Industrial Robot Vibration Anomaly Detection Based on Sliding Window One-Dimensional Convolution Autoencoder

ZhiDan Zhong, Yao Zhao, AoYu Yang, HaoBo Zhang, DongHao Qiao, and ZhiHui Zhang

School of Mechatronics Engineering, Henan University of Science and Technology, Luoyang, Henan 471003, China

Correspondence should be addressed to ZhiDan Zhong; zzd@haust.edu.cn

Received 6 October 2021; Revised 18 March 2022; Accepted 6 May 2022; Published 1 June 2022

1. Introduction

Industrial robots are the product of the fusion of multidisciplinary fields. They can replace humans to do some high-precision, high-intensity, high-risk, and high-repetition work [1]. Industrial robots are booming and becoming an essential part of the automation industry. Statistics from the International Federation of Robotics show that the number of industrial robot applications has increased year by year, and in 2018, it has exceeded 40,000 units [2]. With the booming development of industrial robots and the increasing number of applications, our reliance on them is on the rise, and if they break down during operation, they can cause loss of life and property. Therefore, it is very necessary to detect the abnormal operation of industrial robots.

In recent years, scholars at home and abroad have carried out a lot of research studies on anomaly detection of industrial robots. In general, anomaly detection methods fall into two main categories; the first is model-based anomaly detection methods. Chen et al. [3] proposed an algorithm that successfully detected minor collisions between two rotating planar manipulators and abnormal vibrations by combining a dynamics model and a friction model without the use of additional sensors. Dang et al. [4] proposed a fault detection algorithm and health assessment method for a robot operating system, which successfully detected faults and obtained the health status of industrial robots by considering internal and output measurement disturbances. The second category is based on data-driven anomaly detection methods: Vallachira et al. [5] proposed a combination of four preprocessing techniques to improve the performance of fault diagnosis for industrial robot gearboxes. Cheng et al. [6] proposed an unsupervised fault detection framework based on GMM, which utilizes the current signal to effectively detect the abnormal operation of industrial robots. Long et al. [7] combined sparse autoencoder (SAE) and support vector machine (SVM) to construct a fault diagnosis model for multijoint industrial robots by learning a pose dataset with
multiple fault information. Jaber and Bicker [8] adopt the time-frequency signal analysis method based on discrete wavelet transform to extract the most significant features related to faults and uses an artificial neural network to classify the faults of robot joint gears. Eski et al. [9] proposed a fault detection method based on vibration and noise signals, using radial basis neural networks (RBNN) to process the signals and achieve fault detection in industrial robots. Model-based anomaly detection technology needs to build an accurate mathematical model, which requires researchers to have a solid mathematical foundation and sufficient professional knowledge, in addition to spending a lot of time and effort to adjust parameters and optimize the model. However, data-driven anomaly detection techniques do not require much prior knowledge and detect anomalies completely from the collected signals and datasets.

In the field of data-driven fault diagnosis, there are usually two main types of methods, supervised and unsupervised. Among the supervised approaches, Wen et al. [10] proposed a LeNet-5-based CNN neural network that successfully performed fault diagnosis for bearings, centrifugal pumps, and hydraulic pumps by extracting features from the time-frequency map of the signal. Jing et al. [11] proposed a CNN model that can learn features and detect gearbox faults directly from the frequency data of vibration signal. Zhang et al. [12] proposed a TICNN model for bearing fault diagnosis, which can directly process raw vibration signals. For unsupervised or semisupervised algorithms, Sohaib et al. [13] proposed a deep neural network based on sparse stacked autoencoder (SSAE) to successfully detect bearing faults. Chen et al. [14] proposed a stacked variational autoencoder (VSAE) to adaptively extract features from angle signals to successfully detect diesel engine failures. The above research proves that fault diagnosis in a data-driven direction is very effective. Among them, the classical models of supervised algorithms, such as convolutional neural networks, rely on a large amount of data and labels. However, sometimes the failure data of mechanical equipment are very precious and scarce. For unsupervised or semisupervised algorithms, such as autoencoders, the network connections are usually fully connected layers, which have lower computational performance than convolutional networks.

