A GAN–SVR Prediction Method of the Metal Tube-Bending Rebound with Small Samples

Pengfei Zhang, Ziluo Fang, Liangyou Li, and Tingting Yang

1School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China
2Zhejiang Heliang Intelligent Equipment Co. Ltd., Huzhou, China
3Huzhou Key Laboratory of Intelligent Sensing and Optimal Control for Industrial Systems, School of Engineering, Huzhou University, Huzhou 313000, China
4School of Engineering, Huzhou University, Huzhou, Zhejiang, China

Correspondence should be addressed to Tingting Yang; 02820@zjhu.edu.cn

Received 10 July 2023; Revised 1 November 2023; Accepted 13 November 2023; Published 7 December 2023

Academic Editor: Giovanni Diraco

Copyright © 2023 Pengfei Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper investigates a predictive algorithm for the angle of the metal tube-bending rebound with small samples. First, the generative adversarial network (GAN) approach is introduced to address the issues of insufficient sample data. The proposed method can realize data augmentation through a generator, enhancing training effectiveness compared to conventional model-based and experimental prediction methods. To further reduce the problems caused by the small samples, the Wasserstein distance is utilized as the optimization objective for the GAN approach. Second, after obtaining the augmented dataset, Support Vector Regression (SVR) is employed to predict the rebound model of the metal tube-bending. A novel predictive algorithm for the angle of the metal tube-bending rebound based on GAN–SVR is proposed. It exhibits that the GAN–SVR owns more positive prediction ability and error when dealing with small samples than conventional GAN–radial basis function method (GAN–BP) and GAN–convolutional neural networks. Finally, the effectiveness of the proposed method is validated through experimental results.

1. Introduction

Metal tube fittings are widely used in various fields, such as automotive, aerospace, aviation, shipbuilding, machine tools, chemical, robotics, etc. Different industries employ tubes made of other materials for oil transportation, gas transmission, and fluid delivery. Moreover, in various types of engines, tubes occupy a more significant position [1–3]. From the perspective of materials and structures, metal tube components have characteristics such as good quality and high strength. Cutting, casting, welding, and plastic deformation processes are often applied to the tubes to give tubes specific shapes, lengths, bending angles, and bending radii. Plastic deformation processing has significant advantages regarding material utilization and dimensional accuracy. According to the forming conditions and load requirements, most bending machines used in enterprises perform cold bending on tubes. During the bending process, issues such as tube cracking, wrinkling, cross-section distortion, and rebound may occur [4]. Measures can be taken to prevent or mitigate bending defects, such as improving the installation position, using bending machines with assistive devices, adjusting the mandrel position, and enhancing mold structures. Among these defects, rebound in metal tubes is a critical issue in plastic deformation. It refers to the changes in angles and curvatures after removing the applied tool, leading to problems such as increased tolerance limits, assembly variations, and product durability [5–7]. When the product accuracy requirements are high, the deviation caused by tube-bending rebound may be intolerable; by predicting the amount of rebound, the initial bending Angle can be adjusted to obtain the final size that meets the design requirements [8]. The error caused by the rebound of the bending pipe causes the bending pipe to need multiple machining and adjustment, which wastes a lot of time, manpower, and materials [9, 10]. By accurately predicting the amount of rebound, unnecessary rework and waste generation can be avoided, especially when the material is expensive, which can greatly reduce the cost. To summarize,
rebound has always been a key factor inhibiting bending quality, increasing mold and product costs, and reducing manufacturing efficiency. Therefore, rebound prediction has received increasing attention and research [11–14].

In the actual production process of enterprises, various methods, such as analysis, modeling, and numerical techniques, are commonly employed to address the rebound issue in tube-bending [5, 6]. Al-Qureshi [8] and Al-Qureshi and Russo [9] consider the beam bending theory and derive analytical formulas based on the ideal elastic–plastic theory regarding tube diameter and symmetric bending cross-section. Lou and Stelson [10] and Lou and Stelson [11] further conduct theoretical and experimental studies on rotary stretch bending of tubes. The phenomenon is observed that there exists a linear relationship between the rebound and bending angles when the bending angles are large. Zhan et al. [12] employ numerical analysis to rapidly calculate the rebound angle of thin-walled tubes. By obtaining the physical properties of the tubes and using the mathematical model and physical model to analyze the rebound problem of the tube-bending, the above literature put forward the prediction method for the rebound problem of the tube-bending and realized the estimation of the angle of rebound of the tubes. However, these parameters-based rebound prediction approaches need numerous factors that are difficult to obtain in practice. Moreover, substantial variations among different materials increase the difficulty of achieving accurate predictions. Therefore, further exploration is necessary to realize rapid and precise prediction and compensation of bending rebound.

