \boldsymbol{y}_i=[y_{i,1},y_{i,2},\cdots,y_{i,m}]^T\in\mathbb{R}^m$. And the number of network hidden neurons is $L$.
1. Randomly generate $\boldsymbol{w}_L$ and $b_L$ from $[-1,1]^d$ and $[-1,1]$ respectively, obtain $\boldsymbol{H}$ by Eq.(4).
2. Calculate the LS output weight $\boldsymbol{\beta}_{\mathrm{LS}}=\boldsymbol{H}^{\dagger}\boldsymbol{Y}$ , the corresponding $\boldsymbol{Y}_{\mathrm{LS}}=\boldsymbol{H}\boldsymbol{\beta}_{\mathrm{LS}}$ and $\boldsymbol{e}_{\mathrm{LS}}=\boldsymbol{Y}-\boldsymbol{Y}_{\mathrm{LS}}$.
3. Initialize the output weight $\boldsymbol{\beta}_{\mathrm{GMR}}=[\boldsymbol{\beta}_{\mathrm{LS}};\mathrm{ones}(1,m)]$, $\boldsymbol{\beta}_{\mathrm{GMR}}^{*}=\mathrm{ones}(L+1,m)$, set $\boldsymbol{Y}_{\mathrm{GMR}}=\boldsymbol{Y}_{\mathrm{LS}}$, the index of iteration times
      $k=0$, the number of components $A=10$.
4. While $\max(|\boldsymbol{\beta}_{\mathrm{GMR}}-\boldsymbol{\beta}_{\mathrm{GMR}}^{*}|)>0.001$
5.    Let $k=k+1$
6.    If $k\geq 50$
7.       Break (go to Step 25)
8.    End If
9.    For $c=1,2,\cdots m$, Do
10.      If $k=1$
11.         $\boldsymbol{t}^{u\times p}=\boldsymbol{H}^T$
12.      Else $\boldsymbol{t}^{u\times p}=\boldsymbol{T}(:,A\times(c-1)+1:A\times c)$
13.      End If
14.   Calculate the robust scale estimation $\hat{\sigma}_t$ of $\boldsymbol{t}$ by Eq.(11) so as to calculate $\mu_t=\mathrm{median}(\hat{\sigma}_t)$.
15.   Calculate X-direction weight $\boldsymbol{V}_i(i=1,2,\cdots,N)$ by Eq.(24) and Eq.(25) for the X-direction
         weight matrix $\boldsymbol{V}$.
16.   Renew $\boldsymbol{\beta}_{\mathrm{GMR}}=\boldsymbol{\beta}_{\mathrm{GMR}}^{*}$ and $\boldsymbol{e}_{\mathrm{GMR}}=\boldsymbol{Y}-\boldsymbol{Y}_{\mathrm{GMR}}$.
17.   Calculate the robust scale estimation $\hat{\sigma}_e$ of $\boldsymbol{e}_{\mathrm{GMR}}$ by Eq.(11) so as to calculate $\mu_e=\mathrm{median}(\hat{\sigma}_e)$
         and obtain $a_e$ by Eq.(31).
18.   Calculate Y-direction weight $d_i(i=1,2,\cdots,N)$ by Eq.(27) for the Y-direction weight matrix $\boldsymbol{D}$.
19.   For $i=1,2,\cdots,m$, Do
20.      Calculate $\boldsymbol{\beta}_{\mathrm{GMR}}^{*}(:,i)$ and $\boldsymbol{t}_i$ by the PLS algorithm.
21.      Obtain the corrected scoring vector as $\boldsymbol{T}(:,A\times(i-1)+1:A\times i)=(\mathrm{diag}(\sqrt{\boldsymbol{DV}}))^{-1}\times\boldsymbol{t}_j$
22.   End For
23.   Update the output matrix $\boldsymbol{Y}_{\mathrm{GMR}}=\boldsymbol{H}^T\times\boldsymbol{\beta}_{\mathrm{GMR}}^{*}$.
24. End While
25. Return $\boldsymbol{Y}_{\mathrm{GMR}}=[\boldsymbol{y}_1,\boldsymbol{y}_1,\cdots,\boldsymbol{y}_m]$.
```

distribution into the weighting factor determination of the sample Y-direction. The expression is as follow

$$d_i = f_{\mathrm{Cauchy}}(\boldsymbol{t}_i, a, \mu) = \frac{a}{\pi} \cdot \frac{1}{a^2 + (\boldsymbol{t}_i - \mu)^2}, (x \in \mathbb{R}, \mu \in \mathbb{R}, a > 0) \tag{27}$$

In Eq. (27), if the parameters μ and a are known, the Cauchy function curve can be determined, and it has the properties similar to those of a normal distribution curve, that is

- The curve is symmetrical about $x = \mu$, and for any $h > 0$

$$P\{\mu - h < X \leq \mu\} = P\{\mu < X \leq \mu + h\} \tag{28}$$

- When $x = \mu$

$$f_{\mathrm{Cauchy}}(x)_{\max} = f_{\mathrm{Cauchy}}(\mu) = 1/\pi a \tag{29}$$

For the determination of the Cauchy distribution function the parameters μ and a, the parameter μ should be the standardized residual $r_i/\hat{\sigma}$, that is

$$\mu = \mathrm{median}(r_i/\hat{\sigma}_i) \tag{30}$$

And the parameter a determines the shape of the weight function curve, its value is determined as

$$a = \frac{1}{\sqrt{\sum_{i=1}^{N} \left\| r_i/\hat{\sigma}_i - \sum_{i=1}^{N}(r_i/\hat{\sigma}_i)/N \right\|^2 / N}} \tag{31}$$

Based on the obtained sample X-direction weight matrix $\boldsymbol{V}$ and the sample Y-direction weight matrix $\boldsymbol{D}$, the output weight can be calculate by the PLS function, that is

$$\boldsymbol{\beta}_{\mathrm{GMR}} = f_{\mathrm{PLS}}(\mathrm{diag}(\sqrt{\boldsymbol{DV}}) \times \boldsymbol{H}^T, \mathrm{diag}(\sqrt{\boldsymbol{DV}}) \times \boldsymbol{t}_i^T, A) \tag{32}$$

where $\boldsymbol{t}_i$ represents the score vector of the hidden layer output matrix $\boldsymbol{H}$ and A is the number of components.

3.3. Implementation steps of the proposed GM-R-NNRW algorithm

In summary, in order to eliminate the influence of outliers both in the X- and Y- directions of the samples on the model and to solve the multicollinearity problem, this paper combines the GM-estimation of modified Huber weight function and Cauchy distribution weight function into the PLS-NNRW, proposing the GM-R-NNRW algorithm. The GM-R-NNRW firstly determines the weights of samples by the residual value of the model and the distance information of the input vector in the high-dimensional space according to the GM-estimation. Combining these weights to determine the final model contribution of the sample, so that it can eliminate the influence of the various outliers with abnormality in both input direction and output direction, and greatly improve the robustness and prediction accuracy of the model. Moreover, the improved PLS is used to solve the collinearity problem in the data. Therefore, the general approximation performance of the algorithm is greatly improved, and a sample model with better accuracy and robust performance is obtained. The global strategy structure of the proposed GM-R-NNRW algorithm has been shown in Fig. 1, and its pseudocode is given in the following Table 1.

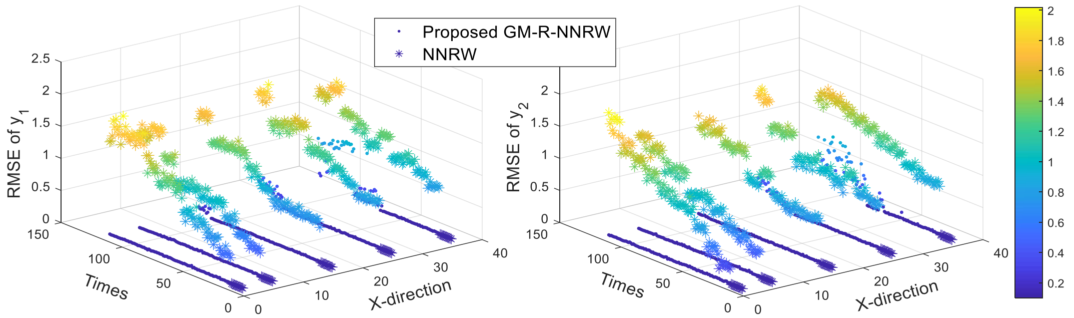


Fig. 2. Testing RMSE comparison of NNRW and GM-R-NNRW under different X, Y-direction outlier proportions. (The Times axis corresponds to an outlier of 5% in the Y-direction for every 10 scales.)

