Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-019-01473-0

Control chart pattern recognition using the convolutional neural network

Tao Zan[1] · Zhihao Liu[1] · Hui Wang[1] · Min Wang[1] · Xiangsheng Gao[1]

Received: 11 December 2018 / Accepted: 1 April 2019

Abstract
Unnatural control chart patterns (CCPs) usually correspond to the specific factors in a manufacturing process, so the control charts have become important means of the statistical process control. Therefore, an accurate and automatic control chart pattern recognition (CCPR) is of great significance for manufacturing enterprises. In order to improve the CCPR accuracy, experts have designed various complex features, which undoubtedly increases the workload and difficulty of the quality control. To solve these problems, a CCPR method based on a one-dimensional convolutional neural network (1D-CNN) is proposed. The proposed method does not require to extract complex features manually; instead, it uses a 1D-CNN to obtain the optimal feature set from the raw data of the CCPs through the feature learning and completes the CCPR. The dataset for training and validation, containing six typical CCPs, is generated by the Monte-Carlo simulation. Then, the influence of the network structural parameters and activation functions on the recognition performance is analyzed and discussed, and some suggestions for parameter selection are given. Finally, the performance of the proposed method is compared with that of the traditional multi-layer perceptron method using the same dataset. The comparison results show that the proposed 1D-CNN method has obvious advantages in the CCPR tasks. Compared with the related literature, the features extracted by the 1D-CNN are of higher quality. Furthermore, the 1D-CNN trained with simulation dataset still perform well in recognizing the real dataset from the production environment.

Keywords Control chart · Pattern recognition · Convolutional neural network · Feature learning · Deep learning

Introduction

As a significant tool for the statistical process control (SPC), a control chart plays an important role in the manufacturing quality control, wherein it is widely used to monitor whether the machining process is in control or not. A manufacturing process is considered natural or normal if only random causes are affecting its operation (Zan et al. 2010). Therefore, unnatural control chart patterns (CCPs) displayed on the control charts can be associated with specific causes that adversely affect the manufacturing processes (Western Electric Company 1958). Nelson (1984) provided possible explanations and corrective actions for unnatural patterns. For instance, he stated that trend patterns might point out to wear or thermal deformation of key parts of a machine tool, while changes in operators, materials, or equipment might result in shift patterns; furthermore, the cyclic patterns might be related to the periodic variation in the power supply (Hachicha and Ghorbel 2012).

When some points exceed the boundaries of the control chart or when the control chart displays an unnatural pattern, that means the monitored manufacturing process is out of control (Montgomery 2007). The former can be easily recognized by the quality practitioners while identifying the latter requires some specific methods. For this reason, many scholars have formulated supplementary rules for detecting the unnatural patterns, such as the Geometric Moving Averages test (Roberts 1959), the Runs rules (Ducan 1986), and the Zone rules (Nelson 1985). However, based on the subsequent research Cheng (1997) pointed out that there is no one-to-one mapping relation between a supplementary rule and an unnatural pattern; moreover, utilization of too

✉ Zhihao Liu
liuzhihao@emails.bjut.edu.cn

Tao Zan
zantao@bjut.edu.cn

[1] Beijing Key Laboratory of Advanced Manufacturing Technology, Beijing University of Technology, Beijing 100124, China

Published online: 11 April 2019

many rules may lead to too many false alarms. Manual experience and knowledge are still needed to identify the alarm authenticity, which introduces an additional burden to the quality control (Ranaee and Ebrahimzadeh 2011). Accordingly, many studies pointed out that supplementary rules are often ineffective in recognizing the CCPs (Davis and Woodall 1988; Yang et al. 2015). Due to the deficiencies of supplementary rules and constantly increasing requirements for intelligent manufacturing, the interest in developing the accurate and automatic control chart pattern recognition (CCPR) algorithms has been significantly increased. Consequently, many CCPR methods have been developed recently, and they can be divided into two categories: expert system methods and machine learning methods.

Due to the limitation of the expert system methods, there is less research in this field. In 1987, Swift designed the first expert system for CCPR (Swift 1987). On this basis, many other studies have been conducted (Cheng and Hubele 1992; Kuo and Mital 1993; He et al. 2013) including the research on developing an expert system for CCPR. Bag (2012) focused on the design and development of an expert system for on-line detection of various control chart patterns to enable the quality control practitioners to initiate prompt corrective actions for an out-of-control manufacturing process. The typically used algorithms mainly included the statistical test and heuristic algorithm. The performance of these expert system methods is not excellent because their judgment rules are flawed (Zhou et al. 2018); however, their engineering application value is worth affirming.

On the other hand, the machine learning methods have been widely applied in the CCPR, and excellent results have been achieved. The machine learning methods can be classified into two categories, namely artificial neural networks (ANNs) and support vector machine (SVM). The ANNs based methods include supervised algorithms such as a multi-layer perceptron (MLP) (Cheng 1997; Pham and Wani 1997; Al-Assaf 2004; Ranaee and Ebrahimzadeh 2013), a learning vector quantization (LVQ) (Guh 2008; Gauri 2010; Yang and Zhou 2015), a probability neural network (PNN) (Cheng and Ma 2008), and a radial basis function (RBF) neural network (Addeh et al. 2018). Additionally, a fuzzy adaptive resonance theory map (ARTMAP) neural network having an incremental learning ability was proposed to recognize the CCPs (Zan et al. 2010); an unsupervised algorithm was used to realize the CCPs clustering analysis. It was reported that this method was superior to the traditional MLP method. However, the clustering results were more than the actual pattern type, resulting in an inconvenient use. In (Awadalla and Sadek 2012), a spiking neural network (SNN) architecture was proposed to be used for the CCPR. The SNNs represent the third ANN generation which considers time as an important feature for information representation and processing. Furthermore, the learning algorithm is improved to provide perfect learning rules. However, the ANNs have certain shortcomings, which hinder their universality and practicability, such as difficulty in convergence, ease relapsing into a local extremum, difficulty in determining a most suitable network structure, etc. As a new generation of the machine learning methods, the SVM has been widely applied in the CCPR achieving good results. In Zhou et al. (2018), a novel CCPR method integrating the Fuzzy SVM (FSVM) with the hybrid kernel function and genetic algorithm (GA) was proposed; the obtained simulation results demonstrated that the proposed method achieved an excellent performance and outperformed other approaches such as the LVQ the MLP, the PNN, fuzzy clustering, and the SVM, in term of the recognition accuracy. Zhao et al. (2017) proposed a CCPR method based on the improved supervised locally linear embedding and SVM. It was used to reduce the dimensionality of a high-dimensional feature set. The results showed that the dimension of the feature set had a great influence on the classification accuracy but the proposed method identified patterns correctly.

Another important problem in the pattern recognition field is a data form to be used as a classifier input, i.e., input data representation. The first form denotes raw data (Pham and Oztemel 1994; Hassan et al. 2003; Cheng and Ma 2008). However, there are some problems when the CCPR directly uses unprocessed CCP data such as high input dimension. Namely, high-dimensional input data usually results in too large classifier size, which leads to a reduction in accuracy and efficiency for complex recognition problems (Ranaee and Ebrahimzadeh 2011). The second form denotes the feature set, consisting of shape features (Pham and Wani 1997; Gauri and Chakraborty 2006, 2009; Pelegrina et al. 2016), statistical features (Pelegrina et al. 2016; Addeh et al. 2018), or wavelet analysis features (Jin and Shi 2001; Ranaee and Ebrahimzadeh 2011). Most related studies showed that the CCP classifiers using the feature set as an input achieved a significantly better performance compared to the classifiers using raw data as input (Hachicha and Ghorbel 2012). Besides, it is well known that choosing the most suitable feature set is the key to improve the recognition accuracy. Although many feature extraction methods have been proposed to solve the CCPR problem, and the features have been designed by experts for the required tasks, the full potential of the feature-based approach has not been fully exploited yet because the discarded raw data still contains much important information. Thus, the most suitable feature set for a CCPR is still unclear, and a more comprehensive and effective feature extraction method such as feature learning is needed.