Data training methods have shown outstanding performance in various fields, such as traffic prediction, risk assessment, and natural language processing. Data training methods are also an exploratory approach in the tube-bending rebound. Inamdar et al. [13] utilized backpropagation neural networks to predict and control the rebound of V-shaped tubes. Ma et al. [14] employed machine learning (ML) to predict and compensate for tube-bending. The above kinds of literature are based on data training, using different ML methods to predict the rebound, and have achieved good results under sufficient samples. Data training methods require fewer parameters and offer high prediction accuracy. However, data training methods such as backpropagation neural networks, convolutional neural networks (CNN), and ML methods require more sample data. Overfitting may occur when dealing with small sample data training, leading to low model prediction accuracy and poor performance. A feasible alternative method must be explored for small samples of the bending rebound data.

Generative adversarial network (GAN) has shown excellent performance in image processing, text generation, and data augmentation. Data enhancement refers to generating more decadent training samples through a series of transformations and expansion operations on the original data to increase the model’s generalization ability [15–18]. Zhang et al. [15] utilize conditional Wasserstein GAN to augment and enhance 1D emotion recognition data, resulting in the improved classification of different emotions. Bhat and Hortal [16] employ Wasserstein GAN (WGAN) to augment and expand the benchmark dataset DEAP for evaluating emotion recognition algorithms, which was then merged with the original dataset to create an augmented dataset for medical diagnosis. Zheng et al. [17] utilize GAN–CNN to augment and estimate groundwater model parameters, accurately characterizing aquifer parameters and significantly improving computational efficiency. Yang et al. [18] employ GAN–2D-CNN to augment small samples bearing fault data, extracted features, and classified bearing fault types. This classification method exhibits high accuracy in fault classification. The above literature applies GAN to data enhancement and classification scenarios and puts forward a GAN model for reference. However, there has been no previous research utilizing GANs for related studies in the field of tube-bending rebound. Therefore, this paper proposes using GAN to augment small samples of bending rebound datasets and introduces Wasserstein distance to address the gradient vanishing problem, which may occur when GAN processes low-dimensional data. The Wasserstein distance is a measure of the difference between two probability distributions. Generally, we can think of two probability distributions as functions representing mass or density. Wasserstein distance considers the best scenario for transferring mass from one distribution to another and calculates the minimum effort required in the process [19, 20].

Support vector regression (SVR) has excellently addressed small sample problems. Despite the augmentation of bending rebound small sample data using GAN, the data volume still may not reach the scale required by neural networks. Therefore, SVR can handle the bending rebound data prediction problem. SVR is a nonlinear regression method that is based on the support vector machine (SVM) algorithm for modeling and prediction. SVR aims to minimize the error between the predicted and true values by finding an optimal hyperplane in the feature space [21, 22]. SVR has advantages over neural networks in processing small samples of data [23]. Dai et al. [21] utilize GAN–SVR to perform industrial quality prediction using limited labeled training data. However, no previous research has employed GAN–SVR to tackle regression problems with small sample data. Therefore, this paper proposes using GAN–SVR to address prediction and compensation issues in the field of tube-bending rebound.

This paper utilizes GAN–SVR for data augmentation and prediction of bending rebound. The proposed GAN–SVR algorithm for bending rebound prediction in this paper offers the following innovations:

1. To solve the problem of insufficient data on metal tube-bending samples, this paper presents the countermeasures of data enhancement. It introduces Wasserstein distance as the optimization objective, effectively alleviating the gradient disappearance problem.

Compared with the literature [5–12], this paper adopts a data training approach, which significantly reduces the number of experiments, product losses, and the need for sample parameters, and the issue of insufficient sample data is also addressed. Moreover, to tackle the problem of small sample capacity in tube-bending rebound data, this paper utilizes
Wasserstein distance as the optimization objective, effectively solving the issue of gradient vanishing during the training process.

(2) Given the small sample size and strong nonlinearity characteristics of metal tube-bending data, this paper presents an SVR method with radial basis function (RBF) as the kernel function to predict the metal tube-bending pipe rebound.

Compared with the literature [15–18], the paper utilizes SVR to fit the augmented data generated by GAN for model prediction. When facing lower data dimensions and smaller capacity, GAN–SVR exhibits better fitting performance than conventional GAN–CNN methods, and the trained model also demonstrates a lower mean squared error.

