mathematics

MDPI

Article

A Novel Scheme of Control Chart Patterns Recognition in Autocorrelated Processes

Cang Wu [1], Huijuan Hou [1], Chunli Lei [1], Pan Zhang [2] and Yongjun Du [1,*]

[1] School of Mechanical Engineering, Lanzhou University of Technology, Lanzhou 730050, China; wucang@lut.edu.cn (C.W.); houhuijuan@lut.edu.cn (H.H.); lyq0216@lut.edu.cn (C.L.)
[2] China School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China; panzhang@mail.nwpu.edu.cn
* Correspondence: yjdu@lut.edu.cn

Abstract: Control chart pattern recognition (CCPR) can quickly recognize anomalies in charts, making it an important tool for narrowing the search scope of abnormal causes. Most studies assume that the observations are normal, independent and identically distributed (NIID), while the assumption of independence cannot always be satisfied under continuous manufacturing processes. Recent research has considered the existence of autocorrelation, but the recognition rate is overestimated. In this paper, a novel scheme is proposed to recognize control chart patterns (CCPs) in which the inherent noise is autocorrelated. By assuming that the inherent noise follows a first-order autoregressive (AR(1)) process, the one-dimensional convolutional neural network (1DCNN) is applied for extracting features in the proposed scheme, while the grey-wolf-optimizer-based support vector machine (GWOSVM) is employed as a classifier. The simulation results reveal that the proposed scheme can effectively identify seven types of CCPs. The overall accuracy is 89.02% for all the autoregressive coefficients, and the highest accuracy is 99.43% when the autoregressive coefficient is on the interval $(-0.3, 0]$. Comparative experiments indicate that the proposed scheme has great potential to identify CCPs in autocorrelated processes.

Keywords: control chart patterns; autocorrelated processes; one-dimensional convolutional neural network; support vector machine

MSC: 62P30

check for updates

Citation: Wu, C.; Hou, H.; Lei, C.; Zhang, P.; Du, Y. A Novel Scheme of Control Chart Patterns Recognition in Autocorrelated Processes. *Mathematics* **2023**, *11*, 3589. https://doi.org/10.3390/math11163589

Academic Editor: Stelios Psarakis

Received: 6 July 2023
Revised: 15 August 2023
Accepted: 17 August 2023
Published: 19 August 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The control chart is an essential tool to monitor whether the manufacturing process is in control or not. There are several key practical efficiencies of using the control chart. First, it is an effective tool for improving the stability of manufacturing processes [1,2]. Second, it provides indicative information which can effectively prevent potential defects [3]. Third, the information on manufacturing system ability is provided [4].

As first proposed by Shewhart, a control chart is composed of two elements: three straight lines parallel to the horizontal axis and sampled observations in chronological order [5]. The traditional control chart is widely used, but there are still some limitations: (1) Analysis methods based on traditional judgment rules cannot point out all abnormal situations. (2) It only focuses on whether the observations are within limits or not, and it cannot provide potential information of previous observations. (3) It cannot provide more clues of abnormal situations, so it is difficult to find out assignable causes. With the development of statistical process control (SPC) and computer technology, the appearance of control chart patterns (CCPs) overcomes these limitations of the traditional control chart. The CCP is composed of continuous points that reflect fluctuations in manufacturing processes [6]. As such, the application of a CCP is exceptionally useful for a rapid diagnosis of any abnormal causal pattern and then for the formulation of a treatment scheme.

Fifteen types of CCPs are introduced in the Statistical Quality Control Handbook [7]. The normal pattern (NOR), cycle pattern (CYC), upward trend pattern (UT), downward trend pattern (DT), upward shift pattern (US), downward shift pattern (DS) and system pattern (SYS) are basic patterns, while the remaining types are combinations of a single pattern and some other special patterns. Since each type of abnormal pattern corresponds to a particular assignable cause [8], control chart pattern recognition (CCPR) dramatically reduces the search scope for abnormal causes. In practice, the type of CCP always corresponds to some specific assignable causes. Thus, CCPR can quickly recognize anomalies in charts, making it an important tool for narrowing the search scope of abnormal causes. The workload of inspectors will be significantly decreased. Developing the CCPR scheme is important for finding out the assignable cause.