Remark 6. From the point of view of practicability and ease of use, the proposed GM-R-NNRW algorithm can effectively deal with outliers data and multicollinearity problems, so it does not need too much data preprocessing before modeling, such as data dimensionality reduction, data outliers manual elimination, etc., which are required for most conventional modeling algorithms. Therefore, the proposed algorithm is easy to be implemented and applied in practice. In fact, as long as the actual industrial data are matched one by one according to the input and output, the proposed algorithm can be directly used to model this original data. As the trained model is applied in practice, it can directly analyze and calculate the obtained industrial data in field, and output the prediction results quickly and accurately, no matter how bad the data quality due to the input and output outliers and multicollinearity. □

4. Function approximation experiment

Aim at comprehensively verify the performance of the proposed GM-R-NNRW algorithm, the experimental results of two normalization problems will be given, including a simulation function approximation experiment in this section and an actual industrial experiment for quality modeling of blast furnace ironmaking process in the next section.

4.1. Model establishment

The function approximation experiment uses a two-input and two-output real-valued function model, and the simulation data is generated by the following nonlinear function.

$$\begin{cases} y_1(k) = \dfrac{y_1(k-1)}{1+y_2^2(k-1)} + u_1(k) + 0.2u_2(k) + 0.4u_1(k-1) \\ \qquad + 0.1u_2(k-1) + e_1(k) \\ y_2(k) = \dfrac{y_1(k-1)y_2(k-1)}{1+y_2^2(k-1)} + 0.3u_1(k) + u_2(k) + 0.1u_1(k-1) \\ \qquad + 0.5u_2(k-1) + e_2(k) \end{cases} \tag{33}$$

where the error term $e(k) = [e_1(k), e_2(k)]$ is a white noise sequence with a mean of 0 and a variance of 0.01. The input variable u_1 is a Gaussian random sequence with a mean of 30 and a variance of 1, and the input variable u_2 is a Gaussian random sequence with a mean of 26 and a variance of 1. It is obvious that there is a certain multicollinearity between the input variables.

In order to test the robust performance of the algorithms, the outliers of the sample X-direction and the sample Y-direction are artificially introduced. The sample X-direction outliers rates are 5%, 15%, 25%, 35%, and the sample Y-direction outliers rates are 0%, 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, 50%, a total of 55 data sets. The way of how the outliers is introduced is the same as what Section 5.2 detailed descripted. Moreover, in order to fully and visually demonstrate the superiority of the algorithm, the proposed GM-R-NNRW is compared with other several baseline algorithms, namely, the basic NNRW, the robust least squares support vector machine (R-LS-SVM) (Suykens, Brabanter, Lukas, & Vandewalle, 2002; Zhou, Guo, & Chai, 2018; Zhou, Guo, Wang et al., 2018) and the Cauchy-weighted M-estimation based NNRW (Cauchy-M-NNRW) (Zhou et al., 2017). The activation functions used for three neural network algorithms (NNRW, Cauchy-M-NNRW and GM-R-NNRW) are all the *Sigmoid* function. The numbers of hidden neurons for all the compared NNRWs are chosen to be 30, the corresponding input weights $\boldsymbol{w}_j$ and biases b_j are randomly assigned range within the same range $[-1, 1]$. Moreover, since these network parameters of the NNRW are randomly assigned, the results of each simulation are not unique. In order to better compare the performance of different algorithms, this paper conducts multiple simulation experiments for each data set and compares the robustness of different algorithms by calculating the modeling root mean square error (RMSE) of the testing data.

4.2. Modeling results

Fig. 2 is a comparison of the RMSE values of the 550 independent modeling experiments of the proposed GM-R-NNRW and the basic NNRW. Figs. 3–7 exhibit the corresponding RMSE boxplots for each algorithm when the sample X-direction outlier ratio is 0%, 5%, 15%, 25%, and 35%, while the sample Y-direction outliers are sequentially increased. It can be seen from Fig. 2 that the model built by the basic NNRW algorithm lacks robustness, cannot resist the interference of the outliers on the model, so that the prediction accuracy of the model is way worse than the proposed algorithm. The robust performance of the GM-R-NNRW algorithm proposed in this paper is greatly improved, it maintains an extremely low RMSE value under different X, Y-direction outlier ratios. The box diagram corresponding to each X, Y-direction outlier proportions shown in Figs. 3∼7 also reflects the above situation. At the same time, it can be seen that the R-LS-SVM essentially weights the sample according to the size of the model residual to improve the robustness of the model, cannot resist the interference of coexistence of outliers in X-direction and Y-direction. Therefore, when the samples in X-direction and Y-direction are abnormal, the model collapses and the prediction accuracy drop sharply. The Cauchy-M-NNRW algorithm in Zhou et al. (2017) also has this problem, and the M-estimation weights sample only by the residual direction. However, the proposed GM-R-NNRW based on the GM-estimation in this paper considers the influences of both the X-direction and the Y-direction outliers on the modeling process, and determines the sample weight according to the location of the hidden output vector and the standardized residual size. Therefore, even when the outliers with abnormality in X-direction and Y-direction coexists, the model is not easily affected by the outliers, much better robust performance can be obtained. From the point that the accuracy of the model obtained from GM-R-NNRW is not obvious changed with the increase of the outlier proportion in the X,

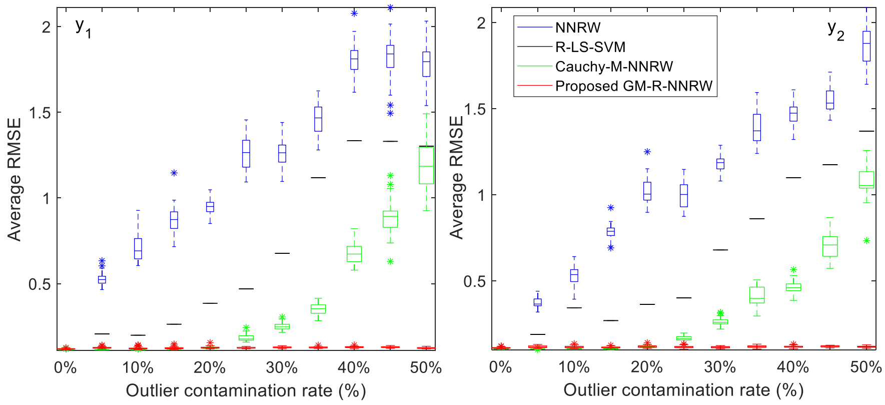


Fig. 3. Boxplot of testing RMSE for four algorithms at different Y-direction outlier proportions while X-direction outlier proportion is 0%.

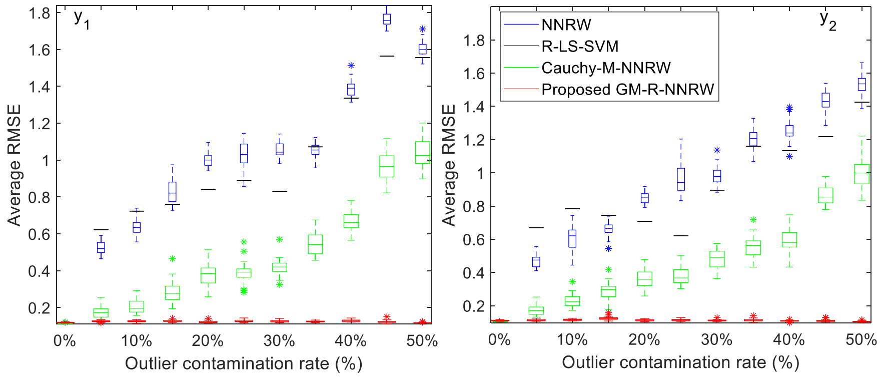


Fig. 4. Boxplot of testing RMSE for four algorithms at different Y-direction outlier proportions while X-direction outlier proportion is 5%.

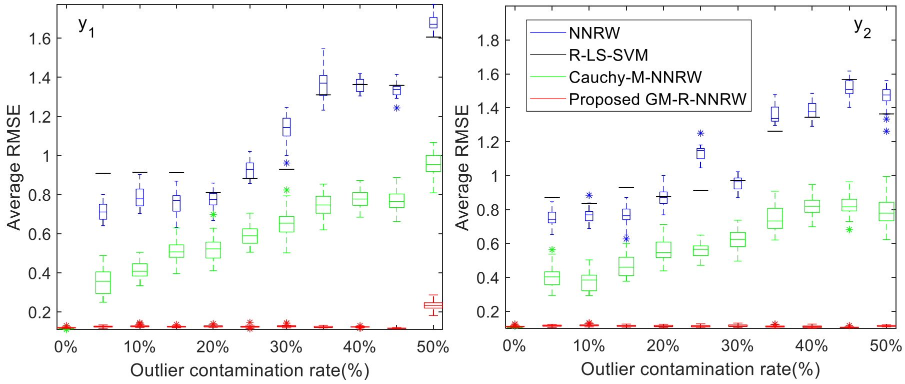


Fig. 5. Boxplot of testing RMSE for four algorithms at different Y-direction outlier proportions while X-direction outlier proportion is 15%.