The rest of this paper is arranged as follows: Section 2 introduces the basic structure of GAN and presents the problem statement. Section 3 introduces Wasserstein’s GAN and the generative and discriminative networks designed for data augmentation and proposes a tube-bending rebound data augmentation algorithm based on Wasserstein’s GAN. Section 4 introduces SVR and proposes a tube-bending rebound prediction algorithm based on GAN–SVR. Section 5 implements the algorithm and compares it with other prediction methods.

2. Problem Statement and Preliminaries

This section introduces the basic structure of GAN and this paper’s objective. The core idea of GAN is inspired by zero-sum games consisting of a generator and a discriminator [24]. The former generates sample data, while the latter continually improves accuracy by discriminating between real and fake data. The generator transforms random noise into counterfeit data sample data, attempting to make their probability distribution as close as possible to that of real data. Although there are differences between the data output by the generator and the original data, this processing can expand and enhance the original data. The discriminator is a binary classifier discriminating between real and fake sample data. As training progresses, the generator gradually learns how to generate more realistic sample data while the discriminator becomes increasingly accurate.

When the data generated by the generator is close enough to real data that the discriminator can no longer accurately identify its source, it indicates that GAN has reached an equilibrium state. At this point, the sample data generated by the generator is increasingly close to the probability distribution of real data, and GAN has achieved its goal of sample data generation. The basic structure of GAN is illustrated in Figure 1.

During training, generator $G$ receives a random variable $z$ and generates fake sample data $G(z)$. The generator aims to make the generated sample data as close to real sample data as possible. The input of the discriminator $D$ consists of two parts, namely the real data $x$ and the data generated by the generator $G(x)$. Its output is usually a probability value representing the probability that $D$ considers the input from the actual distribution. If the input comes from real data, the output is 1; otherwise, it is 0. At the same time, the output of the discriminator is fed back to $G$ to guide its training. Finally, in an ideal scenario, $D$ cannot distinguish whether the input data comes from real data $x$ or generated data. At this point, the model reaches an optimal state.

2.1. Objective. The problems addressed in this paper are as follows:

(1) The tube-bending rebound dataset with small samples is expanded.

(2) The angle of the tube-bending rebound is predicted with data augmentation.

3. Data Augmentation with Wasserstein’s GAN

This section uses the Wasserstein distance instead of the traditional Jensen-Shannon (JS) divergence for dealing with
tube-bending rebound data, which can accurately measure the distance between real and generated samples. The problem of vanishing gradients with low-dimensional data is solved with low-dimensional data. Then, CNN are regarded as the foundational framework in this paper. The establishment process of the layers in the generator and discriminator networks is discussed, and the parameters for each layer are determined. Finally, a data augmentation algorithm is proposed that the amount of tube-bending data can be doubled based on the original dataset.

3.1. Wasserstein’s GAN. To further reduce the problems caused by the small samples, the Wasserstein distance is utilized as the optimization objective for the GAN approach. Let the test sample set contain $i$ data points, and the corresponding measured values be denoted as $x_i$. The relationship between the measured values and the data is represented by $f_i(x)$. Assuming there is a set of noise vectors $z$, following a joint Gaussian distribution $f_z(z)$. By utilizing a deep neural network, we can establish the mapping relationship between $f_i(x)$ and $f_z(z)$ to enable the generation of new data that conforms to the original data distribution by sampling from the known distribution. This mapping is established through training a GAN, which consists of two components: a generator $G(z; \theta^{(G)})$ and a discriminator $D(z; \theta^{(D)})$, where $\theta^{(G)}$ and $\theta^{(D)}$ represent the weights of the respective networks.

During the training process, the generator takes noise vector $z$ as input and gradually learns to approximate the distribution $f_z(z)$ pattern of the sample data $f_i(x)$ through up-sampling steps in a multilayer neural network. Simultaneously, the discriminator is trained, receiving input from the generator-produced and real sample data. Through down-sampling steps opposite to the generator, the discriminator ultimately outputs the probability $p$ of whether the input data is real sample data. The loss functions for the generator and discriminator are as follows:

$$L_G = -E_{z \sim f_z(z)}[D(G(z))],$$
$$L_D = -E_{x \sim f_i(x)}[D(x)] + E_{z \sim f_z(z)}[D(G(z))],$$

where $E$ denotes the desired distribution, $G(z)$ represents the generated data by the generator, and $D(\sim)$ signifies the output of the discriminator network, the training process of the GAN inherently involves a zero-sum game. The objective function of this game is defined as follows:

$$\min_D \max_G V(G, D) = E_{x \sim f_i(x)}[D(x)] - E_{z \sim f_z(z)}[D(G(z))].$$

The above objective indicates that the generator aims to generate data that closely follows the distribution pattern of real data, making it difficult for the discriminator to distinguish between the generated and real data. Upon completion of training, the generator will autonomously learn the distribution pattern of real data without supervision.