In the past, most studies have concentrated on improving the recognition accuracy of CCPs. The rule-based expert system method was originally proposed by Alexander and Jagannathan, who proved that the CCP could be explained and analyzed according to artificial experience [9]. With this breakthrough, excessive false alarms emerged as a critical defect of the system. Subsequently, despite the abundance of research into these problems, recognition accuracy remained low [10]. With the development of computer technology, machine-learning algorithms have been employed in the CCPR field. For example, artificial neural networks [11], support vector machine (SVM) [12–14], random forest [15], decision tree [16,17], fuzzy systems [12] and other algorithms have performed well for CCPR. As a result, recognition accuracy improved dramatically so that a multilayer perceptual neural network identified six types of CCPs with an accuracy of 99.15% [18]. Ranaee, Ebrahimzadeh and Ghaderi [19] applied an improved particle swarm algorithm to optimize SVM, and the recognition accuracy is 99.58%. Kalteh and Babouei [20] suggested an adaptive neuro-fuzzy inference system recognition method based on the intelligent application of shape and statistical features. Meanwhile, a chaotic whale optimization algorithm was used to optimize every layer of the classifier, allowing the recognition accuracy to reach 99.77%.

In the literature, most of the existing research assumes that observations at different time points are normal, independent and identically distributed (NIID). Unfortunately, this assumption cannot satisfy flow industries [21], as for instance, there could be autocorrelation in chemical, pharmaceutical or metallurgical industries. Moreover, with the development of sensor technology, more and more advanced acquisition systems have been employed in production to collect process data [22]. Due to the acquisition method of high-frequency data, the time interval is extremely short between adjacent observations, and as a result, there is an inevitable autocorrelation [23].

If the NIID-based CCPR models are directly employed to monitor the autocorrelated process, there will be a great number of false alarms. At present, there are several studies considering autocorrelated processes in CCPR. Cheng and Cheng [24] used a ruled-based neural network as a classifier to identify five types of abnormal CCPs in autocorrelated processes and employed the Haar discrete wavelet transform for decorrelation and feature extraction. Lin, Guh and Shiue [25] proposed a CCPR model based on SVM for the online recognition of seven abnormal CCPs. The simulation results indicated that the method based on SVM had a better performance compared with the learning vector quantization (LVQ) network. In another study, Yang and Zhou [26] developed an integrated neural network based on the LVQ network. Every individual backpropagation network was trained to identify each pattern with the specific autoregressive coefficient, and the outputs of all backpropagation networks were combined by the LVQ network. In these papers, the process mean was assumed to be the first-order autoregressive (AR(1)) process, and the discrete autoregressive coefficients ($-0.9, \ldots, 0.9$) were selected to represent various autocorrelated levels. Shao et al. [27] proposed a two-stage framework to recognize seven kinds of both single and concurrent CCPs by assuming the autocorrelated processes, and the accurate identification rate is employed to evaluate the performance. De and Pham [21] first considered that the inherent noise was expressed by the AR(1) model and applied a

pattern generation scheme (PGS) to generate datasets. Then, the CCPR model based on the NIID assumption was used to recognize CCPs in autocorrelated processes. Compared with the NO-PGS and PGS datasets, it becomes clear that the former scheme has great recognition accuracy. However, given that a host of samples were discarded when PGS was applied to generate CCP samples, this recognition rate is overestimated.