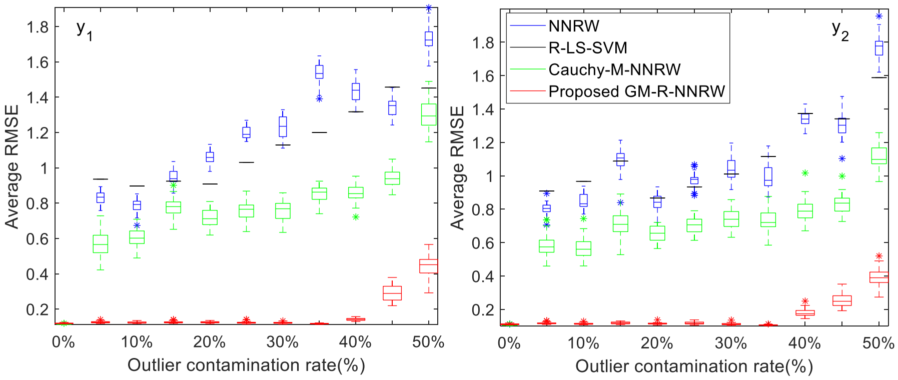


Fig. 6. Boxplot of testing RMSE for four algorithms at different Y-direction outlier proportions while X-direction outlier proportion is 25%.

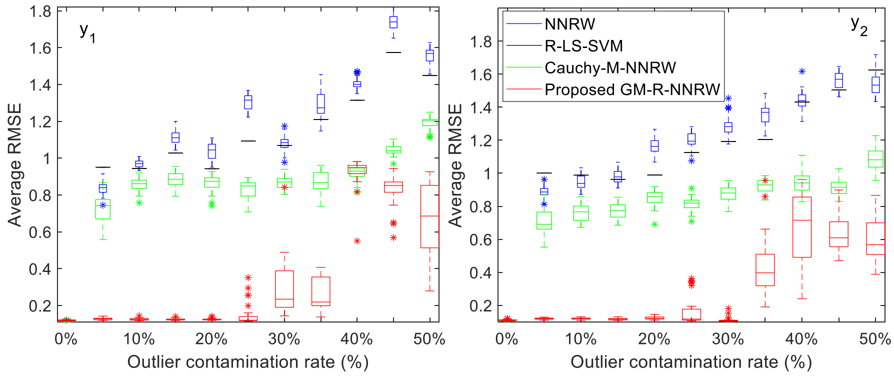


Fig. 7. Boxplot of testing RMSE for four algorithms at different Y-direction outlier proportions while X-direction outlier proportion is 35%.

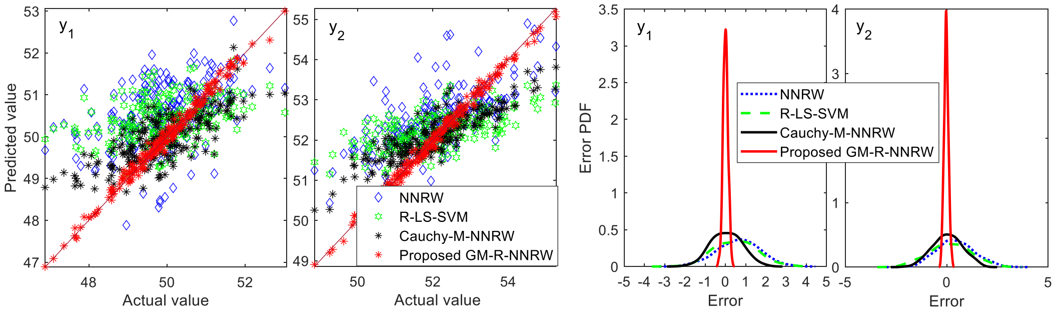


Fig. 8. Scatter diagrams (left) and prediction error PDF shapes (right) of approximation results with different modeling algorithms (X = 25%, Y = 25%).

Y-direction, the proposed algorithm can obtain the optimal modeling performance in any X- and Y- direction outlier proportions.

Further analyze the modeling performance of the proposed GM-R-NNRW algorithm, selecting the dataset with the sample X- direction and Y- direction outliers of 25% (X = 25%, Y = 25%) as the analysis object. Firstly, in the scatter diagram shown in the left subfigure of Fig. 8, the prediction samples from the proposed GM-R-NNRW algorithm are evenly distributed on both sides of the straight line $y = x$, which clearly shows that the prediction accuracy of the proposed algorithm is the highest, while the prediction samples from the other three compared algorithms deviate from the straight line to some extent, so there is a large prediction deviation. Then, the right subfigure of Fig. 8 shows the probability distribution function (PDF) curves of modeling error by the four modeling algorithms to the test data set. It can be clearly seen that the PDF curves corresponding to the proposed GM-R-NNRW are narrower and higher than that corresponding to the other algorithms. For the proposed algorithm, the overall of the PDF curves basically coincide with the vertical axis "0", and the degrees of deviation are the smallest, indicating that the best modeling performance among all the compared algorithms has been obtained by the proposed GM-R-NNRW algorithm.

Based on this function approximation experiment, it can be found that due to the proposed algorithm considers the influence of outliers both in the X- and Y- directions of the sample on the model, the collapse outlier proportion of the developed GM-R-NNRW model is up to 50%. This ensures that the proposed algorithm has much better generalization performance and robust performance than the other compared M-estimation based robust algorithms.

5. Application to quality modeling of blast furnace ironmaking (BFI) process

5.1. Process description and problem statement

Steel manufacturing is the pillar industry that affects the national economy and the people's livelihood (Zhou, Guo, & Chai, 2018; Zhou, Guo, Wang et al., 2018). As an important production link in the long process of steel manufacturing, the blast furnace ironmaking (BFI) as shown in Fig. 9 is to chemically reduce and physically convert the solid iron oxides into the liquid iron called '*hot metal*'. The whole BFI system includes: (1) a blast furnace body, (2) a raw materials feeding system, (3) a hot blast system, (4) a coal injection system, (5) a tapping system of hot metal and slag and (6) a gas treatment system. The raw materials including iron ore, fuel and solvent are loaded by the feeding system at upper part of furnace and move downward. The preheated air and fuel blown into the bottom of furnace through tuyere and generate a large amount of high temperature gas. The gas ascends to the furnace top and goes through a series of complicated physical and chemical reactions with the descending furnace charge. Thus the final hot metal and liquid slag are continuously formed (Li, Hua, Yang, & Guan, 2018; Zhou, Guo, & Chai, 2018; Zhou, Guo, Wang et al., 2018; Zhou et al., 2017).