The optimization objective in Equation (3) is measured using the Wasserstein distance, which is preferred over the traditional JS distance. Using the Wasserstein distance can alleviate the issue of gradient vanishing during the training process and enhance training stability [19]. The Wasserstein distance is defined as follows:

$$W(p_r, p_g) = \inf_{\gamma \sim \Pi} E_{(x,y) \sim \gamma}[|x-y|],$$

where $\Pi(p_r, p_g)$ is a collection of joint probability distributions of $\gamma$ with $p_r$ and $p_g$ as marginal distributions. The variable $W(p_r, p_g)$ is the infimum of the expected value $\gamma(p_r, p_g)$, representing the distance between $x$ and $y$ required to fit $p_g$ onto $p_r$. Due to the computational difficulty in directly calculating the Wasserstein distance between arbitrary distributions, the Kantorovich–Rubinstein duality form is employed [19].

$$W(p_r, p_g) = \frac{1}{K} \sup_{||f||_L \leq K} E_{x \sim p_r}[f(x)] - E_{x \sim p_g}[f(x)],$$

where $||f||_L \leq K$ indicates that the function $f(x)$ satisfies $K$-Lipschitz continuity, with its absolute derivative value bounded. We can obtain the optimization objective under the Wasserstein distance by substituting $f_{0_r}(x)$ for $D(x)$ and $f_{0_g}(x)$ for $G(x)$ in the original objective function.

Remark 1. GANs can enhance data, but conventional GANs are typically designed for image-enhancement tasks. Therefore, to accommodate the tube-bending rebound data characteristics, it is necessary to modify the structure of the GAN. Additionally, considering the small sample size of the data, this paper intends to use Wasserstein distance as the optimization objective for the GAN. Compared to the traditional Jensen–Shannon divergence, the Wasserstein distance effectively alleviates the vanishing gradient problem encountered during training.

3.2. Tube-Bending Rebound Data Augmentation by Designing Generator and Discriminator Network. In this subsection, the generator and discriminator network are utilized to expand the tube-bending rebound data, and it combines CNN with GAN by introducing CNN into the generation model for unsupervised training. The robust feature extraction ability of convolutional networks is utilized to improve the learning efficiency of GANs, resulting in a more stable training process and higher-quality generated sample data. Both the generator and discriminator frameworks adopt CNN. The input of the generator network consists of a 512-dimensional latent variable, which is expanded through fully connected layers and up-sampling layers to adapt to higher dimensional data. Considering that the curvature rebound data is 1D, low-dimensional convolutional layers are used to reduce the dimensionality of the data. Batch normalization layers are added after each low-dimensional convolutional layer [25]. The filter size of the output convolutional layer is set to 1, corresponding to the rebound angle in the dataset. Except for
the output layer, the rest of the layers in the generation model use the ReLU function to address the gradient vanishing problem and accelerate convergence speed.

\[
\text{ReLU}(x) = \begin{cases} 
  x, & x > 0 \\
  0, & x \leq 0 
\end{cases}
\]  

The output layer, however, utilizes the Tanh activation function instead of the ReLU activation function.

\[
\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.
\]

The discriminator and generator networks have symmetric structures, with the key difference being replacing the activation function in the convolutional layers with LeakyReLU to improve discriminative performance [25]. The network ultimately outputs the probability that the input data is real measured data.

\[
\text{LeakyReLU}(x) = \begin{cases} 
  x, & x > 0 \\
  0.2x, & x \leq 0 
\end{cases}
\]

Before introducing the GAN algorithm, it is necessary to preprocess the data. This paper collected the real bending rebound data for 400 mm stainless steel tubes from the cooperative enterprise, as presented in Figure 2. The four input parameters are the inner diameter of the tube \(r\), the outer diameter of the tube \(R\), the wall thickness \(h\), and the bending angle \(\theta\). The output parameter is the rebound angle \(\theta'\). The real dataset of tube-bending rebound is divided into a training set and a Prediction set in an 8:2 ratio, as shown in Figure 2.

Due to the significant value differences among the parameters, data normalization is performed before model training to scale the values between [0,1]. This step aims to improve the convergence speed and accuracy of the model.

\[
x^*_i = \frac{x_i - \min_{i \in \mathbb{N}} \{x_j\}}{\max_{i \in \mathbb{N}} \{x_j\} - \min_{i \in \mathbb{N}} \{x_j\}}.
\]

The state \(x\) represents the input parameters before feature scaling, the state \(x^*_i\) represents the input parameters after feature scaling, and the variable \(n\) represents the total number of sample data.