Feature learning refers to the collection of techniques that learn a transformation or sequence of transformations of raw data so that the data is optimally represented for a required task (Janssens et al. 2016). In recent years, the deep neural networks (DNNs) have been widely applied in feature learning. Compared with the traditional ANNs, the special structure of the DNNs makes it possible to extract features from raw data (Ajm and Hulzebosch 1996). As a representative type of the DNNs, the convolutional neural networks (CNNs) have made remarkable achievements in the image recognition field. The CNNs use the alternating convolution and pooling layers to transform input data and optimize the convolution kernel by back propagation (BP) algorithm to minimize the cost (or loss) function, so that at the classification step, the input data is transformed optimally, i.e., the optimal features are learned for the classification task. With the popularity of deep learning, scholars have begun to use the CNNs for fault diagnosis (Liao et al. 2017; Xia et al. 2018; Xie and Zhang 2017). Janssens et al. (2016) used the CNNs to extract the vibration signal features and conduct the rotating machinery fault diagnosis, and the excellent result was obtained.

In this paper, a one-dimensional convolutional neural network (1D-CNN), which is sensitive to the time sequence, is proposed to extract the CCP features and realize the CCPR. To the best of our knowledge, only one article presenting the CNN-based CCPR has been published recently (Miao and Yang 2019). In Miao and Yang (2019), the mathematical methods were used to extract features (statistical and shape features) from raw data, and then a CNN was applied to the extracted features. However, it should be noted that their work differs from the work presented in this paper because here, no feature is extracted from the CCPs and the CNN-based model is applied to the raw data so that the network can learn optimal features for the CCPR. In other words, feature learning not only simplifies the model and reduces time consumption, but also can get more optimal features than the one extracted manually. In addition, the method we propose is helpful to improve the automation and intelligence level of quality management in the manufacturing process.

The rest of the paper is organized as follows. "Methodology" Section explains the simulation method of the CCPs and presents the recognition algorithm. "Proposed method" Section introduces the proposed method in detail. "Experiments" Section presents the verification test and gives the results. Subsequently, the results are discussed in "Discussion" Section. Lastly, "Conclusion" Section concludes the paper.

Methodology

Simulation method of CCPs

As shown in Fig. 1, there are generally six typical CCPs in the production process, namely, normal (NOR) pattern, upward- and downward-shift (US and DS) patterns, upward- and downward-trend (UT and DT) patterns, and cycle (CYC) pattern. Except for the NOR pattern, the rest of the unnatural patterns corresponding to certain unusual changes in a monitored manufacturing process.

The Monte-Carlo simulation is mostly used by scholars to provide a large number of the CCPs for the recognition algorithm. The process mean and two noise components are used to create the data points for the various patterns (Zan et al. 2010) as follows:

$$y(t) = \mu + x(t) + d(t) \tag{1}$$

where $y(t)$ denotes the value of a sample collected at time t, t is the time of sampling, μ is the statistical mean when the

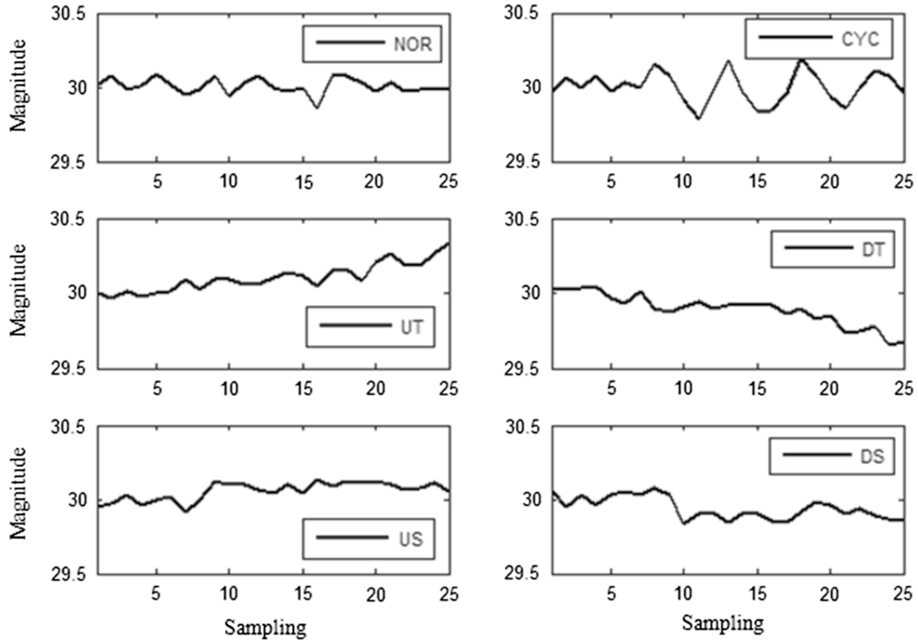


Fig. 1 Typical CCPs in the production process

process is in control, $x(t)$ is a random noise at time t, and it obeys normal distribution, $x(t) \sim N(0, \sigma)$; σ is the standard deviation when the process is in control, and $d(t)$ is the special disturbance caused by specific factors in the manufacturing process at time t.

Based on Eq. (1), the simulation method of various typical CCPs is as follows.

The NOR pattern is given by:

$$d(t) = 0 \tag{2}$$

The US and DS patterns are given by:

$$d(t) = \pm v \times s \tag{3}$$

where v is parameter determining the shift position, and it is equal to 0 before the shift and to 1 after the shift, s is the shift magnitude; sign "+" is used for the US pattern, and sign "−" is used for the DS pattern.

The UT and DT patterns are given by:

$$d(t) = \pm v \times d \times t \tag{4}$$

where v is parameter determining the trend position, and it is equal to 0 before the trend and to 1 after the trend, d is the slope of a trend; the sign "+" is used for the UT pattern, and sign "−" is used for the DT pattern.

The CYC pattern is given by:

$$d(t) = v \times a \times \sin(2\pi t/\omega) \tag{5}$$

where a is the amplitude of a cycle, and ω is the period of a cycle.

CNN model

Deep learning has made outstanding achievements in the field of pattern recognition. Compared with the traditional machine learning methods, deep learning has many advantages. (1) Data pre-processing can be omitted completely, and raw data can be directly used for model training and testing. (2) A multi-layer neural network can be used to learn deeper knowledge, and it is competent for more complex tasks. (3) The most appropriate features can be learnt for classification tasks. The comparison of deep learning and traditional machine learning methods is shown in Fig. 2.

A CNN is a type of a deep neural network. As shown in Fig. 3, a CNN is made up of three types of layers: convolutional layer, subsampling layer (or pooling layer), and fully connected layer with a cost function. A typical CNN structure can, therefore, be divided into two parts; namely, the convolutional and pooling layers work as a feature extractor, and the fully connected layer works as a classifier (Xie and Zhang 2017).

A convolutional layer is the most important component of a CNN. The weights and biases of a convolutional layer are organized into a series of convolutional kernels (or filters). A set of output feature maps can be acquired by using different filters. Each output feature map is the result of a convolution of multiple input feature maps and multiple convolutional kernels, which is given by:

$$\boldsymbol{x}_j^l = f\left(\sum_{i=1}^{D_{l-1}} \boldsymbol{x}_i^{l-1} * \boldsymbol{\omega}_{ij}^l + \boldsymbol{b}_j^l\right), \quad j = 1, 2, \ldots, D_l \tag{6}$$

where * represents the convolution operation, l represents the serial number of current network layer, D is the number of feature maps, $\boldsymbol{\omega}^l$ is a convolutional kernel connecting the (l−1)th layer to the lth layer, and its size is $r \times c$, r represents height, c represents width, $\boldsymbol{x}_j^l$ represents the jth output feature map, $\boldsymbol{b}$ is the additive bias of each output feature map, and f is an activation function.