The hot metal is the final product of the BFI process, and its temperature is as high as 1500°C. In actual ironmaking production, the hot metal quality (HMQ) mainly includes the molten iron temperature (MIT, °C), the molten iron sulfur content ([S], %), the molten iron phosphorus content ([P], %), and the molten iron silicon content ([Si], %). These four quality indicators not only reflect the BFI operation status but also reflect the energy consumption of the whole ironmaking production process. Therefore, in order to achieve the optimal operation of the BFI process, it is necessary to closely monitor the state of the hot metal quality. However, there are many process variables of the BFI process that affect the HMQ to different degrees, and the coupling between these influence parameters is quite serious. There is also a strong correlation between the process data collected on site or calculated from the other measurable variables, which greatly increases the difficulty in quality modeling of BFI process. Moreover, the BFI process is extremely complicated, the smelting environment and the operating conditions change greatly, which makes it is hardly possible to establish an applicable model to accurately describe the real-time state of the hot metal by the conventional methods (An, Shen, Wu, & She, 2019; Li et al., 2018; Zhou et al., 2017). At the same time, due to

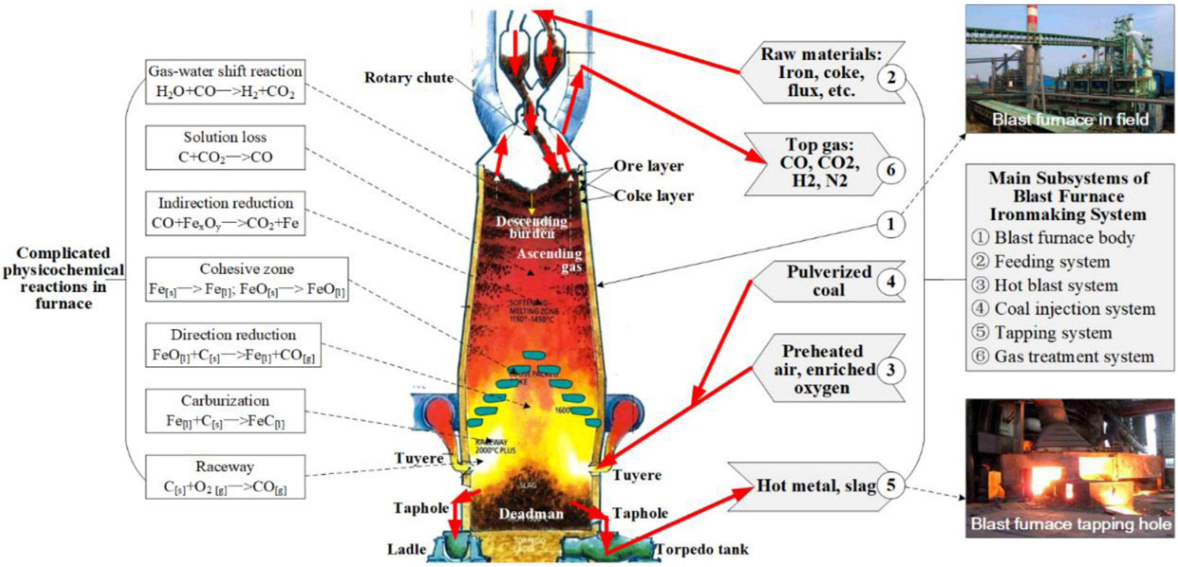


Fig. 9. Simplified diagram of a typical BFI production process.

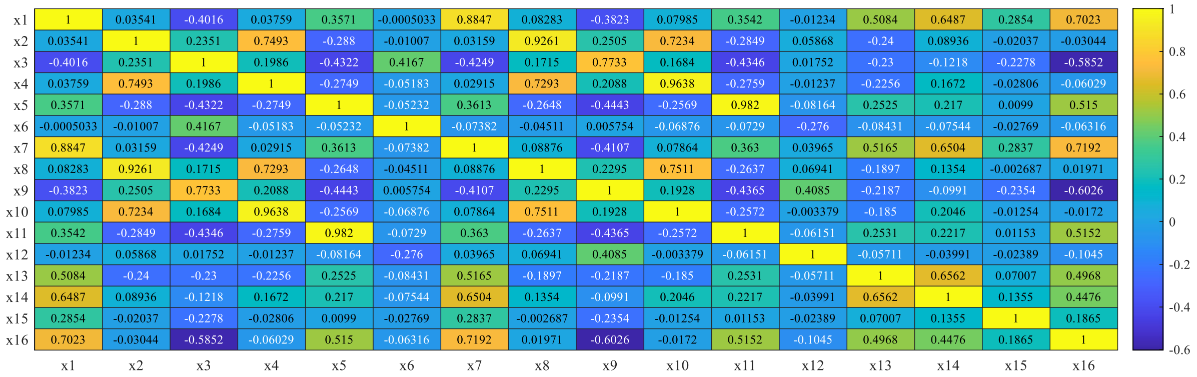
Table 2
The main process variables and their measurement or calculation methods in field (where h_c denotes the Hydrogen content in coal).

Variable	Unit	Instrumentation or calculation model
x_1-Flux of cold air	m^3/min	HH-WLB differential pressure flowmeter
x_2-Oxygen-rich flow	m^3/h	A+K balance flowmeter
x_3-Blast pressure	kPa	DPharp EJA high accuracy pressure transmitter
x_4-Furnace top pressure	kPa	DPharp EJA high accuracy pressure transmitter
x_5-Blast temperature	°C	Hongguang SBW temperature transmitter
x_6-Blast humidity	g/m^3	Air humidity sensor
x_7-Coal injection rate	t/h	Coal powder flowmeter
x_8-Blast ratio	1/min	$f_1(x_1)$
x_9-Gas permeability	m^3/min× kPa	$f_2(x_{1,}\ x_{3,}x_4)$
x_{10}-Bosh gas index	$m^3/(\text{min}\times m^2)$	$f_3(x_{15})$
x_{11}-Drag coefficient	–	$((10000x_3)^2 - 100x_4^2)/x_{15}^{1.7}$
x_{12}-Actual wind speed	m/s	$(27.66 + 0.101x_5)/(27.66 + 273x_3) \times (3.14x_1/300)$
x_{13}-Adiabatic fame temperature	°C	$1559 + 0.839x_5 + 4972x_2/x_1 - 6.033x_6 - 3.15x_7 \times 1000000/x_1$
x_{14}-Oxygen rich rate	vol%	$((97.8x_2 + ((21 + 0.036x_6)x_1))/(x_2 + x_1) - (12.6 + 0.02175x_6))$
x_{15}-Bosh gas volume	m^3/min	$(1.21x_1 + 2x_2 + 0.448x_6x_1 + 0.448x_6x_2 + 112x_7h_c)/60$
x_{16}-Blast kinetic energy	kJ/s	$(1.05x_1 + 0.67(x_1x_6 + x_2x_6)/(803.6 - x_6))/0.2x_{12}^2$

the failure of the detection equipment, aging and other human factors, resulting in numerous outliers in the actual BFI operational data, which mainly includes the input direction outliers in measurement of the process variables and the output direction outliers in offline laboratory analysis of the four quality indicators.

This paper conducts the quality modeling application for industrial BFI process based on actual production data of a large ironmaking plant. In this paper, 16 key BFI process variables as listed in Table 2 are selected as the modeling input variables, and above four HMQ indicators (that are the MIT, the [S], the [P], and [Si]) are used as the output variables. Table 2 also lists the measurement or calculation methods of these key process variables in actual industrial production. Fig. 10 is the correlation coefficient figure between each key process variable, showing the multicollinearity problem in the HMQ modeling which must be consider. Due to the complicated nonlinear dynamics of the BFI process, and the large time lag of process input–output response, the HMQ model is designed as a nonlinear autoregressive structure by the following Eq. (34). This means that the quality outputs at current time is not only related to the process inputs at the current moment, and it is also related to the inputs and quality outputs of the previous moment, capturing the nonlinear dynamics of the smelting process.

$$\boldsymbol{Y}(t) = f\,\{\boldsymbol{X}(t), \boldsymbol{X}(t-1), \boldsymbol{Y}(t-1)\} \tag{34}$$



	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
x1	1	0.03541	-0.4016	0.03759	0.3571	-0.0005033	0.8847	0.08283	-0.3823	0.07985	0.3542	-0.01234	0.5084	0.6487	0.2854	0.7023
x2	0.03541	1	0.2351	0.7493	-0.288	-0.01007	0.03159	0.9261	0.2505	0.7234	-0.2849	0.05868	-0.24	0.08936	-0.02037	-0.03044
x3	-0.4016	0.2351	1	0.1986	-0.4322	0.4167	-0.4249	0.1715	0.7733	0.1684	-0.4346	0.01752	-0.23	-0.1218	-0.2278	-0.5852
x4	0.03759	0.7493	0.1986	1	-0.2749	-0.05183	0.02915	0.7293	0.2088	0.9638	-0.2759	-0.01237	-0.2256	0.1672	-0.02806	-0.06029
x5	0.3571	-0.288	-0.4322	-0.2749	1	-0.05232	0.3613	-0.2648	-0.4443	-0.2569	0.982	-0.08164	0.2525	0.217	0.0099	0.515
x6	-0.0005033	-0.01007	0.4167	-0.05183	-0.05232	1	-0.07382	-0.04511	0.005754	-0.06876	-0.0729	-0.276	-0.08431	-0.07544	-0.02769	-0.06316
x7	0.8847	0.03159	-0.4249	0.02915	0.3613	-0.07382	1	0.08876	-0.4107	0.07864	0.363	0.03965	0.5165	0.6504	0.2837	0.7192
x8	0.08283	0.9261	0.1715	0.7293	-0.2648	-0.04511	0.08876	1	0.2295	0.7511	-0.2637	0.06941	-0.1897	0.1354	-0.002687	0.01971
x9	-0.3823	0.2505	0.7733	0.2088	-0.4443	0.005754	-0.4107	0.2295	1	0.1928	-0.4365	0.4085	-0.2187	-0.0991	-0.2354	-0.6026
x10	0.07985	0.7234	0.1684	0.9638	-0.2569	-0.06876	0.07864	0.7511	0.1928	1	-0.2572	-0.003379	-0.185	0.2046	-0.01254	-0.0172
x11	0.3542	-0.2849	-0.4346	-0.2759	0.982	-0.0729	0.363	-0.2637	-0.4365	-0.2572	1	-0.06151	0.2531	0.2217	0.01153	0.5152
x12	-0.01234	0.05868	0.01752	-0.01237	-0.08164	-0.276	0.03965	0.06941	0.4085	-0.003379	-0.06151	1	-0.05711	-0.03991	-0.02389	-0.1045
x13	0.5084	-0.24	-0.23	-0.2256	0.2525	-0.08431	0.5165	-0.1897	-0.2187	-0.185	0.2531	-0.05711	1	0.6562	0.07007	0.4968
x14	0.6487	0.08936	-0.1218	0.1672	0.217	-0.07544	0.6504	0.1354	-0.0991	0.2046	0.2217	-0.03991	0.6562	1	0.1355	0.4476
x15	0.2854	-0.02037	-0.2278	-0.02806	0.0099	-0.02769	0.2837	-0.002687	-0.2354	-0.01254	0.01153	-0.02389	0.07007	0.1355	1	0.1865
x16	0.7023	-0.03044	-0.5852	-0.06029	0.515	-0.06316	0.7192	0.01971	-0.6026	-0.0172	0.5152	-0.1045	0.4968	0.4476	0.1865	1

Fig. 10. Correlation coefficient between each key process variable.