During the training of the discriminator network, we initially collected \(m\) sample data from a joint Gaussian distribution and historical data to create a batch of training data. We then input \(z\) into the generator to generate measured data. Afterward, we calculate the discriminator’s loss value based on the optimization objective and update the network parameters using the Adam optimizer [26]. During the training of the generator network, we keep the discriminator network weights fixed. Similarly, we calculate the generator network’s loss value and update the network parameters using the Adam optimizer. Before each update of the generator network parameters, we update the discriminator network parameters to enhance training speed. The specific network architecture is shown in Figure 3.
Remark 2. By using the Wasserstein distance-based adversarial generative network designed in this paper, it is possible to expand the small sample tube-bending data. However, the generated data maintains the same interval of bending angle as the original data. Therefore, the extended data cannot be directly used for predicting tube-bending rebound. To this end, a prediction model is still required to fit the augmented dataset. Considering the characteristics of small sample tube-
bending data, this paper introduces SVR for predicting the tube-bending rebound model.

4. Rebound Model Prediction Algorithm Based on GAN–SVR

This section introduces SVR as a method suitable for handling small sample data. Then, this paper proposes a tube-bending rebound prediction algorithm based on GAN–SVR, which can effectively predict the tune bending rebound angle.

4.1. Data Fitting by SVR. SVR is a mathematical model that utilizes SVMs for regression calculations. It overcomes the limitations of traditional statistical learning theories in solving high-dimensional, nonlinear, and small sample problems. SVR applies the idea of SVMs by mapping low-dimensional data to a high-dimensional space and performing regression in the transformed space. It aims to find a hyperplane in this high-dimensional space that minimizes the variance between the points in the space and the hyperplane. This effectively solves the regression problem [22]. The bending rebound problem studied in this paper has the problem of a small number of samples. In terms of data type, tube-bending rebound data is nonlinear [27, 28], and SVR can adapt to such problems by changing the kernel function [23, 29], so using SVR to solve small samples and nonlinear problems is very effective. In this study, the bending rebound angle is taken as the output variable \( Y \), while the inner diameter, outer diameter, wall thickness, and bending angle of the tube are considered as input variables \( X_1, X_2, X_3, \) and \( X_4 \). The relationship between the input variables and the output variable can be expressed as follows:

\[
f(x) = (X_1, X_2, X_3, X_4).
\]

Given training sample data (Equation (11)) and the regression model is constructed as shown in Equation (12):

\[
D = \{(X_{1n}, Y_1), (X_{2n}, Y_2), (X_{3n}, Y_3) \ldots (X_{mn}, Y_n)\}. \tag{11}
\]

\[
f(x) = w^T x + b. \tag{12}
\]

In the equation, the state \( w \) represents the weight coefficients and \( b \) represents the bias term.

Figure 4 depicts the schematic diagram of SVR. SVR exhibits a distinct characteristic compared to conventional linear regression methods, necessitating a strict linear relationship and calculating loss when \( f(x) \) is not equal to \( y \) [30]. SVR allows for a tolerance deviation in the linear relationship. This is achieved by introducing a margin, as shown in Figure 4, with a distance of \( \epsilon \) on either side of the linear function. No loss calculation is needed if the sample data falls within this margin. However, some sample data in practical applications may not fall within this margin. Therefore, slack variables \( \xi_i \) and \( \xi'_i \) are introduced to relax the requirements for the margin.

After presenting the slack variables, the optimization problem and the constraint conditions for the objective function are shown as Equations (13) and (14):

\[
\frac{1}{2}||a^0|| + C \sum_{i=1}^{m} (\xi_i + \xi'_i). \tag{13}
\]

\[
\begin{align*}
& f(x_i) - y_i \leq \epsilon + \xi_i, \quad i = 1, 2, 3 \ldots m. \\
& y_i - f(x_i) \leq \epsilon + \xi'_i.
\end{align*} \tag{14}
\]

In the equation, \( a^0 \) represents the complexity of the model, and \( c \) represents the penalty factor. By introducing Lagrange multipliers \( \alpha, \alpha'_i, \mu, \mu'_i \), Equation (15) is obtained.

\[
L(\alpha, \xi, \xi') = \frac{1}{2}||a^0|| + C \sum_{i=1}^{m} (\xi_i + \xi'_i) - \sum_{i=1}^{m} \alpha_i ((\xi_i + \epsilon) - y_i + f(x_i)) - \sum_{i=1}^{m} \alpha'_i ((\xi'_i + \epsilon) - y_i + f(x_i)) - \sum_{i=1}^{m} (\mu_i \xi_i + \mu'_i \xi'_i).
\]