Most commonly used nonlinearity activation functions are Sigmoid function and rectified linear units (ReLU) function, which are respectively given by:

$$f(x) = 1/(1 + e^{-x}) \tag{7}$$

$$f(x) = max(0, x) \tag{8}$$

The size of the output feature map of the lth convolution layer is $R_l \times C_l$, R represents height, C represents width, and it is calculated by:

$$R_l \times C_l = \left[\left(R_{l-1} - r\right)/s + 1\right] \times \left[\left(C_{l-1} - c\right)/s + 1\right] \tag{9}$$

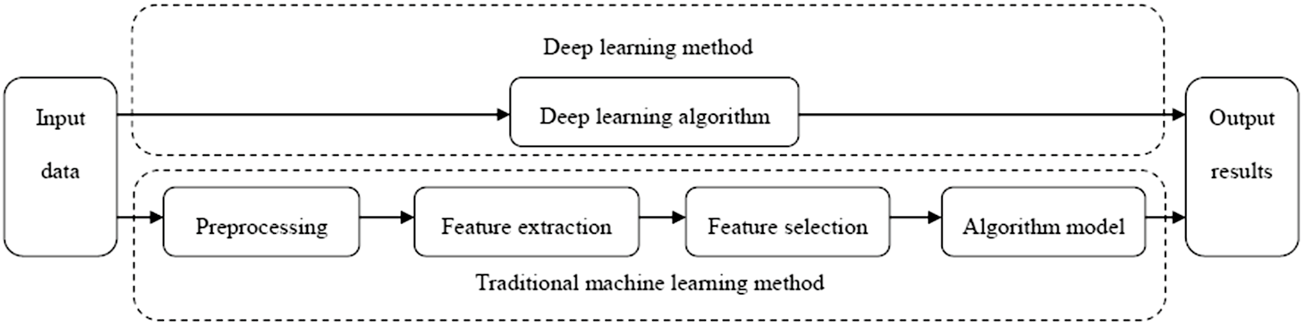


Fig. 2 Comparison of deep learning and traditional machine learning

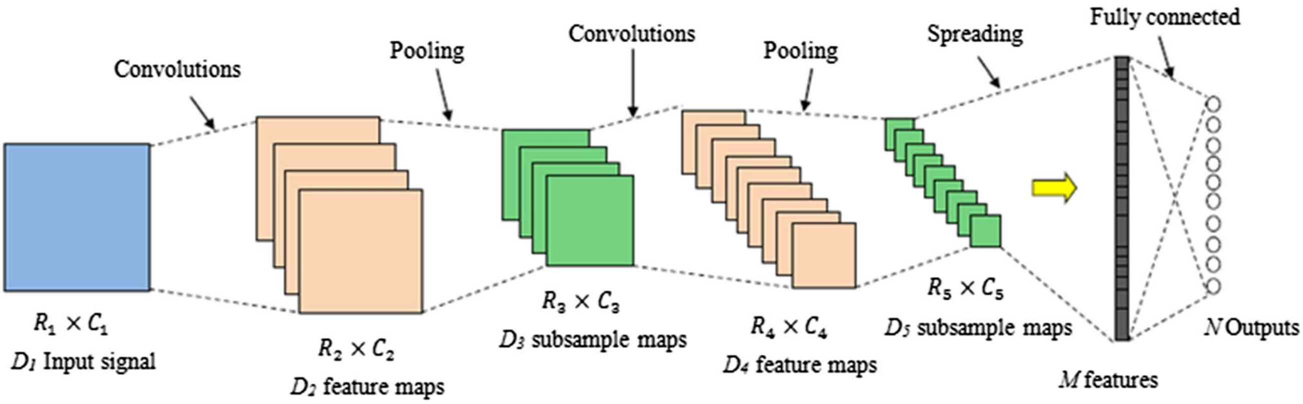


Fig. 3 Typical CNN structure

where s is the moving stride of a convolution kernel, and in this work, s is set to 1.

In the subsampling layer, the downsampling is completed so that the dimension of the feature maps can be quickly reduced. A subsampling layer is mathematically represented by:

$$\boldsymbol{x}_j^l = down\left(\boldsymbol{x}_j^{l-1}\right), \quad j = 1, 2, \dots, D_l \tag{10}$$

where l represents the serial number of the current network layer, D_l is the number of input feature maps, $\boldsymbol{x}_j^l$ represents the jth output subsample map, and *down* represents a pooling function.

The commonly used subsampling strategies are the max pooling and the average pooling. In the max pooling strategy, the maximum value of the subsampling region is taken as a new feature, while, in the average pooling strategy, the mean value of the subsampling region is taken as a new feature. Scholars believe that max pooling reflects the most striking feature of the subsampling region, while the average pooling selection is more smooth (Xie and Zhang 2017).

The size of the output subsample map of the lth subsampling layer is $R_l \times C_l$, and it is calculated by:

$$R_l \times C_l = \left(R_{l-1}/u\right) \times \left(C_{l-1}/u\right) \tag{11}$$

where u is the step size of pooling operation. In this work, u is set to 2.

As shown in Fig. 3, the feature maps are expanded and spliced together in the first fully connected layer. The number of neurons in this layer is M, representing M features extracted by the CNNs. The M is calculated by:

$$M = R_{l-1} \times C_{l-1} \times D_{l-1} \tag{12}$$

Neurons in a fully connected layer have full connections to all neurons in the previous layer like in a regular neural network. Hence, they can be computed with a matrix multiplication followed by a bias offset:

$$\boldsymbol{O} = f\left(\boldsymbol{\omega}_o \boldsymbol{f}_v + \boldsymbol{b}_o\right) \tag{13}$$

where $\boldsymbol{f}_v$ is the input vector of a fully connected layer, $\boldsymbol{b}_o$ is the bias vector, and $\boldsymbol{\omega}_o$ is the weight matrix.

The last layer in a CNN is the output layer, containing N neurons represents the number of pattern types to be identified. Usually, the activation function of the output layer is a Sigmoid function or a Softmax function, which are respectively given by Eqs. (7) and (14). Finally, in the training phase, the BP algorithm is used to optimize the weights and biases ($\boldsymbol{\omega}_{ij}^l$, $\boldsymbol{\omega}_o$, $\boldsymbol{b}_j^l$, and $\boldsymbol{b}_o$) in a CNN to minimize the value of the cost function.

$$f\left(x_i\right) = e^{x_i} / \sum_{j=1}^{N} e^{x_j} \tag{14}$$

In this paper, a 1D-CNN, which is slightly different from a typical CCN, is used to complete the CCPR. More details about the 1D-CNN structure are introduced in the following sections.

Proposed method

Manual feature extraction from raw CCP data can improve the recognition accuracy, but it also increases the workload and complexity of the quality control. Therefore, this paper proposes a 1D-CNN to complete the CCPR so as to extract the CCP features through the feature learning.

Compared with the traditional machine learning method, the advantage of a CNN is that it can realize end-to-end recognition or diagnosis. In other words, the input of the neural network is raw data, and the output is the pattern type. Feature extraction, feature selection, and feature optimization are completed by the convolution and pooling layers of a CNN. The weights and biases of a CNN structure are optimized and adjusted by the BP algorithm by minimizing the cost function. Thus, the best feature set is obtained through the CNN learning process. In that way, a lot of manpower is saved, and complex work is finished by a neural network. The structure of the proposed 1D-CNN is slightly different from a common CNN structure. The feature mapping in the 1D-CNN structure is not a matrix but a vector, which makes the 1D-CNN particularly sensitive to the time sequence such as a CCP.

The structure of the proposed CCPR method is shown in Fig. 4, where it can be seen that the proposed CNN structure consists of two convolution layers, two pooling layers, and a fully connection layer, such that the convolution and pooling layers complete feature extraction, and the fully connection layer realizes classification. Since the proposed network represents a 1D-CNN, the CNN structural parameters C and c (the width of feature maps and convolutional kernels) equal to 1. The dimension of the raw CCP data is 25, and there are six different patterns as mentioned earlier; thus, the structural parameters are $R_1 = 25$ and $N = 6$. The other structural parameters of the 1D-CNN, such as the number of feature maps (D) and the height of the convolution kernel (r), will be discussed and optimized in the next section. The pooling function of the subsampling layer is max pooling.

Since the deep learning is used in the proposed CCPR method, the main steps of the proposed method are very simple, and they are as follows:

Step 1 The training and test sets containing the raw CCP data are generated by the method introduced in "Methodology" Section.

Step 2 Training set is used to train the 1D-CNN and optimize the network weights and biases.

Step 3 The optimized 1D-CNN is validated by the test set.

Experiments

A series of simulations were conducted to verify the feasibility and effectiveness of the proposed method. The 1D-CNN was implemented by using Python programming program, and it was run on a personal computer with a 2.20-GHz CPU and 2 GB RAM. The correct recognition ratio (CRR) was used as an estimation criterion, and it represented the ratio of the number of correctly classified patterns to the total number of test patterns. The confusion matrix was used to present the evaluation results more intuitively. Finally, the validity of 1D-CCN was evaluated through the comparison with the existing methods of the same type.