In this study, a sliding window one-dimensional convolutional autoencoder (SW1DCAE)-based approach for unsupervised vibration anomaly detection in industrial robots is proposed. The main contributions are as follows:

(1) The SW1DCAE model uses convolutional, pooling, upsampling, and deconvolution layers instead of the fully connected layers of conventional autoencoders, allowing the raw sensor signal to be used directly as input to the model.

(2) The sliding window algorithm is used for data enhancement and shaping samples of a specific length, and the dropout algorithm is introduced to prevent the model from overfitting and enhance the generalization ability. The effects of different convolution kernel widths and sliding window lengths on the performance of the model are discussed, and reasonable parameter settings are selected to improve the accuracy and $F_1$ score of the model.

(3) SW1DCAE is an unsupervised detection algorithm that does not require label data and can be applied online without data accumulation. It has strong practical value and can be widely used in other fields such as vibration anomaly detection of industrial robots.

2. Theoretical Background

2.1. Autoencoder. Autoencoder is a classic unsupervised learning model, which can extract the hidden features of input samples and realize the reconstruction of samples. It usually consists of two parts, encoder and decoder, as shown in Figure 1.

Suppose the input signal is defined as $X = \{x_0, x_1, x_2, \ldots, x_t\}$.

The encoding process is

$$H = g(W_0X + b_0).$$

The decoding process is

$$\hat{Y} = g(W_1X + b_1),$$

where $X$ is the input vector, $H$ is the hidden feature, $\hat{Y}$ is the output vector, $W_0$ and $W_1$ are weight matrices, $b_0$ and $b_1$ are the biases, and $g$ is the activation function.

ReLU is the activation function, and the formula is

$$g = \max(0, x).$$

The loss function is the mean square error function, and the formula is expressed as

$$J = \frac{1}{n} \sum_{i=0}^{n} (\hat{Y} - X)^2.$$  

2.2. Convolutional Neural Network. Among supervised algorithms, the convolutional neural network is a very typical learning model. Convolutional neural networks usually consist of convolutional layers, pooling layers, and fully connected layers, as shown in Figure 2.

The convolution layer consists of a convolution kernel, which performs the convolution calculation. Its output equation is as follows:

$$y^m_n = \sum_{i=1}^{k} X^m_{i-1} \times K^m_{i-n} + b^m_n,$$

where $y^m_n$ is the convolution output of the $m$th channel of the $n$th layer, $X^m_{i-1}$ is the feature vector of the $i$th channel of the $n-1$th layer, $K^m_{i-n}$ is the convolution kernel, and $b^m_n$ is the offset.

The pooling layer is usually the maximum pooling layer, which can reduce the dimension of the feature surface and the training cost. The output formula is

$$p = \max(y_1, y_2, y_3, \ldots, y_n),$$

where $n$ is the width of the pooling area and $y_n$ is the $n$th data of the pooling area.
3. Methods

3.1. Dropout. This study introduces the node random dropout technique (Dropout) proposed by Srivastava et al. [15] to prevent overfitting and improve the generalization ability of the model. The principle is shown in Figure 4. The former is an ordinary neural network, and the latter is a neural network with dropout added.

Without dropout, the output of the neural network is
\[
    z_i^{(l+1)} = W_i^{(l)} \times Y_i^{(l)} + b_i^{(l)},
\]
\[
    y_i^{(l+1)} = \sigma(z_i^{(l+1)}),
\]
(8)

When dropout is introduced, the output of the neural network is
\[
    \bar{y}_i^{(l)} = r_i^{(l)} \cdot y_i^{(l)},
\]
\[
    z_i^{(l+1)} = w_i^{(l+1)} \times \bar{y}_i^{(l)} + b_i^{(l+1)},
\]
\[
    y_i^{(l+1)} = f(z_i^{(l+1)}),
\]
(9)

where \( r_i^{(l)} \) follows a Bernoulli distribution with parameter \( p \).

The dropout technique makes the neuron nodes choose randomly, and the abandoned neurons no longer participate in the calculation, thereby weakening the common adaptive relationship of neurons and improving the generalization ability.