5.2. Industrial data-based modeling data design

In order to comprehensively evaluate the robustness of the proposed GM-R-NNRW algorithm, two different types of data sets are designed based on the actual BFI operational data.

- The first type of data set is designed to be training set containing anomalies only in the Y-direction of the sample. In the case of limiting the maximum amplitude of the outliers to 2, the effect of different sample Y-direction outlier ratios on the prediction accuracy of the improved algorithm is compared. The specific implementation details are as follows: randomly select the sample points $y_{i,\text{Outlier}}$ with the interval of 5% and the range from 5% to 50%. Then the selected samples are processed by the following Eq. (35).

$$y_{i,\text{Outlier}} = y_i + 2 \times \text{sign} \times \left[\text{rand}(0,1) \times y_{\text{maxmin}}\right], i = 1,2,3,4 \tag{35}$$

where $y_{\text{maxmin}} = \max(\boldsymbol{y}_i) - \min(\boldsymbol{y}_i)$ represents the difference between the maximum and minimum values of each quality measurement data in the actual BFI process.

- The second type of data set is designed to be training set containing anomalies both in the X-direction and Y-direction of the sample. In the case of limiting the maximum amplitude of the outliers to 2, the effect of different sample X, Y-directions outlier ratios on the prediction accuracy of the improved algorithm is compared. The design of the Y-direction outliers is consistent with the design of the first type of dataset. The X-direction outliers are designed as follows: randomly select the sample points $x_{j,\text{Outlier}}$ with the interval of 10% and the range from 5% to 35%. Then the selected samples are processed by the following Eq. (36).

$$x_{j,\text{Outlier}} = x_j + 2 \times \text{sign} \times \left[\text{rand}(0,1) \times x_{\text{maxmin}}\right], j = 1,2,\ldots,16 \tag{36}$$

where $x_{\text{maxmin}} = \max(\boldsymbol{x}_j) - \min(\boldsymbol{x}_j)$ represents the difference between the maximum and minimum values of each process variable in the actual BFI data.

In Eqs. (35) and (36), when sign = 1, the introduced outlier is a positive outlier, while sign = −1, the introduced outlier is a negative outlier. Moreover, the ratio of positive and negative outliers in this modeling experiment is 2:1, and the test set uses the data set under normal conditions. In order to ensure the accuracy of the modeling test, 20 sets of independent modeling tests were performed for each set of data sets, and the modeling performance of the algorithm was comprehensively compared, and relevant analysis figures were obtained based on the relevant test data.

5.3. Quality modeling and prediction results

Fig. 11 is a comparison of the RMSE values of the model obtained from the 1100 modeling experiments of the proposed GM-R-NNRW and the basic NNRW. As the outlier rate increases in the X, Y-directions, the RMSE value of the HMQ model obtained by the basic NNRW increases continuously, which means that the accuracy of the model is getting worse and worse. While the HMQ model obtained by the proposed GM-R-NNRW does not show this situation, the extremely low RMSE value can be stably maintained at each outlier rate, and the model accuracy is accurate and stable.

Figs. 12–16 represent the test RMSE box plot with the sample Y-direction outliers increasing sequentially as the sample X-direction outlier ratio is 0%, 5%, 15%, 25%, and 35%, respectively. It can also be seen from these figure that the basic NNRW modeling algorithm lacks robustness and the prediction accuracy is always the worst. Compared with the basic NNRW, the robust performance of the R-LS-SVM and the Cauchy-M-NNRW have been improved, but when there are outliers in the X and Y-directions, the prediction accuracy of the model is greatly reduced. For example, in Fig. 14, as the sample Y-direction outliers are continuously increased, due to the coexistence of the outliers in the X and Y-directions, both the prediction accuracy of the models by the R-LS-SVM and the Cauchy-M-NNRW are significantly reduced. Although the latter is better than the former, it cannot maintain stable prediction accuracy at various outlier rates. It obverses that in these figures, only the proposed GM-R-NNRW algorithm can maintain a low RMSE value in each X, Y-direction outliers, resulting in stable model prediction accuracy and stable modeling performance.

Further analyze the modeling performance of the proposed GM-R-NNRW algorithm, selecting the dataset with the sample input outliers of 25% and output outliers of 35% (X = 25%, Y = 35%) as the analysis object. Fig. 17 depicts the HMQ prediction results of the four algorithms to the test data. It is obvious that the prediction curve obtained by the proposed GM-R-NNRW algorithm has the best HMQ prediction to the real data curve and can accurately predict the changes of the four HMQ curves. In the PDF curve of the residual shown in Fig. 18, the PDF curves of the four HMQ indicators obtain by the proposed GM-R-NNRW algorithm are not only thin and high, but also coincide with the "0" vertical axis as a whole, which means the deviation of HMQ prediction by the proposed GM-R-NNRW is the smallest. These curves above show that the proposed GM-R-NNRW has higher prediction accuracy than other compared modeling algorithms.

In order to more intuitively reflect the superiority of the proposed algorithm, the approximation performance of the four different algorithms was obtained using the evaluation indicators commonly used in regression modeling, and Tables 3 and 4 are obtained. For the first three evaluation indicators (RMSE, MAE, MAPE), the smaller values of these indicators are, the better prediction performance of the model is. As for the R square indicator, the closer the value is to 1, the better the ability to approximate the data, thus the better predictive performance of the model is. Through the comprehensive comparison of these indicators, it is obvious that the robustness and prediction accuracy of the proposed GM-R-NNRW algorithm is the highest among all the compared algorithms. Now, the HMQ soft-sensor software system with the proposed GM-R-NNRW algorithm has been developed by utilizing

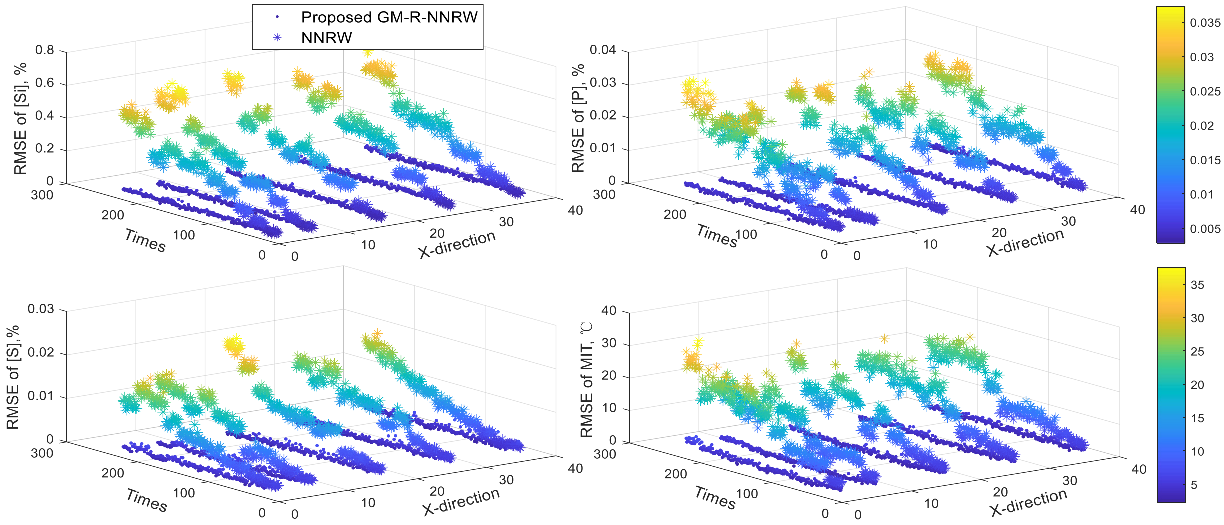


Fig. 11. HMQ testing RMSE comparison of NNRW and GM-R-NNRW under different X, Y-direction outlier proportions. (The Times axis corresponds to an outlier of 5% in the Y-direction for every 20 scales.)