CCP parameters

Six common CCPs were considered, namely NOR, UT, DT, US, DS, and CYC. The Monte-Carlo simulation algorithm introduced in "Methodology" Section was used to generate

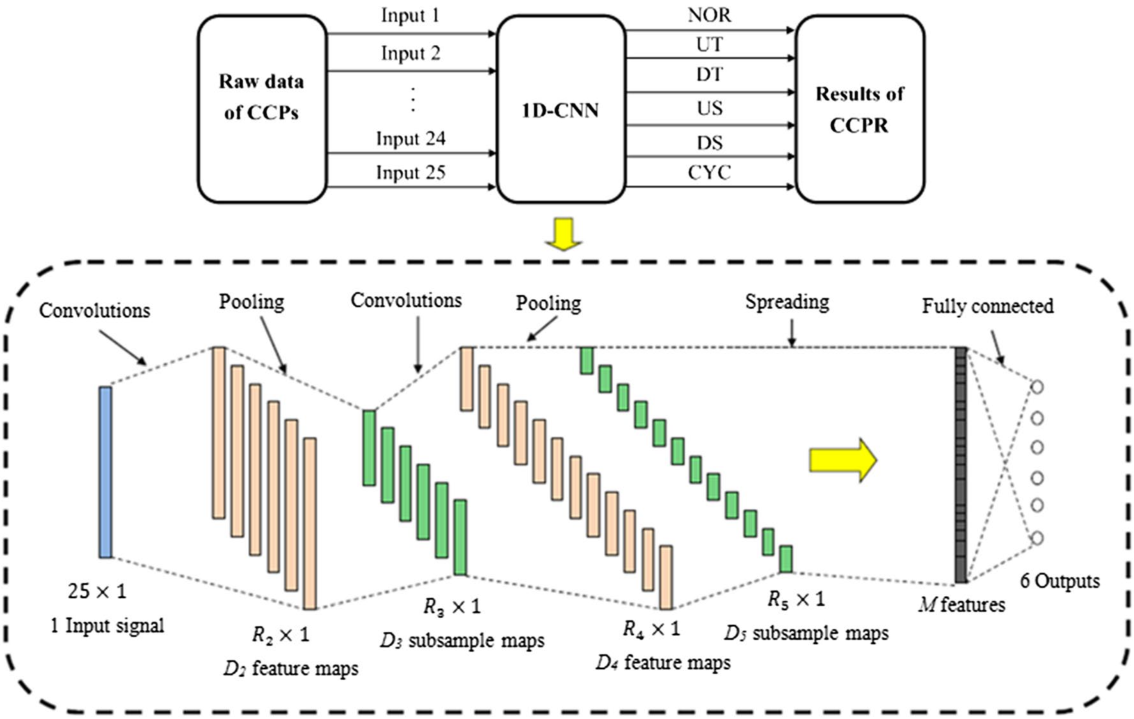


Fig. 4 The structure of the proposed CCPR method

datasets for training and testing of the 1D-CNN. The parameters of the six CCPs are shown in Table 1.

In the actual manufacturing process, the change of a monitored object is very complicated. In order to simulate this situation more truly, the CCP parameters were randomly selected within a certain range, using the uniform distribution. The mean μ was set to 30, and the standard deviation σ was set to 0.05. The value of the slope d varied in the range [0.1σ, 0.3σ]. The magnitude of the shift s changed in the range [1.5σ, 3σ], and the amplitude of cyclic patterns a varied in the range [1.5σ, 4σ]. The value of the period of cycle ω was respectively set to 4, 5, 6, 7, and 8. The starting position of each unnatural pattern was in the range [4, 9]. The data length of the CCPs was 25. By comparing the recognition rate of different sample size, it was found that when the number of samples exceeds 2000, the improvement of the recognition rate will not be significant. At the same time, the computational cost of the experiment increases exponentially. After taking these factors into account, the generated dataset contained 2000 samples per CCP, which was a total of 12,000 different samples. The dataset was random divided into two parts, of which 9600 samples were used to train 1D-CNN, and the rest was used for testing.

1D-CNN performance dependence on structural parameters

In order to determine an optimal 1D-CNN structure, the 1D-CNN performance was compared for different structural parameters. The comparison results are shown in Table 2. Namely, two sets of experiments were carried out with the same dataset. The activation function of each 1D-CNN layer was the Sigmoid function. The purpose of the first experiment was to optimize the size of the convolution kernel (r), and the results are given in Table 2. Afterward, the second experiment was carried out to optimize the number of feature maps (D), and the results are also presented in Table 2. The mean square error (MSE) of the experiments is shown in Fig. 5.

We trained the neural network with different structural parameters. The respective CRR, the training time of each epoch, and the MSE were determined. As it can

Table 1 The parameters of six CCPs

Pattern	Mathematical representation	Parameter value/range
NOR	$y(t) = \mu + x(t)$	$\mu = 30,\ \sigma = 0.05$
UT	$y(t) = \mu + x(t) + v \times d \times t$	$d \in [0.1\sigma, 0.3\sigma]$
DT	$y(t) = \mu + x(t) - v \times d \times t$	$d \in [0.1\sigma, 0.3\sigma]$
US	$y(t) = \mu + x(t) + v \times s$	$s \in [1.5\sigma, 3\sigma]$
DS	$y(t) = \mu + x(t) - v \times s$	$s \in [1.5\sigma, 3\sigma]$
CYC	$y(t) = \mu + x(t) + v \times a \times \sin(2\pi t/\omega)$	$a \in [1.5\sigma, 4\sigma], \omega \in \{4, 5, 6, 7, 8\}$

Table 2 The 1D-CNN performance at different structural parameters

	No.	D_2, D_3	r_2	D_4, D_5	r_4	CRR (%)	Time (s/epoch)
Experiment 1	**1**	**6**	**2**	**12**	**9**	**97.23**	**0.935**
	2	6	2	12	7	97	0.962
	3	6	2	12	5	96.5	0.992
	4	6	2	12	3	97.27	1.010
	5	6	4	12	8	97.1	0.923
	6	6	4	12	6	96.87	0.987
	7	6	4	12	4	96.67	1.038
	8	6	4	12	2	96.73	1.052
	9	6	8	12	6	96.87	0.905
	10	6	8	12	4	96.07	1.014
	11	6	8	12	2	96.33	1.091
	12	6	16	12	2	96.13	0.873
Experiment 2	13	1	2	2	9	81.07	0.129
	14	2	2	4	9	95.13	0.211
	15	3	2	6	9	96.2	0.340
	16	12	2	24	9	96.93	3.290
	17	24	2	48	9	97.23	11.624

The best results are shown in bold

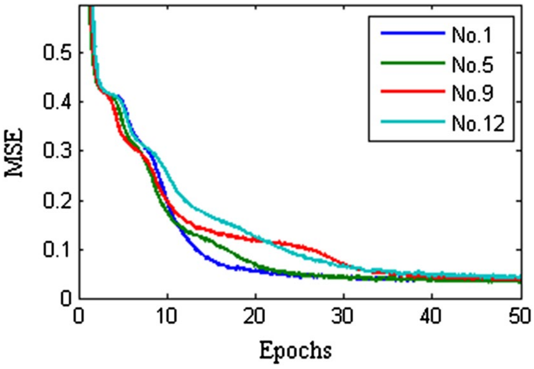


Fig. 5 The MSE of the experiments. The numbers in the figure legend correspond to the numbers in Table 2

be seen in Table 2 and Fig. 5, the first combination of structural parameters had the highest CRR, the shorter training time, and the fastest convergence speed. Therefore, it denoted the best-performance 1D-CNN structure, which was used in the subsequent experiments. All the network parameters were determined, and they are summarized in Table 3.

Influence of activation function on recognition performance

To select the most convenient activation function, we performed a set of simulations whose results are shown in Table 4, and the corresponding MSE is shown in Fig. 6. In scheme 1, the activation function of each layer in the 1D-CNN was the Sigmoid function, and in scheme 2, the activation function of the convolution layer was the ReLU function, and the activation function of the output layer was the Softmax function. As it can be seen in Table 4, the 1D-CNN with the ReLU and Softmax activation functions performed better than that with the Sigmoid function; both the CRR and the convergence speed were significantly better.