3.2. Construction of the Model. The one-dimensional convolutional autoencoder takes the autoencoder as the basic architecture and the convolution kernel as the basic computing unit. It effectively integrates the convolutional neural network and the autoencoder model and extracts input samples through convolution, pooling, and deconvolution. The hidden features of the network structure are shown in Figure 5.

The SW1DCAE model is divided into three working phases: data processing, encoding, and decoding.

In the data processing stage, through the sliding window algorithm, the long time series is cut into several short time series, and training samples are generated so that the network can better capture the serial correlation of the data. Suppose the time step is \( T \) (the length of the window), the dimension of the observation data in the window is \( dx \), and the dimension of each input data is \( (T, dx) \).

The encoder consists of a one-dimensional convolutional layer, a dropout layer, and a one-dimensional pooling layer. The convolutional and pooling layers encode the data to find the implied features, and for the convolutional kernel, the output is

\[
    C_i = \text{ReLU} \left( \sum X \cdot \omega_i + b_i \right),
\]
(10)

where the ReLU activation function is
\[
    \text{ReLU}(x) = \max(0, x).
\]
(11)

The dropout layer effectively prevents overfitting. The pooling halves the dimensionality of the data and the output is
\[ p(n) = \max_{0 \leq n \leq L/s} T(nW, (n + 1)W), \quad \text{(12)} \]

where the pooling step size, \( s \), is set to 2 and the pooling window size, \( W \), is set to 2.

The decoder consists of a deconvolution layer, an upsampling layer, and the output of the upsampling layer is

\[ U_k = \begin{cases} 0, & k \neq j_k, \\ X, & k = j_k, \end{cases} \quad k \in [t, 2t], \quad \text{(13)} \]

where \( j_k \) is the location of the largest value recorded by the pooling layer.

The deconvolution layer is the reverse calculation of the convolution layer, and the output of the \( i \)th channel is

\[ D_i = \text{ReLU}(\sum X \otimes \omega_i + b_i), \quad \text{(14)} \]

where \( \omega_i \) is the deconvolution kernel and \( \otimes \) represents the deconvolution calculation. The training objective of this model is to minimize the reconstruction error between the input and output. The mean squared error function (MSE) was chosen as the loss function and expressed as follows:

\[ J = \frac{1}{n} \sum_{i=0}^{n} (\tilde{y} - x')^2. \quad \text{(15)} \]

The magnitude of the reconstruction error (MSE) is evaluated to determine whether an anomaly has occurred in the current input. In this study, the box-line plot method is chosen as the criterion for determining anomalies. The box-line graph method is based on quartile and interquartile distances and is more robust than the 3σ criterion as it does not require a hypothetical distribution of the data. This is shown in Figure 6.

When using the boxplot method to determine the abnormality, the data position other than the upper and lower edges is the abnormal point position, which is used as the judgment threshold. The formula for calculating the upper and lower edges is
where $Q_3$ is the upper quartile of the reconstruction error distribution produced by the SW1DCAE model in the normal state of the robot, $Q_1$ is the lower quartile of the reconstruction error distribution produced by the SW1DCAE model in the normal state of the robot, and IQR is the interquartile range.

The abnormal judgment is based on

$$\begin{cases} 
\text{LL} \leq \text{Loss} \leq \text{UL}, & \text{normal,} \\
\text{Loss} > \text{UL} \text{ or Loss} < \text{LL}, & \text{abnormal.}
\end{cases}$$

(17)

When the loss generated by the sample input into the SW1DCAE model exceeds the threshold, it is determined to be abnormal.

3.3. Model Training and Optimization. The SW1DCAE model learns the implicit features by reconstructing the input signal, and the loss function of the network model is

$$\text{Loss} = \text{avg} |X - \tilde{X}|^2,$$

(18)

where $X$ is the input signal and $\tilde{X}$ is the output signal.

The model parameters are optimized by the gradient descent method, and the calculation method for the gradient of the convolutional layer is

$$\frac{\partial L}{\partial K(j)} = \frac{\partial L}{\partial C} \frac{\partial C}{\partial K(j)} = \sum (x_i) \times \text{flip} \left( \frac{\partial L}{\partial C} \right),$$

(19)

where $x$ is the input and $C$ is the output, representing the convolution operation, and flip represents the transpose of the matrix.