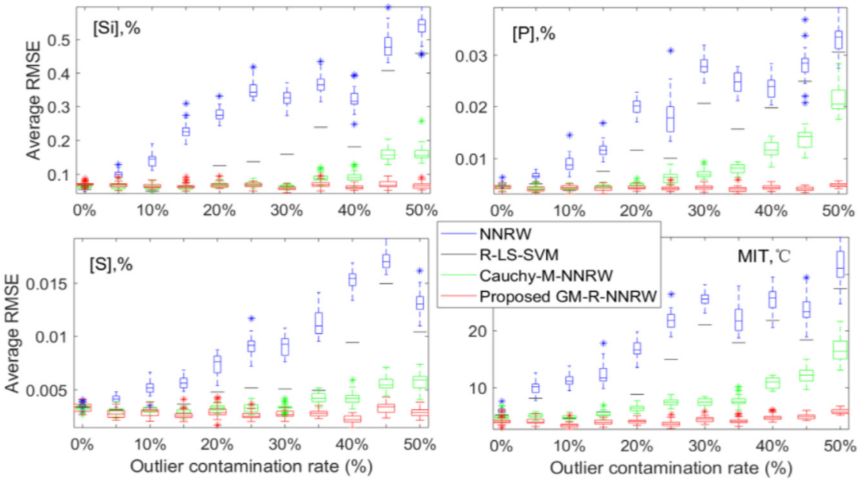


Fig. 12. Boxplot of HMQ testing RMSE for four modeling algorithms at different Y-direction outlier ratios while X-direction outlier ratio is 0%.

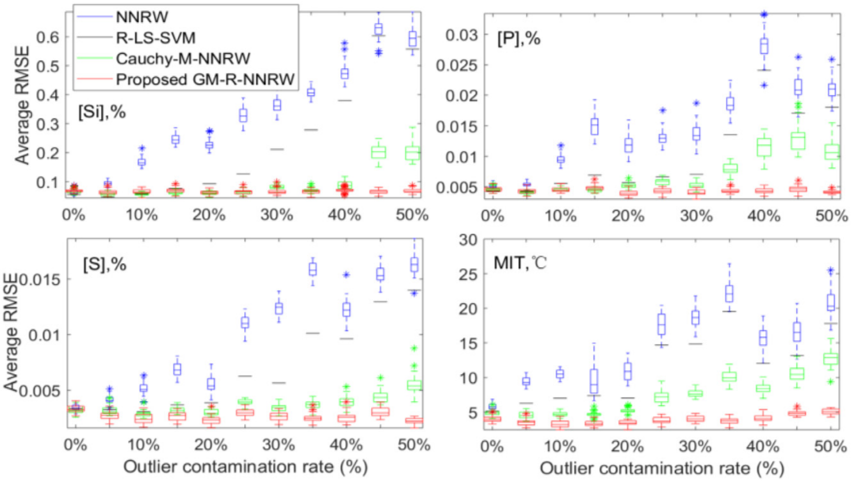


Fig. 13. Boxplot of HMQ testing RMSE for four modeling algorithms at different Y-direction outlier ratios while X-direction outlier ratio is 5%.

Table 3
Comparison of RMSE and MAE for different modeling algorithms (X = 25%, Y = 35%).

Algorithm	RMSE				Mean absolute error (MAE)			
	[Si], %	[P], %	[S], %	MIT, °C	[Si], %	[P], %	[S], %	MIT, °C
NNRW	0.3789	0.0294	0.0138	23.305	0.3429	0.0260	0.0124	18.180
R-LS-SVM	0.3002	0.0257	0.0110	22.669	0.2773	0.0240	0.0099	18.208
Cauchy-M-NNRW	0.0937	0.0111	0.0034	7.6914	0.0621	0.0085	0.0022	5.8310
GM-R-NNRW	**0.0649**	**0.0053**	**0.0025**	**3.7792**	**0.0463**	**0.0040**	**0.0017**	**2.6412**

Table 4
Comparison of MAPE and R square for different modeling algorithms (X = 25%, Y = 35%).

Algorithm	Mean absolute percentage error (MAPE)				R square			
	[Si], %	[P], %	[S], %	MIT, °C	[Si], %	[P], %	[S], %	MIT, °C
NNRW	0.7843	0.2111	0.4981	0.0121	−8.1128	−12.964	−2.7071	−2.3148
W-LS-SVM	0.6295	0.1946	0.3986	0.0122	−5.2516	−10.840	−1.5415	−2.4441
Cauchy-M-NNRW	0.1465	0.0698	0.0888	0.0039	0.5768	−0.9059	0.8228	0.6521
GM-R-NNRW	**0.1034**	**0.0322**	**0.0599**	**0.0018**	**0.5164**	**0.5906**	**0.8292**	**0.6317**

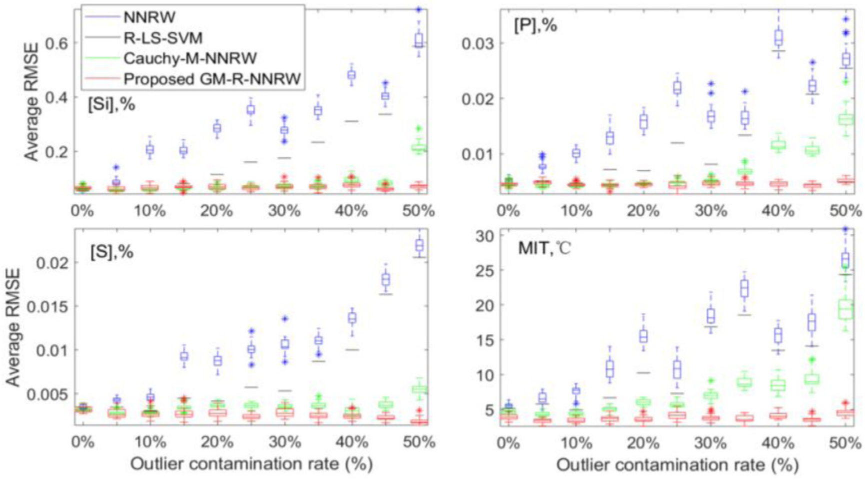


Fig. 14. Boxplot of HMQ testing RMSE for four modeling algorithms at different Y-direction outlier ratios while X-direction outlier ratio is 15%.

the C# language, and applied in the #2 blast furnace of a large iron & steel plant Southern China, which is shown in Fig. 19. The soft measurement results of this soft-sensor system can provide important metal quality information for the process monitoring system to guide and assist the operators to make daily decision-making operations on BFI production.

6. Conclusion

In this paper, a novel GM-R-NNRW algorithm based on generalized-M estimation and PLS is proposed for imperfect data modeling of complicated industrial processes. The main excellent features of the proposed algorithm can be summarized into the following aspects:

- For eliminating the impact of X-direction (input) and Y-direction (output) outliers coexist in the complicated industrial process data, the proposed GM-R-NNRW uses the improved generalized M-estimation to simultaneously consider the influence of outliers in the X, Y-direction. The sample weights are respectively determined according to the position of the hidden layer output vector and the standardized residual size. Even when the outliers with abnormality in X-direction and Y-direction coexists, the GM-R-NNRW can be unaffected by them, which greatly improves the robust performance of the model.
- For solving the multicollinearity problem existing in complicated industrial process data, the GM-R-NNRW obtains the hidden output weight by the improved PLS method, that is, extracts the principal element with the largest variance change from the input variable set and the output variable set respectively. Under the condition that both the orthogonality and the normalized constraint are satisfied, the covariance between the principal elements of the input and output variable sets is guaranteed to be the largest, thus effectively solving the adverse effects of multicollinearity modeling.