Table 4 Comparison of 1D-CNNs with different activation functions

Activation function	CRR (%)	MSE	Time (s/epoch)
Sigmoid	97.23	0.0343	0.935
ReLU-Softmax	**98.33**	**0.0126**	**0.910**

The best results are shown in bold

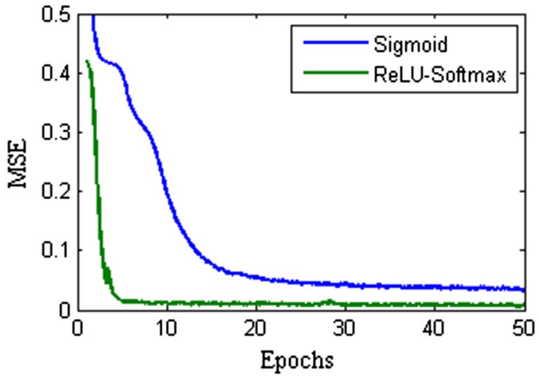


Fig. 6 The MSE for different activation functions

Performance comparison between MLP and 1D-CNN

To evaluate the performance of the proposed method further, a comparative experiment of 1D-CNN, CNN and MLP was conducted. As already mentioned, the MLP as a machine learning method with an excellent performance has been widely used in the CCPR and has achieved good results (Pham and Wani 1997; Al-Assaf 2004; Ranaee and Ebrahimzadeh 2013), which is why we used it in this work. The input of 1D-CNN was raw data, the input of CNN was control chart image data, which were 60×60 pixels in size ($R_1 = C_1 = 60$), and the MLP input was either the raw data or the feature set. The feature set included the statistical features (mean, standard deviation, skewness, and kurtosis) and shape features (S, NC1, NC2, APML, APSL), which have been proved to be very effective in the CCPR (Pham and Wani 1997; Hassan et al. 2003;

Table 3 Structural parameters of the 1D-CNN

Parameter	Value
Number and size of the input layer	$D_1 \times R_1 \times C_1 = 1 \times 25 \times 1$
Number and size of the first convolutional layer feature map	$D_2 \times R_2 \times C_2 = 6 \times 24 \times 1$
Size of the first convolutional kernel	$r_2 \times c_2 = 2 \times 1$
Number and size of the first pooling layer subsample map	$D_3 \times R_3 \times C_3 = 6 \times 12 \times 1$
Number and size of the second convolutional layer feature map	$D_4 \times R_4 \times C_4 = 12 \times 4 \times 1$
Size of the second convolutional kernel	$r_4 \times c_4 = 9 \times 1$
Number and size of the second pooling layer subsample map	$D_5 \times R_5 \times C_5 = 12 \times 2 \times 1$
Number of features	$M = 24$
Number of neurons in the output layer	$N = 6$

Gauri and Chakraborty 2009; Pelegrina et al. 2016). A detailed description of the features can be found in (Pham and Wani 1997; Hassan et al. 2003). The MLP used in the experiment was a typical three-layer neural network, containing 25 neurons that were receiving the CCPs raw data, or 9 neurons which were receiving the feature set. There were 15 neurons in the hidden layer, and 6 neurons in the output layer corresponding to the six typical CCPs. The activation function of all layers was the Sigmoid function. The structural parameters of the CNN were as follows: $r_2 = c_2 = r_4 = c_4 = 5$, $u_3 = 4$, $u_5 = 2$, and other parameters were the same as the 1D-CNN. In order to evaluate the classifier performance more intuitively, the confusion matrix was used. The values on the confusion matrix diagonal denoted the percentage of correctly recognized patterns. The other values in the matrix represent the percentage of misclassifications (Sokolova and Lapalme 2009). The CRR was equal to th e average value of all the elements on the matrix diagonal. The confusion matrix is shown in Fig. 7, where it can be clearly seen that the performance of the 1D-CNN exceeded other methods performance. Namely, the 1D-CNN achieved higher recognition accuracy and lower error rate.

In addition, the performance in the three cases was compared using different number of samples per pattern. As shown in Table 5, the number of samples of each CCP was respectively set to 200, 2000, or 20,000. The recognition rate and time of each epoch were determined, and obtained results are shown in Table 5, where it can be demonstrated that the proposed method was superior to the traditional MLP method and the CNN method with image data as input in terms of both recognition accuracy and time consumption.

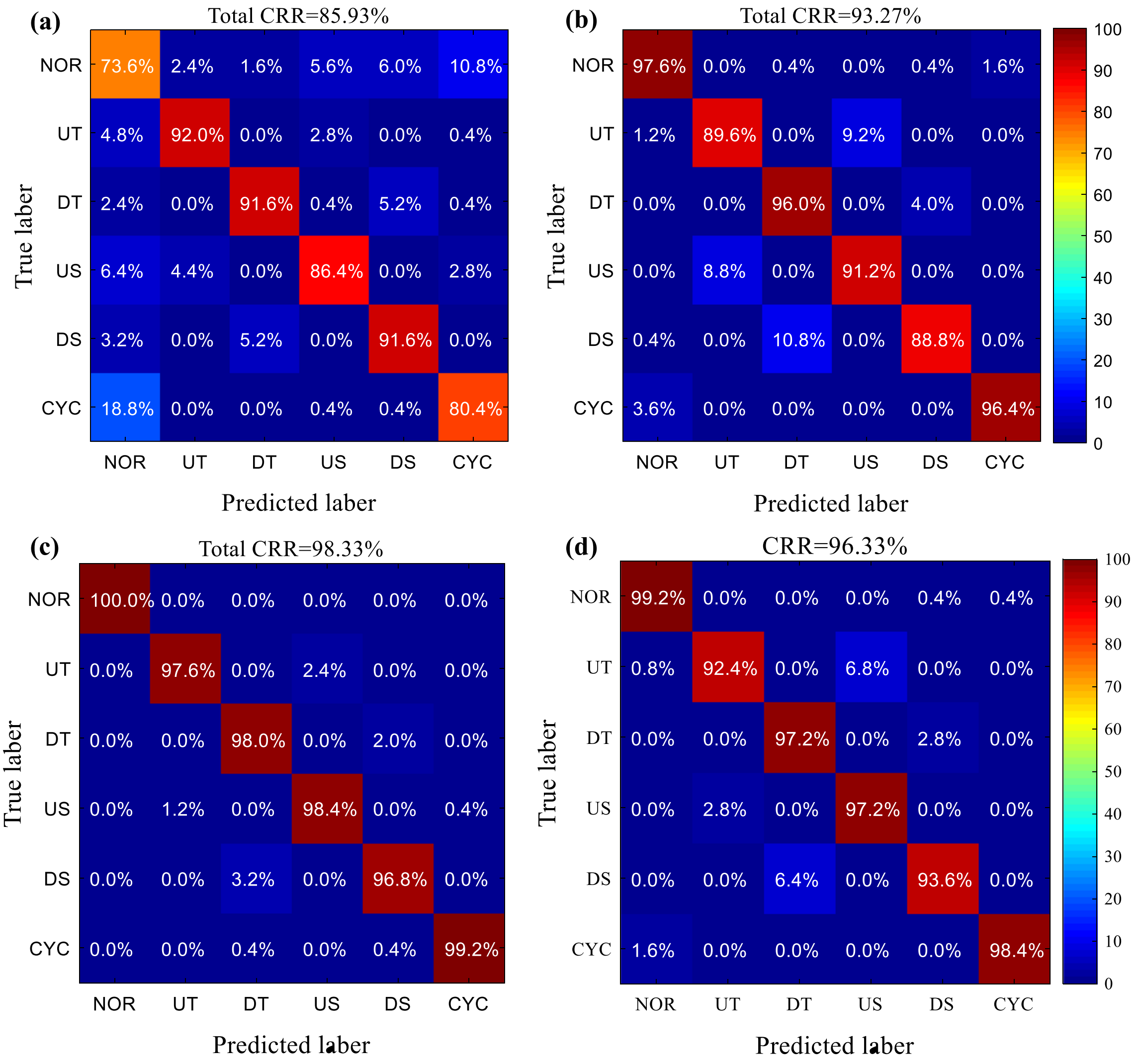


Fig. 7 The CCPR confusion matrix for **a** the MLP and raw data, **b** the MLP and feature set, **c** the 1D-CNN and raw data, and **d** the CNN and image data

Table 5 Comparison results for different numbers of samples

Number of samples per pattern	MLP and Raw data		MLP and Feature set		1D-CNN and Raw data		CNN and Image data	
	CRR (%)	Time (s/epoch)	CRR (%)	Time (s/epoch)	CRR (%)	Time (s/epoch)	CRR (%)	Time (s/epoch)
200	82.61	1.14	92.00	0.33	92.40	0.11	93.20	2.66
2000	85.93	6.59	93.27	2.20	98.33	0.91	96.33	23.16
20,000	87.07	52.44	93.54	17.43	98.31	9.02	96.78	228.96

Comparison of 1D-CNN and other methods

To further evaluate the effectiveness of the CNN, the proposed method was compared with the methods reported in the related literature. As shown in Table 6, many classifiers, including the MLP, PNN, Fuzzy ARTMAP, SVM, and RBF, were used for the CCPR. The input was either raw data or feature set. Based on the results presented in Table 6, using the features as an input was generally more effective than using the raw data. Particularly, when image and statistical features were combined, the performance was most prominent. When the input of the 1D-CNN was a raw signal, the final output of the convolution and pooling layers could be understood as an optimal feature set extracted by the neural network through the learning process. The box-plots of some features are plotted in Fig. 8. The total number of features M was equal to 24, but due to the limited space, only six features are displayed in Fig. 8. In Zhou et al. (2018) and Ranaee and Ebrahimzadeh (2013) similar box-plots of shape and statistical features were given. In Fig. 8, the values of features of the same CCP type were closer to each other and separated from that of the other types in the feature space, which proved that CNN learned features from the raw data more than excellent.