For max-pooling layers, the gradient is calculated as

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial p} \frac{\partial p}{\partial x} = \begin{cases} 
\frac{\partial L}{\partial p}, & x = \text{max}(x), \\
0, & x \neq \text{max}(x),
\end{cases}$$

(20)

where $x$ is the input and $p$ is the output.

For upsampling layers, the gradient is calculated as

$$\frac{\partial L}{\partial U_i} = \frac{\partial L}{\partial U_i} \frac{\partial U_i}{\partial x_i} = \frac{\partial L}{\partial U_i} \text{ for } i \in [tk, (t+1)k],$$

(21)

where $x_i$ is the $i$th input, $U_i$ is the $i$th output, and $k$ is the upsampling step size.

The gradient of the deconvolution layer is calculated as

$$\frac{\partial L}{\partial K(j)} = \frac{\partial L}{\partial D} \frac{\partial D}{\partial K(j)} = \begin{cases} 
0, & i \neq j + km, \\
x_{k+1}, & i = j + km, \quad k = 0, 1, \ldots, t.
\end{cases}$$

(22)

Among them, $K(j)$ is the $j$th deconvolution kernel, $x$ is the input signal, $D$ is the output, $m$ is the deconvolution step size, and $t$ is the length of $x$ of the input signal.

The training and testing process for the SW1DCAE model is shown in Algorithm 1.

3.4. Vibration Anomaly Detection Process Based on the SW1DCAE Model. The vibration anomaly detection process based on the SW1DCAE model is shown in Figure 7. The process is divided into two parts: offline training and online detection. In the offline training part, we train the model.
4. Experimentation and Analysis

4.1. Experimental Setup. To demonstrate the validity of the SW1DCAE model, vibration anomaly detection was performed on vibration signal data collected from the APE sv45 six-axis industrial robot experimental rig. The experimental rig is shown in Figure 8 and contains the six-axis robot body, actuator, controller, and accelerometer.

The vibration signals were collected by accelerometers fixed to the robot axes at a sampling frequency of 150 Hz. The robot was operated in a PTP mode between set points, and the vibration signals collected during the experiment totaled over 110,000 points. The signals collected at this time are normal signals, which show periodic changes with the fixed running cycle of the robot. These data can be used to train the network model. To simulate anomalous conditions, we caused abnormal vibrations by manually hitting the robot. The criteria for determining abnormal vibration signals are based on statistics: signals exceeding the upper and lower limits of the normal signal amplitude box plot are identified as abnormal vibration amplitudes, and such signals are used as abnormal samples to construct the test set. Finally, choose the acceleration of the third axis and make a training set with 5836 samples and a test set with 5836 samples. All samples in the training set are normal signal samples, and the test set contains 2500 abnormal signal samples. Before training, the data were preprocessed to a standard score, i.e., a mean of 0 and a variance of 1. Some of the original time-domain signals are shown in Figure 9. The data stream needs to be cleaned before online detection to solve the common problem of missing data. The problem of missing data is solved by using the data completion method, that is, the nearest completion method. The last existing data are selected as the current missing value, which consumes almost no additional data and ensures real-time detection.

4.2. Experimental Part. During the experiment, the size of the sliding window was set to 50, the dropout ratio was set to 0.2, the batch size was set to 128, the Adam optimizer was selected, and the initial learning rate was 0.001. The structure of each layer in the network is shown in Table 1.

For the encoder part, a wider convolution kernel was chosen in the first convolution layer for better feature extraction. A window size of 2 and a step size of 2 were chosen in the pooling layer so that the dimensionality of the data could be reduced while retaining the important features. The parameter settings for the decoder part, the upsampling layer, and the deconvolution layer are the same as those corresponding to the encoder part.

The loss function during training computed with the mean squared error function (MSE) is shown in Figure 10. The loss functions of the training set and validation set gradually decrease with the increase of the number of iterations, and the training effect is good.