Taking the partial derivatives of the above equation concerning \( \alpha, \alpha'_i, \xi, \xi'_i \) and setting them equal to 0, substituting the obtained results into Equation (15), we bring the dual problem of SVR. Under the Karush-Kuhn-Tucker conditions, the final regression function is derived as Equation (16).

\[
f(x) = \sum_{i=1}^{m} (\alpha'_i - \alpha_i) K(x_i, x) + b. \tag{16}
\]

In the above equation, \( K(x_i, x) \) represents the kernel function. The performance of SVMs is somewhat dependent on the choice of the kernel function. The literatures [27, 28] proposes that the rebound data of bending pipe has strong
nonlinear characteristics, and this paper makes predictions based on small sample data, which usually has complex nonlinear relationships. The common linear kernel is for the case where the sample data is linearly divisible, and due to the small sample data due to the small amount of data, the polynomial kernel gives a non-zero weight on the entire feature space, while the RBF kernel takes a smaller value for the region far from the support vector. This means that the RBF kernel pays more attention to sample points near the support vector [23, 29, 31]. So, when the sample number is small and the sample data is nonlinear, the RBF kernel is more applicable. Therefore, we choose Gaussian RBF as the kernel function in this study, with the specific form shown in Equation (17).

\[
K(x, x) = \exp\left(-\frac{\|x_i - x\|^2}{2\sigma^2}\right). \tag{17}
\]

The operation of SVMs includes two parts: training and testing. Different penalty factors \(C\) and kernel parameters \(\sigma\) can affect the prediction accuracy of SVR. Therefore, an optimization algorithm is needed to select the optimal parameters.

4.2. The Prediction Algorithm of Tube-Bending Rebound by GAN–SVR. With the tube-bending rebound data augmentation obtained by the generator and discriminator network in Section 3.2, The Prediction Algorithm Of Tube-bending rebound by GAN–SVR is proposed by Algorithm 2.

The main steps of the GAN–SVR algorithm designed in this paper for predicting a rebound in tube-bending are depicted in Figure 5.

Remark 3. This paper uses SVR as the predictive model after data augmentation with GAN. In handling small sample problems, SVR exhibits good adaptability. Compared to conventional back-propagation (BP) methods and GAN–CNN methods, SVR is less prone to overfitting. Additionally, SVR can effectively handle nonlinear relationships in the data by utilizing kernel functions.

5. Experimental Study

In this section, we implement the algorithm using measured tube-bending rebound data and compare it with other algorithms for analysis. The real bending rebound data for 400 mm stainless steel tubes from the cooperative enterprise is collected in this paper. First, the 40-sample data of tube-bending rebound in the initial training set were randomly divided into four groups. Each group consists of 10 sample data of bending rebound data ranging from 10° to 100°, labeled as Group 1, Group 2, Group 3, and Group 4, respectively. The BP neural network and SVR were employed to train the four groups of real tube-bending data sets, and their performance was evaluated using a separate test set.

The comparative results based on the evaluation of the above criteria are shown in Figures 6 and 7.

According to Figures 6 and 7, it can be concluded that the BP neural network has certain limitations in predicting the rebound data of small samples tube-bending. When the data-set is small, the BP neural network has poor fitting capability for the data. This paper introduces the evaluation of the indicators as MSE, RMSE, and MAE below.

\[
\begin{align*}
\text{Mean squared error} & : \text{MSE} = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \\
\text{Root mean squared error} & : \text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2} \\
\text{Mean absolute error} & : \text{MAE} = \frac{1}{m} \sum_{i=1}^{m} |y_i - \hat{y}_i|
\end{align*}
\]  

(18)

The smaller the values of these indicators, the better the fitting effect of the model. Figures 6 and 7 show that compared to the BP neural network, the SVR demonstrates better appropriate performance for the small tube-bending rebound data samples, with smaller error values. This confirms that SVR outperforms the BP neural network in handling the small sample tube-bending rebound problem.