The results showed that the recognition rate of the proposed method was 98.33%, representing one of the higher results, but not the highest, which might be caused by the following reasons. The datasets used in training and testing classifiers were different. Also, in some literature, the parameter setting lacked the randomness when generating simulation data, such as selecting only a few fixed values instead of uniform distribution in a certain range, or the parameter setting range was too narrow, which could deviate from the actual CCPs. With the aim to make the simulation data more close to the real patterns, the parameters presented in Table 1 were adopted, and the parameters obeyed a uniform distribution in a wide range. This inevitably increased the difficulty of the CCPR, resulting in a decrease in the recognition rate, which was caused by large randomness of parameters and the existence of a random noise $x(t)$ [see Eq. (1)] so that the CCPs dataset generated by the simulation algorithm contained some dirty data. In Fig. 9a, the pattern computed by Eqs. (1) and (4) is presented, and according to its category label in the dataset it represented a UT pattern, but this pattern was more like a US pattern, and it was also recognized as a US pattern by the 1D-CNN. Even when the CCP was identified manually, the same result was obtained as that by the 1D-CNN. In other words, the 1D-CNN actually achieved the correct recognition. Similarly, in Fig. 9b, a US pattern was identified, but actually it should be UT pattern. Therefore, some misclassification in Fig. 7c was partly caused by dirty data in the dataset, not real errors. As shown in the last row in Table 6, after removing the dirty data from the dataset, the 1D-CNN had the CRR of more than 99.30%.

Table 6 Performance comparison of different methods

Reference	Number of CCP patterns	Input representation	Classifier	CRR (%)
Guh and Tannock (1999)	4	Raw data	MLP	94.38
Hassan et al. (2003)	6	Statistical features	MLP	96.80
Assaleh and Al-Assaf (2005)	4	Frequency features	MLP	97.22
Cheng and Ma (2008)	6	Raw data	PNN	95.58
Zan et al. (2010)	6	Autoregressive (AR) spectrum	Fuzzy ARTMAP	95.00
Ranaee and Ebrahimzadeh (2013)	6	Shape and statistical features	MLP	99.15
Kao et al. (2016)	7	Independent component analysis (ICA)	SVM	98.94
Zhou et al. (2018)	6	Shape and statistical features	FSVM	99.28
Addeh et al. (2018)	8	Shape and statistical features	Bees-RBF	99.63
This work	6	Raw data	1D-CNN	98.33
This work	6	Raw data—after removing dirty data from datasets	1D-CNN	99.30

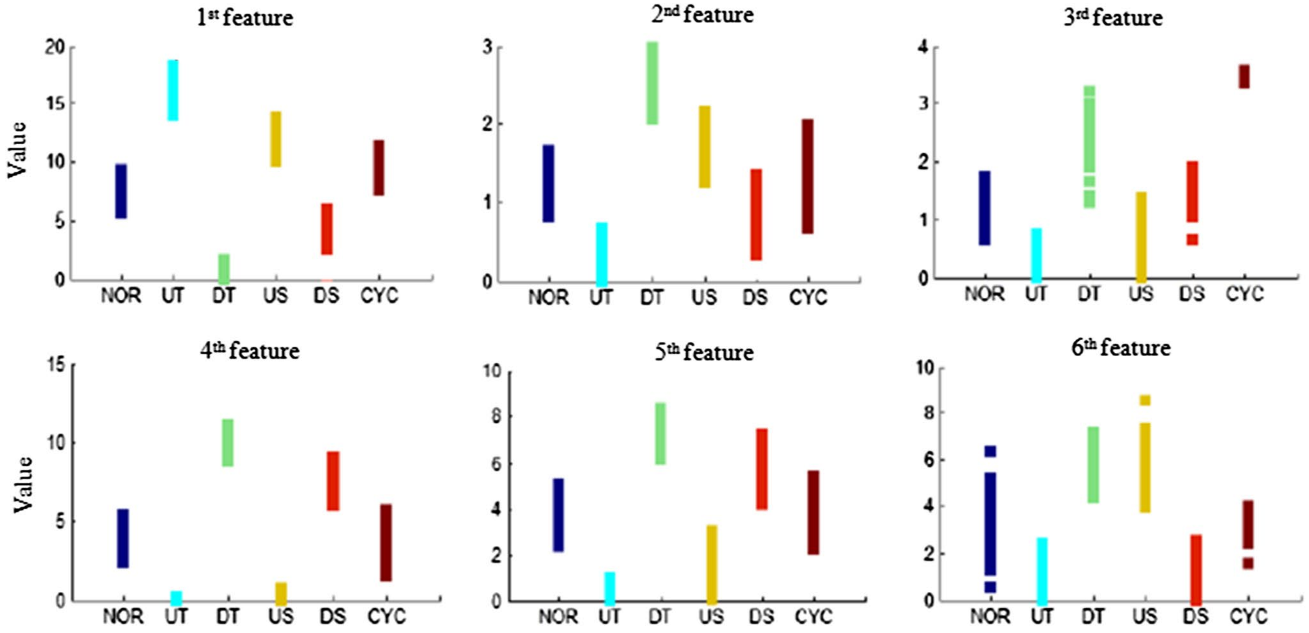
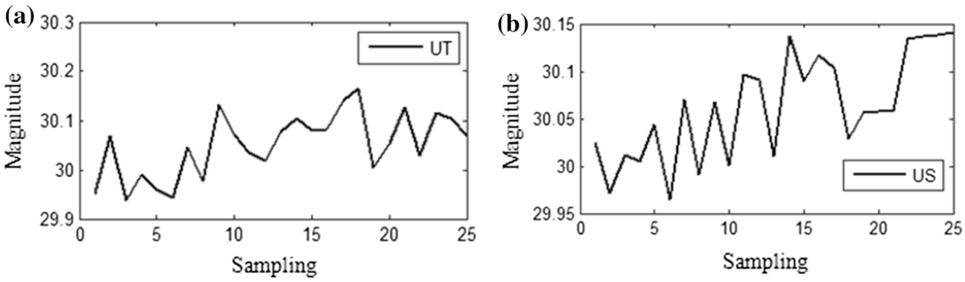


Fig. 8 Box-plots of some features for different CCP types

Fig. 9 Examples of dirty data in the datasets for **a** control chart with a UT label, and **b** control chart with a US label



Besides, in most literature, there were more sampling points per control chart sample, about 60 sampling points, which could ensure a larger difference between different CCP types; moreover, the trend patterns could become more evident with the time, helping the achievement of a high recognition rate. (In this case, the recognition rate of the proposed method was over 99.73%.) In this work, the number of sampling points per control chart sample was 25 to detect the abnormalities in the manufacturing process more timely and reduce the loss of manufacturing enterprises. However, the patterns could become very similar, especially trend pattern and shift pattern, which would make the recognition even more difficult.

Application in production environment

In order to further illustrate the proposed methodolog, a control chart diagnostic system including Nelson rules and 1D-CNN has been developed and applied to the monitoring of a real dataset from the production environment. The diameter of parts is regarded as the key quality characteristic, which can be measured with a three-coordinates measuring machine.

As shown in Fig. 10, in the analysis of control charts by Nelson rules, only a small number of second-type anomalies (9 continuous points on the same side of the central line) were found, and the process was basically stable and controlled. At the same time, the 1D-CNN method was implemented for a moving window on control charts. Like the number of sampling points in simulation experiments, the window size of CCPR was also set as 25. Unlike the results of Nelson rules method, the 1D-CNN can identify all 5 unnatural CCPs, as shown in Fig. 11.