To test the reconfiguration effect of the SW1DCAE model on the samples, the vibration signal of the industrial robot in its normal operating state was used as input to observe the reconfiguration effect, as shown in Figure 11.

As can be seen from the figure, inputting the robot vibration signal in the normal state into the
Figure 7: Vibration anomaly detection process.

Figure 8: Industrial robot experiment bench.

Figure 9: Time-domain distribution of the original signal.
SW1DCAE model can reconstruct the input signal and correctly characterize the robot vibration signal in the normal state.

To test the response of the SW1DCAE model to abnormal signals, the experiment simulates vibration anomalies by impacting the robot and then inputs the signal containing the abnormal vibration amplitude into the model. The original and reconstructed signals are shown in Figure 12.

As can be seen from the diagram, the reconstructed signal produced a large error from the input original signal under the impact, which means that an anomaly may have occurred in the robot.

A comparison of the sample reconstruction error calculated by the model with the threshold is shown in Figure 13. The red dotted line represents the threshold, and the reconstruction error exceeding the threshold is judged as abnormal.

The model calculates the reconstruction error of the test sample and compares it with the abnormality judgment threshold. The sample whose reconstruction error exceeds the threshold can be judged as abnormal.

Since the model calculation takes a certain amount of time, the signal reconstruction and detection are delayed, and the delay time is about 9 ms. The model evaluation uses precision, recall, and $F_1$ score as evaluation metrics:

$$\text{precision} = \frac{TP}{TP + FP},$$

$$\text{Recall} = \frac{TP}{TP + FN},$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$  \hspace{1cm} (23)

To further demonstrate the superiority of the SW1DCAE model, it was compared experimentally with other detectors, as shown in Table 2.

1. SW1DCAE: model proposed in the text
2. AE: unsupervised learning of robotic vibration signals to detect anomalies using a classically structured autoencoder
3. KNN: a method for anomaly detection using the distance between different eigenvalues
4. SW1DCAE (binary cross-entropy loss function): binary cross-entropy is used as the loss function of this model for abnormal detection of vibration signals

As can be seen from the table, the model proposed in this study outperforms other models in terms of accuracy, recall, and $F_1$ score. The model can extract features directly from the original signal and reconstruct the signal, combining convolutional neural networks, autoencoder, and sliding windows effectively, thus showing good performance in robot vibration anomaly detection.
5. Parameter Sensitivity Analysis

In the model of this study, there are some important parameters, such as the length of the sliding window, the width of the convolution kernel, and the dropout ratio. In this section, we analyze these parameters.

The length of the sliding window plays an important role in whether the model can acquire data features effectively; on the contrary, the length of the sliding window is related to the computational cost and directly affects the performance of the model. The figure shows the performance of the model on the data when the sliding window length is between 30 and 150 and the first convolution kernel width is 32 by comparing the $F_1$ score and detection time. The graph shows that the $F_1$ score of the model is approximately 0.97 when the sliding window length is 50. As the sliding window length increases, the detection time also increases, which means that the training cost is increasing and the $F_1$ score tends to decrease, i.e., the performance of the model is decreasing. Therefore, we should not choose a sliding window that is too long, and the sliding window length of this model is chosen to be 50, as shown in Figure 14.

The input to the model is a one-dimensional vibration signal, and given that the convolution kernel of SW1DCAE has a filtering effect on it, it is necessary to discuss the effect of the width of the convolution kernel on the performance of the model. The figure shows the model training loss for different convolutional kernel widths for a sliding window length of 50. Wider convolution kernels can be effective in extracting data features from one-dimensional signals, but too wide a size can lead to computational complexity and redundant parameters, which in turn affect the features extracted. The fastest and lowest drop in loss occurs when the convolution kernel width is 32. This is shown in Figure 15.

Different convolution kernel widths have different extraction effects on the vibration signal. As can be seen from the figure, the $F_1$ score of the model tends to increase as the width of the convolution kernel increases, but too wide a convolution kernel does not continue to improve the

![Reconstruction diagram of abnormal signal.](image)

**Figure 12: Reconstruction diagram of abnormal signal.**
performance of the model but instead takes up a lot of resources and the detection time becomes longer. Therefore, in this study, 32 is chosen as the width of the convolution kernel for the first layer of convolution.