Next, we divided 40 sample data from the real tube-bending rebound training set into four groups in the following experiments. Each group consists of 10 bending rebound data sample data ranging from 10° to 100°, labeled as Group 1, Group 2, Group 3, and Group 4, respectively. The four groups were mixed based on the pairs: Group 1 and Group 2, Group 2 and Group 3, Group 3 and Group 4, Group 4 and Group 1. Each mixed group contains 20 bending rebound data sample data labeled as Mixed Group 1, Mixed Group 2, Mixed Group 3, and Mixed Group 4. The four new groups were used as inputs for the GAN, and ultimately, four sets of generated data were obtained from the generator, labeled as Group 5, Group 6, Group 7, and Group 8, respectively. The
**FIGURE 5: GAN–SVR algorithm process.**

**TABLE 2: Tube-bending rebound prediction algorithm.**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Data preprocessing&lt;br&gt;This step involves cleaning the dataset, handling missing values, and scaling the data. Scaling the data using the formula: ( x'<em>i = \frac{x_i - \min</em>{j \leq n} x_j}{\max_{j \leq n} x_j - \min_{j \leq n} x_j} ). Feature selection is performed on the bending tube dataset, which is partitioned.</td>
</tr>
<tr>
<td>2.</td>
<td>GAN training&lt;br&gt;The bending tube training dataset trains a generative adversarial network. The training process involves optimizing the objective function ( W(p_r, p_g) = \frac{1}{K} \sup_{f} \sum_{k \leq K} E_{x \sim p_r}[f(x)] - E_{x \sim p_g}[f(x)] ). The generator network learns to generate synthetic sample data resembling real rebound data.</td>
</tr>
<tr>
<td>3.</td>
<td>Synthetic data generation&lt;br&gt;The trained generator network generates multiple sets of synthetic rebound data sample data.</td>
</tr>
<tr>
<td>4.</td>
<td>Augmented dataset generation&lt;br&gt;The real and synthetic sample data are combined to create an enhanced dataset for training the support vector regression model.</td>
</tr>
<tr>
<td>5.</td>
<td>SVR model training&lt;br&gt;The SVR model is trained using the augmented dataset, where the selected features from step 2 are the input, and the target variable is the spring-back value. The final regression function in this paper is ( f(x) = \sum_{i=1}^{m} (\alpha_i - \alpha) \exp(- \frac{</td>
</tr>
<tr>
<td>6.</td>
<td>Prediction&lt;br&gt;The trained SVR model predicts the spring-back values for the bending tube data.</td>
</tr>
<tr>
<td>7.</td>
<td>Evaluation&lt;br&gt;The performance of the GA–SVR algorithm is evaluated using appropriate metrics such as:&lt;br&gt;( \text{MSE} = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}<em>i)^2 ), ( \text{RMSE} = \sqrt{\frac{1}{m} \sum</em>{i=1}^{m} (y_i - \hat{y}<em>i)^2} ), ( \text{MAE} = \frac{1}{m} \sum</em>{i=1}^{m}</td>
</tr>
<tr>
<td>8.</td>
<td>Iteration and optimization&lt;br&gt;The algorithm can be iterated and optimized to improve prediction accuracy by adjusting hyperparameters, feature selection, GAN training, or SVR training processes.</td>
</tr>
</tbody>
</table>
Figure 6: BP neural network prediction results.

Figure 7: SVR prediction results.

Figure 8: Augmenting data with generative adversarial networks.

Table 3: Generating rebound dataset using generative adversarial networks.

<table>
<thead>
<tr>
<th>Number</th>
<th>r (mm)</th>
<th>R (mm)</th>
<th>h (mm)</th>
<th>θ (°)</th>
<th>θ' (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.18</td>
<td>7.94</td>
<td>0.88</td>
<td>10</td>
<td>2.3</td>
</tr>
<tr>
<td>2</td>
<td>6.16</td>
<td>7.96</td>
<td>0.90</td>
<td>20</td>
<td>1.3</td>
</tr>
<tr>
<td>3</td>
<td>6.02</td>
<td>7.94</td>
<td>0.92</td>
<td>30</td>
<td>1.5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>40</td>
<td>6.12</td>
<td>7.94</td>
<td>0.90</td>
<td>100</td>
<td>2.9</td>
</tr>
</tbody>
</table>

Table 4: 400 mm aluminum tubes rebound dataset.

<table>
<thead>
<tr>
<th>Number</th>
<th>r (mm)</th>
<th>R (mm)</th>
<th>h (mm)</th>
<th>θ (°)</th>
<th>θ' (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17.10</td>
<td>18.90</td>
<td>0.90</td>
<td>20</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>17.10</td>
<td>18.90</td>
<td>0.90</td>
<td>25</td>
<td>0.6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>30</td>
<td>17.44</td>
<td>19.2</td>
<td>0.88</td>
<td>90</td>
<td>1.5</td>
</tr>
</tbody>
</table>
data generated by GANs cannot completely represent the data in the real world but can be regarded as similar to the data in the real world for use. The literature [15–18] all use the data generated by GANs as enhanced data sets and complete the tasks of data classification, data enhancement, and data prediction. In the experiment, after obtaining the generated data of GANs, we evaluated the data, and it can be seen from the final experimental results that the generated data of GANs can be used as reliable data similar to the real world for training and analysis.