Through communication with enterprise engineers, it was found that tool status and equipment exchange may lead to the emergence of these unnatural CCPs in the processing process. The 1D-CNN can accurately identify unnatural CCPs that Nelson rules can not recognize. This means that the 1D-CNN trained with simulation dataset still performed well in recognizing the real dataset from the production

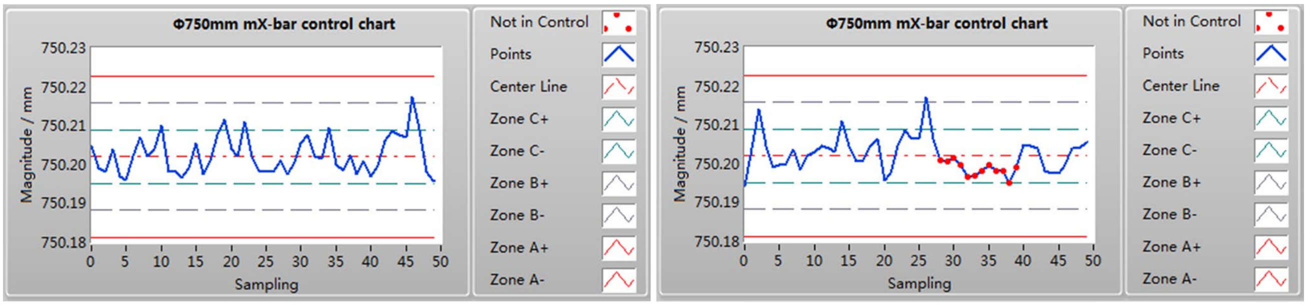


Fig. 10 CCPR result for real data based on Nelson rules

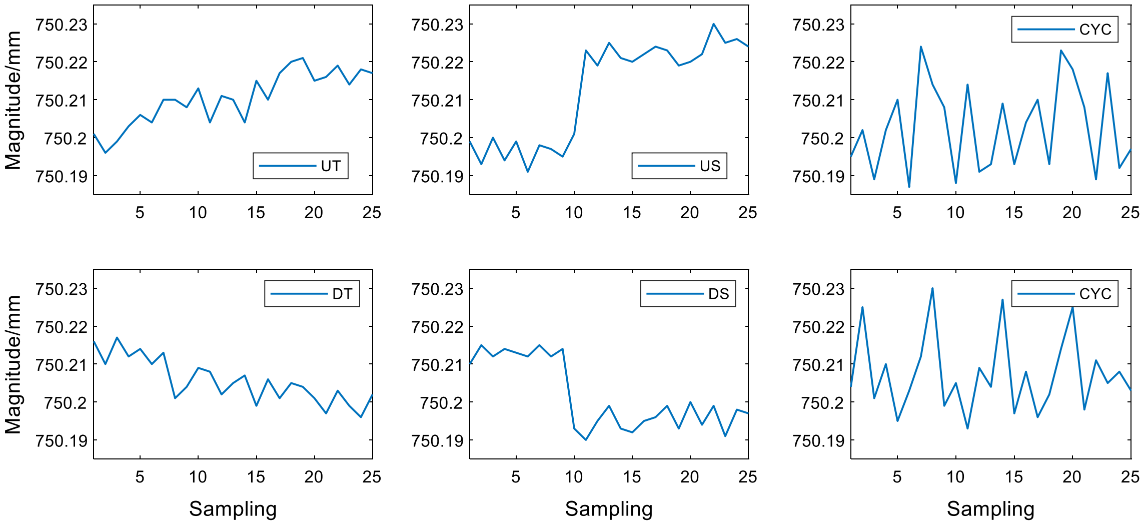


Fig. 11 Unnatural CCPs detected from real data by 1D-CNN

environment. In addition, Nelson rules and 1D-CNN recognition results are complementary, and the use of both can make the recognition system more perfect.

Discussion

The network structural parameters used in the comparative experiment are shown in Table 2. After many epochs of training, the final recognition rate varied a little, and the training time of each epoch differed slightly. Changing the number of feature maps, we found out that if their number was too small, the recognition rate decreased significantly, indicating that the number of features extracted by the 1D-CNN was insufficient. On the other hand, when the number exceeded a certain value, the difference in the recognition rate with the increase in parameter D was not obvious, which indicated that the number of features extracted by the 1D-CNN was enough to describe the raw data, but the training time increased significantly, decreasing the efficiency. Under various structural parameters, the convergence rate of the MSE was different, but the final convergence results were close. As shown in Fig. 5, test No. 1 converged at the 20th epoch but tests No. 9 and No. 12 converged at the 40th epoch. In the first convolutional layer, the convolutional kernel of a smaller size converged fast. To sum up, the structural parameters of the 1D-CNN had a little effect on the final recognition rate but had a significant impact on the training time. Therefore, to achieve higher recognition rate, the number of feature maps should be ensured to be enough.

According to the results given in Table 4 and Fig. 6, it is obvious that the 1D-CNN with ReLU and Softmax functions performed better than with the Sigmoid function. Both the CRR and the convergence speed improved significantly. Thus, the ReLU activation function was more suitable for the 1D-CNN because the gradient dissipation problem in deep learning could be effectively solved.

In Fig. 7, it is shown that there were different numbers of misclassification of the two recognition algorithms. However, the performance of the 1D-CNN was significantly better than that of the MLP. In the case of the MLP and raw data, the recognition results showed the tendency of misclassification the NOR patterns with the CYC patterns. At the same time, there was also a severe misclassification between the UT and US patterns, and between the DT and DS patterns. The misclassification of the NOR patterns with CYC patterns was about 10.8%, and of the CYC patterns with the NOR patterns about 18.8%. Such a result can be considered as a very bad result, which means that the Type I errors (false alarms) and Type II errors (missed disturbances) of this CCPR system were severe. In addition, the misclassification between the NOR and CYC patterns was effectively improved when the feature set was used as MLP input, which proved the effectiveness of the feature-based method, but the CRR was still unsatisfactory. In contrast, the NOR and CYC patterns were accurately identified by the 1D-CNN, and the CRR was high, it was 98.33% exactly. Other CCP types also had lower error rates. More importantly, there was no misclassification between the unnatural patterns and NOR patterns. In other words, there were no Type I and Type II errors in this CCPR system. The experimental results given in Table 5 show that the 1D-CNN performance decreased significantly when the number of samples was small compared with the traditional MLP method, which is typical for deep learning. However, the recognition rate of the 1D-CNN was still higher than that of the MLP. In addition, the iteration time of the proposed method was much smaller than that of the MLP, regardless of the number of samples. When the number of samples was large, the 1D-CNN still achieved good training efficiency.

In the CCPR process, extracting a large number of features improved the recognition accuracy of the CCPs, but it also increased the quality control workload. Moreover, the feature extracted is becoming more and more complicated. Although the features were specially designed by the experts for the CCPR, they could not be guaranteed to be the best ones. Also, the shape features were specially designed for the CCPR, which led to their poor versatility in other fields. In contrast, there were no such problems in the feature learning process. The raw data of the CCPs was continuously transformed by the convolution and pooling layers to get the feature set, and the network weights and biases were optimized by the learning algorithm to minimize the cost function. Although the features extracted by the neural network did not have any apparent physical meaning like the shape and statistical features did, it could be ensured that the feature set obtained by the neural network was the best for the recognition task. The results presented in Fig. 8 also prove this. Namely, the method based on the 1D-CNN and raw data achieved higher recognition rate than the method based on the MLP and traditional feature set due to the high-quality feature set obtained by the 1D-CNN. In addition, the raw data was fed directly to the network input, which not only reduced the workload but also provided strong generality in other fields.

Conclusion

In this paper, a 1D-CNN is proposed for feature learning and realization of an end-to-end CCPR. Based on the obtained results, the following conclusions can be drawn. (1) Although there were many network structural parameters, parameters selection was not complicated. When the number of feature maps was enough, good recognition accuracy was obtained regardless of the convolutional kernel size. (2) The ReLU activation function was more suitable for deep learning, and it significantly improved the 1D-CNN recognition accuracy and convergence speed. (3) The feature set learned by the proposed 1D-CNN was superior in quality over the one extracted manually. When the raw CCP data was used as an input, the patterns were correctly recognized by the proposed 1D-CNN with the recognition accuracy of 98.33%. The identification accuracy, convergence speed, and iteration time of the proposed model were significantly better than of the traditional MLP model.

In summary, the proposed 1D-CNN avoids the problem of extracting all kinds of complex features, making it more conducive to the practical quality control and helping to improve the automation and intelligence level of quality management in enterprises. In the future, an on-line prediction and diagnosis system of control chart based on a variety of deep learning algorithms will be studied.