Dropout can suppress the occurrence of network overfitting, improve the generalization ability of the network, and make the neuron representation more independent. The dropout ratio affects the performance of the model, generally taking a value between 0.2 and 0.5 [15], and different ratios were chosen for comparison experiments, and the results are shown in Figure 16.

The curves in the figure represent the $F_1$ scores at different ratios, and the bars represent the loss of the model at different ratios. The experimental results show that when dropout reaches 0.2, the $F_1$ score of the model is the highest and the loss is the smallest, so the dropout ratio is chosen to be 0.2.

Batch size is the number of samples selected for training, and its size affects the degree of optimization and speed of the model [16]. Using $F_1$ score and detection time as evaluation metrics for different batch sizes, it can be seen from the figure that the $F_1$ score of the model is highest when the batch size is chosen as 128, considering the computational cost. This is shown in Figure 17.

![Figure 13: Sample reconstruction errors and thresholds.](image1)

**Table 2: Performance results of different methods.**

| Methods                  | Precision | Recall | $F_1$ score |
|--------------------------|-----------|--------|-------------|
| SW1DCAE                  | 0.9620    | 0.9743 | 0.9681      |
| AE                       | 0.8172    | 0.4871 | 0.6104      |
| KNN                      | 0.6278    | 0.7441 | 0.6802      |
| SW1DCAE (binary cross-entropy) | 0.55      | 0.07   | 0.125       |

![Figure 14: Performance of the model under different sliding windows.](image2)
Figure 15: Performance of the model with different convolution kernel widths.

Figure 16: The performance of the model under different dropout ratios.
6. Conclusion

This study proposes an unsupervised detection method for vibration anomalies in industrial robots based on SW1DCAE. The method uses a sliding window to directly shape the raw time-series signal to better identify normal and abnormal patterns in the data. In addition, the convolutional structure is introduced into the autoencoder model, which improves the learning ability of the model. Finally, experiments verify the superiority and feasibility of the method. The main conclusions are as follows:

(1) A one-dimensional convolution kernel is introduced as a computing unit in the autoencoder to effectively extract the spatiotemporal features of the data and improve the performance of identifying normal and abnormal patterns of the data. A sliding window algorithm is introduced to shape the original data into samples of suitable length to better capture the sequence correlation of the data.

(2) The method does not require much a priori knowledge of industrial robots and does not require the construction of mathematical or dynamical models, and the required vibration signals are common and easy to measure.

(3) Compared with other algorithms, SW1DCAE does not require data annotation and can be applied online without data accumulation. It has more practical value and can be widely used in other fields such as vibration anomaly detection of industrial robots.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.
neural network,” *International Journal of Prognostics and Health Management*, vol. 7, no. 2, 2016.

[9] I. Eski, S. Erkaya, S. Savas, and S. Yildirim, ”Fault detection on robot manipulators using artificial neural networks,” *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 115–123, 2011.

[10] L. Wen, X. Li, L. Gao, and Y. Zhang, ”A new convolutional neural network-based data-driven fault diagnosis method,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.

[11] L. Jing, M. Zhao, P. Li, and X. Xu, ”A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox,” *Measurement*, vol. 111, pp. 1–10, 2017.

[12] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, ”A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load,” *Mechanical Systems and Signal Processing*, vol. 100, pp. 439–453, 2018.

[13] M. Sohaib, C.-H. Kim, and J.-M. Kim, ”A hybrid feature model and deep-learning-based bearing fault diagnosis,” *Sensors*, vol. 17, no. 12, p. 2876, 2017.

[14] K. Chen, Z. Mao, H. Zhao, Z. Jiang, and J. Zhang, ”A variational stacked autoencoder with harmony search optimizer for valve train fault diagnosis of diesel engine,” *Sensors*, vol. 20, no. 1, p. 223, 2019.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, ”Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, 2014.

[16] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. Tang, ”On Large-Batch Training for Deep Learning,” *Generalization Gap and Sharp Minima*, 2016, https://arxiv.org/abs/1609.04836.