- Due the above two features, the proposed algorithm can be easily implemented and applied in engineering practice. The application of the GM-R-NNRW does not need too much extra data

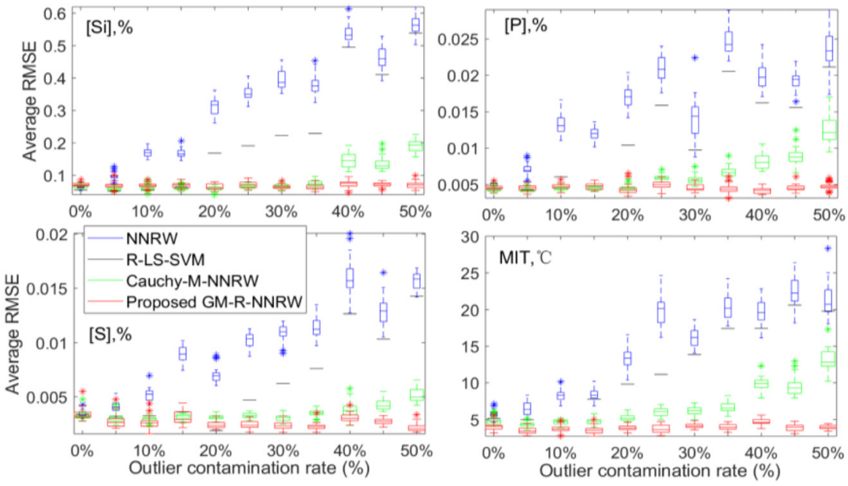


Fig. 15. Boxplot of HMQ testing RMSE for four modeling algorithms at different Y-direction outlier ratios while X-direction outlier ratio is 25%.

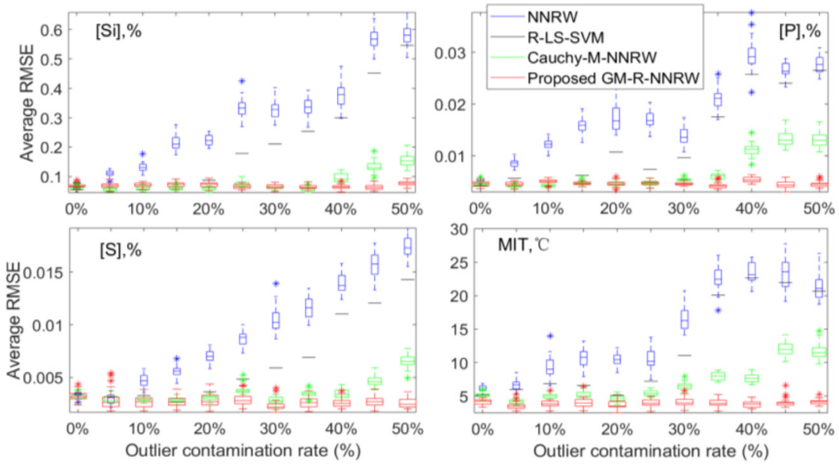


Fig. 16. Boxplot of HMQ testing RMSE for four modeling algorithms at different Y-direction outlier ratios while X-direction outlier ratio is 35%.

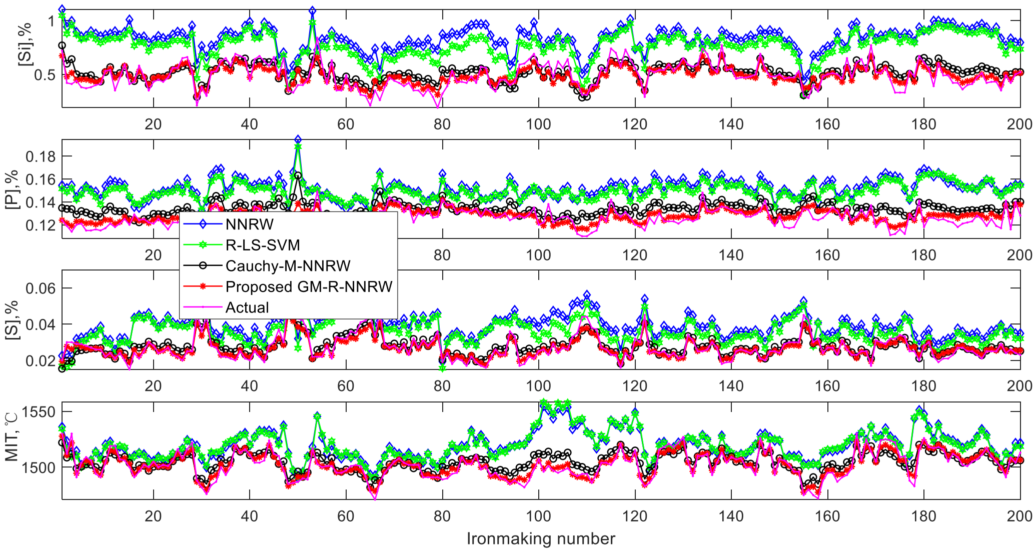


Fig. 17. HMQ estimation results of four modeling algorithms (X = 25%, Y = 35%).

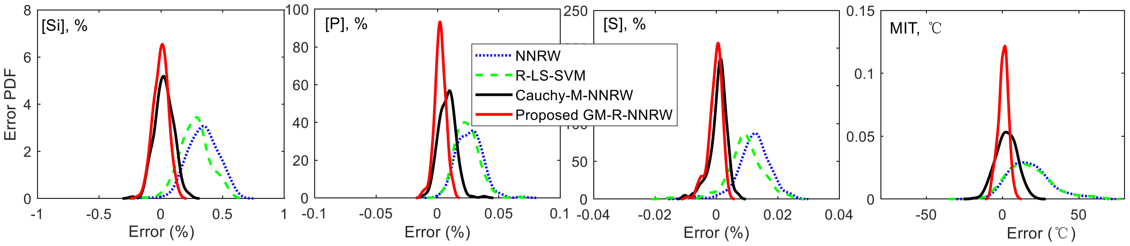


Fig. 18. Error PDF shapes of HMQ estimation with different algorithms (X = 25%, Y = 35%).

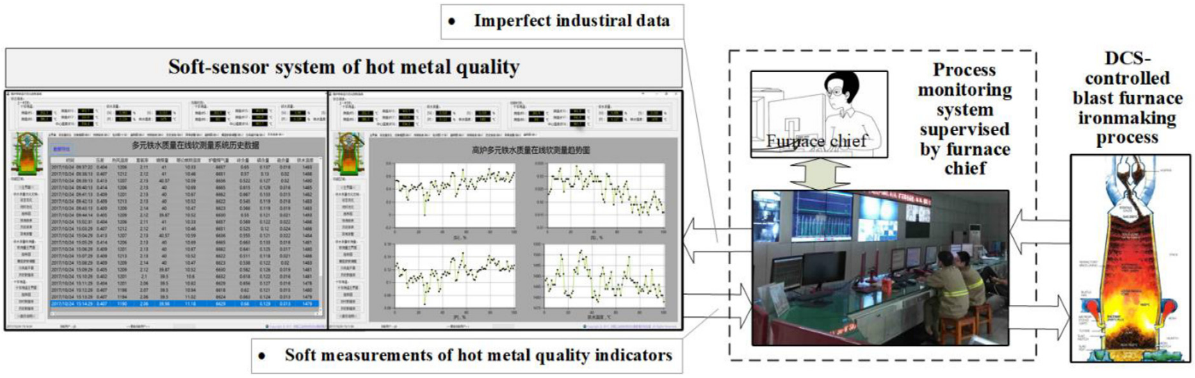

Imperfect industiral data
Soft-sensor system of hot metal quality
多元铁水质量在线软测量系统历史数据
高炉多元铁水质量在线软测量趋势图
Furnace chief
Process monitoring system supervised by furnace chief
DCS-controlled blast furnace ironmaking process
Soft measurements of hot metal quality indicators

Fig. 19. HMQ soft-sensor software system with the proposed GM-R-NNRW algorithm for guiding and assisting the operators in field to make daily decision-making operations of BFI production.

preprocessing, such as data dimensionality reduction, data outlier manual elimination, etc. As long as the actual industrial data are matched according to the input and output, the proposed method can be directly used in practical modeling and prediction applications with these original data, and outputs the results quickly and accurately.

Based on the function approximation experiment and the hot metal quality modeling application of BFI process, it is confirmed that the proposed GM-R-NNRW can achieve accurate modeling with excellent robust performance. Therefore, the proposed robust modeling method has high application potential for various complicated industrial processes.

Declaration of competing interest

None declared.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (61890934, 61790572), the Liaoning Revitalization Talents Program (XLYC1907132), and the Fundamental Research Funds for the Central Universities (N180802003). The work was done when Hong Wang was with the University of Manchester, UK.