Acknowledgements This study is supported by National Natural Science Foundation of China (No. 51575014), Science and Technology Project of Beijing Municipal Commission of Education (KM201410005026) and National Fund for Studying Abroad (201806545032). Special thanks to Dr. Jifeng Liang of Tongyu Heavy Industry Co., Ltd for providing production data.

References

Addeh, A., Khormali, A., & Golilarz, N. A. (2018). Control chart pattern recognition using RBF neural network with new training algorithm and practical features. *ISA Transactions*. https://doi.org/10.1016/j.isatra.2018.04.020.

Ajm, T., & Hulzebosch, A. A. (1996). Computer vision system for on-line sorting of pot plants using an artificial neural network classifier. *Computers and Electronics in Agriculture, 15*(1), 41–55.

Al-Assaf, Y. (2004). Recognition of control chart patterns using multiresolution wavelets analysis and neural networks. *Computers & Industrial Engineering, 47*(1), 17–29.

Assaleh, K., & Al-Assaf, Y. (2005). Features extraction and analysis for classifying causable patterns in control charts. *Computers & Industrial Engineering, 49*(1), 168–181.

Awadalla, M. H. A., & Sadek, M. A. (2012). Spiking neural network-based control chart pattern recognition. *Alexandria Engineering Journal, 51*(1), 27–35.

Bag, M. (2012). An expert system for control chart pattern recognition. *International Journal of Advanced Manufacturing Technology, 62*(1–4), 291–301.

Cheng, C. S. (1997). A neural network approach for the analysis of control chart patterns. *International Journal of Production Research, 35*(3), 667–697.

Cheng, C., & Hubele, N. F. (1992). Design of a knowledge based expert system for statistical process control. *Computers & Industrial Engineering, 22*(4), 501–517.

Cheng, Z., & Ma, Y. Z. (2008). A research about pattern recognition of control chart using probability neural network. In *ISECS international colloquium on computing, communication, control, & management*. IEEE.

Davis, R. B., & Woodall, W. H. (1988). Performance of the control chart trend rule under linear shift. *Journal of Quality Technology, 20*(4), 260–262.

Ducan, A. J. (1986). *Quality control and industrial statistics* (5th ed.). Homewood, IL: Richard D. Irwin.

Gauri, S. K. (2010). Control chart pattern recognition using feature-based learning vector quantization. *International Journal of Advanced Manufacturing Technology, 48*(9–12), 1061–1073.

Gauri, S. K., & Chakraborty, S. (2006). Feature-based recognition of control chart patterns. *Computers & Industrial Engineering, 51*(4), 726–742.

Gauri, S. K., & Chakraborty, S. (2009). Recognition of control chart patterns using improved selection of features. *Computers & Industrial Engineering, 56*(4), 1577–1588.

Guh, R. S. (2008). Real-time recognition of control chart patterns in autocorrelated processes using a learning vector quantization network-based approach. *International Journal of Production Research, 46*(14), 3959–3991.

Guh, R. S., & Tannock, J. (1999). A neural network approach to characterize pattern parameters in process control charts. *Journal of Intelligent Manufacturing, 10*(5), 449–462.

Hachicha, W., & Ghorbel, A. (2012). *A survey of control-chart pattern-recognition literature (1991–2010) based on a new conceptual classification scheme*. Oxford: Pergamon Press Inc.

Hassan, A., Baksh, M., Shaharoun, A. M., & Jamaluddin, H. (2003). Improved SPC chart pattern recognition using statistical features. *International Journal of Production Research, 41*(7), 1587–1603.

He, S., He, Z., & Wang, G. A. (2013). Online monitoring and fault identification of mean shifts in bivariate processes using decision tree learning techniques. *Journal of Intelligent Manufacturing, 24*(1), 25–34.

Janssens, O., Slavkovikj, V., Vervisch, B., Stockman, K., Loccufier, M., Verstockt, S., et al. (2016). Convolutional neural network based fault detection for rotating machinery. *Journal of Sound and Vibration, 377,* 331–345.

Jin, J., & Shi, J. (2001). Automatic feature extraction of waveform signals for in process diagnostic performance improvement. *Journal of Intelligent Manufacturing, 12*(3), 257–268.

Kao, L. J., Lee, T. S., & Lu, C. J. (2016). A multi-stage control chart pattern recognition scheme based on independent component analysis and support vector machine. *Journal of Intelligent Manufacturing, 27*(3), 653–664.

Kuo, T., & Mital, A. (1993). Quality control expert systems: A review of pertinent literature. *Journal of Intelligent Manufacturing, 4*(4), 245–257.

Liao, Y., Zeng, X., & Li, W. (2017). Wavelet transform based convolutional neural network for gearbox fault classification. In *Prognostics and system health management conference* (pp. 1–6). IEEE.

Miao, Z., & Yang, M. (2019). Control chart pattern recognition based on convolution neural network. In B. Panigrahi, M. Trivedi, K. Mishra, S. Tiwari, & P. Singh (Eds.), *Smart innovations in communication and computational sciences. Advances in intelligent systems and computing* (Vol. 670). Singapore: Springer.

Montgomery, D. C. (2007). *Introduction to statistical quality control*. London: Wiley.

Nelson, L. S. (1984). The Shewhart control chart: Test for special causes. *Journal of Quality Technology, 16*(4), 237–239.

Nelson, L. S. (1985). Interpreting Shewhart X-bar control charts. *Journal of Quality Technology, 17*(2), 114–116.

Pelegrina, G. D., Duarte, L. T., & Jutten, C. (2016). Blind source separation and feature extraction in concurrent control charts pattern recognition: Novel analyses and a comparison of different methods. *Computers & Industrial Engineering, 92,* 105–114.

Pham, D. T., & Oztemel, E. (1994). Control chart pattern recognition using learning vector quantization networks. *International Journal of Production Research, 32*(3), 721–729.

Pham, D. T., & Wani, M. A. (1997). Feature-based control chart pattern recognition. *International Journal of Production Research, 35*(7), 1875–1890.

Ranaee, V., & Ebrahimzadeh, A. (2011). Control chart pattern recognition using a novel hybrid intelligent method. *Applied Soft Computing Journal, 11*(2), 2676–2686.

Ranaee, V., & Ebrahimzadeh, A. (2013). Control chart pattern recognition using neural networks and efficient features: A comparative study. *Pattern Analysis and Applications, 16*(3), 321–332.

Roberts, S. W. (1959). Control chart tests based on geometric moving averages. *Technometrics, 1*(3), 239–244.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management, 45*(4), 427–437.

Swift, J. A. (1987). *Development of a knowledge-based expert system for control-chart pattern recognition and analysis*. Stillwater: Oklahoma State Universi.

Western Electric Company. (1958). *Statistical quality control handbook*. Indianapolis: Western Electric Co., Inc.

Xia, M., Li, T., Xu, L., Liu, L., & Silva, C. W. D. (2018). Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks. *IEEE/ASME Transactions on Mechatronics, 23*(1), 101–110. https://doi.org/10.1109/TMECH.2017.2728371.

Xie, Y., & Zhang, T. (2017). Fault diagnosis for rotating machinery based on convolutional neural network and empirical mode decomposition. *Shock and Vibration, 10,* 15. https://doi.org/10.1155/2017/3084197.

Yang, W. A., & Zhou, W. (2015). Autoregressive coefficient-invariant control chart pattern recognition in autocorrelated manufacturing processes using neural network ensemble. *Journal of Intelligent Manufacturing, 26*(6), 1161–1180.

Yang, W. A., Zhou, W., Liao, W., & Guo, Y. (2015). Identification and quantification of concurrent control chart patterns using extreme-point symmetric mode decomposition and extreme learning machines. *Neurocomputing, 147*(1), 260–270.

Zan, T., Wang, M., & Fei, R. Y. (2010). Pattern recognition for control charts using AR spectrum and fuzzy ARTMAP neural network. *Advanced Materials Research, 97–101,* 3696–3702.

Zhao, C., Wang, C., Hua, L., Liu, X., Zhang, Y., & Hu, H. (2017). Recognition of control chart pattern using improved supervised locally linear embedding and support vector machine. *Procedia Engineering, 174,* 281–288.

Zhou, X., Jiang, P., & Wang, X. (2018). Recognition of control chart patterns using fuzzy SVM with a hybrid kernel function. *Journal of Intelligent Manufacturing, 12,* 1–17.