**Table 5: The evaluation metrics for the four prediction algorithms.**

<table>
<thead>
<tr>
<th>Prediction methods</th>
<th>MSE</th>
<th>MAE</th>
<th>RMSE</th>
<th>Reduction in error magnitude after using an augmented dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>1.32</td>
<td>0.92</td>
<td>1.15</td>
<td>—</td>
</tr>
<tr>
<td>SVR</td>
<td>0.15</td>
<td>0.30</td>
<td>0.39</td>
<td>—</td>
</tr>
<tr>
<td>GAN–BP</td>
<td>1.03</td>
<td>0.78</td>
<td>1.01</td>
<td>22%</td>
</tr>
<tr>
<td>GAN–SVR</td>
<td>0.09</td>
<td>0.22</td>
<td>0.30</td>
<td>40%</td>
</tr>
</tbody>
</table>

**Figure 11:** SVR with linear kernel (a), polynomial kernel (b), and RBF kernel (c) prediction results.
In Figure 8, GAN learns from the combined dataset to generate new tube-bending rebound data at every 10° interval.

In Table 3, the generating dataset consisting of Group 5 to Group 6, with the real dataset, was retrained using the BP neural network and SVR and evaluated using the test set. The comparison results based on the above evaluation metrics are shown in Figures 9 and 10.

In Figures 9 and 10, it can be observed that GAN–BP shows lower MSE to small samples rebound data. However, its prediction accuracy is still not high. On the other hand, GAN–SVR exhibits a significant reduction in prediction errors compared to SVR. The evaluation metrics for the four prediction algorithms are compared in Table 4.

In Table 5, We use MSE as an evaluation index for calculation and compare the mean square error of GAN–BP with that of BP and, correspondingly, compare the mean square error of GAN–SVR with that of SVR.

\[
E_{\text{BP}} = \frac{0.132 - 1.03}{0.132} = 0.219. \tag{19}
\]

\[
E_{\text{SVR}} = \frac{0.15 - 0.09}{0.15} = 0.400. \tag{20}
\]

Compared to BP, GAN–BP has shown a 22% increase in predictive accuracy, demonstrating the effectiveness of GAN in data augmentation. Additionally, SVR is better than GAN–BP when using the real dataset. It has shown that SVR is more effective than BP neural networks in handling small sample data. Furthermore, GAN–SVR exhibits a 40% improvement in predictive accuracy compared to SVR, which indicates that dataset expansion with GAN can effectively alleviate the problem of insufficient data.

In addition, in order to verify the rationality of RBF, we measured the rebound data of another kind of tube from our partner company and recorded it from 20° to 90° (due to the large diameter of the tubes, the rebound angle of 10° and 15° is very serious, so the company’s measuring equipment could not measure its rebound angle, so the data started from 20°). The recording is shown in Table 4. In Table 4, we took the data from 20° to every 10° as a training set, used SVR of three different kernel functions (linear kernel, polynomial kernel, and RBF kernel) to train and predict the data, and used the remaining data as a test set to evaluate the prediction accuracy of SVR. Specifically, we used MSE and RMSE to evaluate the prediction accuracy of SVR.

In Figure 11, it can be observed that the RBF kernel is superior to the linear kernel and polynomial kernel in dealing with small samples and nonlinear problems.

6. Conclusion

This paper proposes a predictive algorithm for the angle of the tube-bending rebound with small samples. Based on the experimental results and evaluation metrics of various models with real datasets. The predictive accuracy of GAN–SVR is superior to SVR, which is superior to GAN–BP, which is superior to BP. The effectiveness of our proposed novel GAN–SVR has been demonstrated.

The potential and prospects of GAN–SVR in the field of tube-bending rebounds have been validated through experiments conducted with real bending rebound data from the cooperative enterprise. This has significant practical implications for the processing and predicting tube-bending rebound data in the engineering field, providing new insights and methods for further research.

Data Availability

The rebound data of bent pipes used in this research institute was provided by Zhejiang Heliang Intelligent Equipment Co., Ltd, China. Therefore, the data used in this study are not publicly available. If readers need to know the reliability of the data, do not hesitate to get in touch with the corresponding author.

Conflicts of Interest

The author declares that the prediction methods involved in this paper do not involve conflicts of interest.

Acknowledgments

This work was supported by the Huzhou Key Laboratory of Intelligent Sensing and Optimal Control for Industrial Systems under Grant 2022-17 and the enterprise postdoctoral joint training project between China University of Mining and Technology and Zhejiang Heliang Equipment Co., Ltd.

References