References

Aguilera, A. M., Escabias, M., & Valderrama, M. J. (2006). Using principal components for estimating logistic regression with high-dimensional multicollinear data. *Computational Statistics & Data Analysis*, *50*(8), 1905–1924, Available from: http://dx.doi.org/10.1016/j.csda.2005.03.011.

An, J. Q., Shen, X. L., Wu, M., & She, J. H. (2019). A multi-time-scale fusion prediction model for the gas utilization rate in a blast furnace. *Control Engineering Practice*, *92*, Article 104120, Available from: http://dx.doi.org/10.1016/j.conengprac.2019.104120.

Bastien, P., Vinzi, V. E., & Tenenhaus, M. (2005). PLS generalized linear regression. *Computational Statistics & Data Analysis*, *48*(1), 17–46, Available from: http://dx.doi.org/10.1016/j.csda.2004.02.005.

Browne, A., & Sun, R. (2002). Connectionist inference models. *Neural Networks*, *14*(10), 1331–1355, Available from: http://dx.doi.org/10.1016/S0893-6080(01)00109-5.

Chai, T. Y. (2009). The challenge of control and optimization theory method for production and manufacturing process optimization control. *Acta Automatica Sinica*, *35*(6), 641–649, Available from: http://dx.doi.org/10.3724/SP.J.1004.2009.00641.

Chen, G. J., & Ge, Z. Q. (2020). Robust Bayesian networks for low-quality data modeling and process monitoring applications. *Control Engineering Practice*, *97*, Article 104344, Available from: http://dx.doi.org/10.1016/j.conengprac.2020.104344.

Daszykowski, M., Heyden, Y. V., & Walczak, B. (2007). Robust partial least squares model for prediction of green tea antioxidant capacity from chromatograms. *Journal of Chromatography A*, *1176*(1–2), 12–18, Available from: http://dx.doi.org/10.1016/j.chroma.2007.10.100.

Filzmoser, P., & Todorov, V. (2013). Robust tools for the imperfect world. *Information Sciences*, *245*(10), 4–20, Available from: http://dx.doi.org/10.1016/j.ins.2012.10.017.

Frahm, G., Nordhausen, K., & Oja, H. (2020). M-estimation with incomplete and dependent multivariate data. *Journal of Multivariate Analysis*, *176*, Article 104569, Available from: http://dx.doi.org/10.1016/j.jmva.2019.104569.

Gou, Z., & Fyfe, C. (2004). A canonical correlation neural network for multicollinearity and functional data. *Neural Networks*, *17*(2), 285–293, Available from: http://dx.doi.org/10.1016/j.neunet.2003.07.002.

Huber, P. J., & Ronchetti, E. M. (2009). *Robust statistics* (2nd ed.). New York: Wiley, Available from: http://dx.doi.org/10.1002/9780470434697.

Igelnik, B., & Pao, Y. H. (1995). Stochastic choice of basic functions in adaptive function approximation and the functional-link net. *IEEE Transactions on Neural Networks*, *6*(6), 1320–1329, Available from: http://dx.doi.org/10.1109/72.471375.

Katrutsa, A., & Strijov, V. (2017). Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria. *Expert Systems with Application*, *76*(15), 1–11, Available from: http://dx.doi.org/10.1016/j.eswa.2017.01.048.

Li, J. P., Hua, C. H., Yang, Y. N., & Guan, X. P. (2018). Bayesian Block structure sparse based T-S fuzzy modeling for dynamic prediction of hot metal silicon content in the blast furnace. *IEEE Transactions on Industrial Electronics*, *65*(6), 4933–4942, Available from: http://dx.doi.org/10.1109/TIE.2017.2772141.

Muller, C. J., & Craig, I. K. (2015). Modelling of a dual circuit induced draft cooling water system for control and optimisation purposes. *Journal of Process Control*, *25*, 105–114, Available from: http://dx.doi.org/10.1016/j.jprocont.2014.11.010.

Pao, Y. H., & Takefuji, Y. (1992). Functional-link net computing: theory, system architecture, and functionalities. *Computer*, *25*(5), 76–79, Available from: http://dx.doi.org/10.1109/2.144401.

Pitselis, G. (2013). A review on robust estimators applied to regression credibility. *Journal of Computational and Applied Mathematics*, *239*, 231–249, Available from: http://dx.doi.org/10.1016/j.cam.2012.09.009.

Rousseeuw, P. J., & Croux, C. (1993). Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, *88*(424), 1273–1283, Available from: http://dx.doi.org/10.2307/2291267.

Schmidt, W. F., Kraaijveld, M. A., & Duin, R. P. W. (1992). Feedforward neural networks with random weights. In *Pattern recognition, 1992. II. Conference B: Pattern recognition methodology and systems, proceedings. 11th IAPR international conference on*. IEEE Computer Society, Available from: http://dx.doi.org/10.1109/ICPR.1992.201708.

Serneels, S., Croux, C., Filzmoser, P., & VanEspen, P. J. (2005). Partial robust m-regression. *Chemometrics & Intelligent Laboratory Systems*, *79*(1), 55–64, Available from: http://dx.doi.org/10.1016/j.chemolab.2005.04.007.

Suykens, J. A. K., Brabanter, J. D., Lukas, L., & Vandewalle, J. (2002). Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, *48*(1), 85–105, Available from: http://dx.doi.org/10.1016/S0925-2312(01)00644-0.

Wakelinc, I. N., & Macfie, H. J. H. (1992). A robust PLS procedure. *Journal of Geographical Sciences*, *6*(4), 189–198, Available from: http://dx.doi.org/10.1002/cem.1180060404.

Westerhuis, J. A., Kourti, T., & Macgregor, J. F. (1998). Analysis of multiblock and hierarchical PCA and PLS models. *Journal of Chemometrics*, *12*(5), 301–321, Available from: http://dx.doi.org/10.1002/(SICI)1099-128X(199809/10)12:53.0.CO;2-S.

Wold, H. (1973). Nonlinear iterative partial least squares (NIPALS) modelling: Some current developments. *Multivariate Analysis–III*, *1973*, 383–407, Available from: http://dx.doi.org/10.1016/B978-0-12-426653-7.50032-6.

Yu, W., & Zhao, C. (2020). Robust monitoring and fault isolation of nonlinear industrial processes using denoising autoencoder and elastic net. *IEEE Transactions on Control Systems Technology*, *28*(3), 1083–1091, Available from: http://dx.doi.org/10.1109/TCST.2019.2897946.

Zhang, J., Tang, Z. H., Xie, Y. F., Chen, Q., An, M. X., & Gui, W. H. (2020). Timed key–value memory network for flotation reagent control. *Control Engineering Practice*, *98*, Article 104360, Available from: http://dx.doi.org/10.1109/TCST.2019.2897946.

Zhao, C., & Sun, Y. (2013). Subspace decomposition approach of fault deviations and its application to fault reconstruction. *Control Engineering Practice*, *21*(10), 1396–1409, Available from: http://dx.doi.org/10.1016/j.conengprac.2013.06.008.

Zhao, C., Wang, F., & Zhang, Y. (2009). Nonlinear process monitoring based on kernel dissimilarity analysis. *Control Engineering Practice*, *17*(1), 221–230, Available from: http://dx.doi.org/10.1016/j.conengprac.2008.07.001.

Zhou, P., Chai, T. Y., & Wang, H. (2009). Intelligent optimal-setting control for grinding circuits of mineral processing process. *IEEE Transactions on Automation Sciences and Engineering*, *6*(4), 730–743, Available from: http://dx.doi.org/10.1109/TCST.2019.2897946.

Zhou, P., Guo, D. W., & Chai, T. Y. (2018). Data-driven predictive control of molten iron quality in blast furnace ironmaking using multi-output LS-SVR based inverse system identification. *Neurocomputing*, *308*, 101–110, Available from: http://dx.doi.org/10.1016/j.neucom.2018.04.060.

Zhou, P., Guo, D. W., Wang, H., & Chai, T. Y. (2018). Data-driven robust M-LS-SVR-based NARX modeling for estimation and control of molten iron quality indices in blast furnace ironmaking. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(9), 4007–4021, Available from: http://dx.doi.org/10.1109/TNNLS.2017.2749412.

Zhou, P., Lv, Y. B., Wang, H., & Chai, T. Y. (2017). Data-driven robust RVFLNs modeling of blast furnace ironmaking process using Cauchy distribution weighted M-estimation. *IEEE Transactions on Industrial Electronics*, *64*(9), 7141–7151, Available from: http://dx.doi.org/10.1109/TIE.2017.2686369.