The existing researches on CCPR have established the autocorrelated model in two ways: one assumes that the process mean follows an AR(1) model, and the other assumes that the inherent noise follows an AR(1) model. If both the process mean and the inherent noise obey the AR(1) model, the two models are equivalent when the process mean is set to zero. In other instances, when the process mean is described by the AR(1) model, there will be a remarkable deviation that can be monitored and recognized easily. For this reason, it is important to identify CCPs in tiny autocorrelated processes by assuming that the inherent noise follows AR(1).

In this paper, a novel scheme is proposed to recognize the control chart patterns in autocorrelated processes, in which the one-dimensional convolutional neural network (1DCNN) is utilized to extract features, and the grey-wolf-optimizer-based support vector machine (GWOSVM) is used as the classifier. The novel scheme provides a very appealing option for the CCPR at a wide range of autocorrelation levels and different types of patterns. Experiments show that the proposed scheme is comparable for some levels of autocorrelation and better for overall accuracy. The rest of this study is organized as follows. Section 2 introduces the basic concepts comprising the convolutional neural network (CNN) and SVM algorithms. Section 3 describes the proposed scheme, and Section 4 contains a series of experimental results and discussions. Finally, Section 5 presents the conclusion.

2. Basic Concepts

2.1. Convolutional Neural Network

CNN is a deep feedforward neural network, consisting of an input layer, a convolution layer, a pooling layer, a fully connected layer and an output layer. The convolution and pooling layers are mainly responsible for the feature extraction, selection and optimization. Compared with traditional machine learning, CNN's advantage is its automatic feature extraction capability [28].

The convolution layer is an essential layer, in which the convolution kernel is used to carry on the convolution operation. A convolution kernel can extract a feature map, so multiple convolution kernels can acquire multiple feature maps. The convolution operation is expressed as follows:

$$x_j^l = f\left(\sum_{i=1}^{G_{l-1}} x_i^{l-1} * k_{ij}^l + a_j^l\right) \tag{1}$$

where l is the sequential amount of the current network layer, f represents an activation function, G represents the number of feature maps, $*$ indicates the convolution operation, k_{ij}^l represents a convolution kernel, x_l^j is the jth output feature map and a_j^l represents the additive bias.

The most commonly used activation functions are the rectified linear unit function (ReLU), Sigmoid function and Hyperbolic Tangent function (Tanh). As discussed in Reference [28], the ReLU function is appropriate for the 1DCNN. This paper adopts a ReLU function given by the following:

$$f(x) = \max(0, x) \tag{2}$$

After the convolution operation, the size of the feature map also changes, which is calculated by the following:

$$H_l \times W_l = [(H_l - h)/m + 1] \times [(W_l - w)/m + 1] \tag{3}$$

where H_l and W_l represent the height and width, h represents the height of the convolutional kernel, w represents the width of the convolutional kernel, m represents the moving stride and m is set to 1 in this paper. Because 1DCNN is applied, W_l and w are also set to 1.

The pooling layer is simpler than the complex convolution layer, mainly decreasing the dimensions of features and improving calculation speed. There are two common pooling methods [26], max pooling and average pooling, and the latter is adopted in this paper. The average pooling operation tends to retain the overall information, expressed as follows:

$$x_j^l = down\left(x_j^{l-1}\right),\ j = 1, 2, 3, \cdots, G_l \tag{4}$$

where G_l indicates the number of input feature maps, and *down* is a pooling function.

After the pooling operation, the size of the output feature map also changes, which is calculated by the following:

$$H_l \times W_l = (H_{l-1}/n) \times (W_{l-1}/n) \tag{5}$$

where n represents the step size of the pooling operation and is set to 2 in this paper.

A fully connected layer can link the feature maps extracted through a series of convolution and pooling operations. The equation is given by the following:

$$\boldsymbol{x}_i = f(\boldsymbol{\omega}_0 \boldsymbol{f}_v + \boldsymbol{b}_0) \tag{6}$$

where $\boldsymbol{f}_v$ represents the input vector, and $\boldsymbol{b}_0$ and $\boldsymbol{\omega}_0$ are the bias vector and the weight matrix, respectively.

In the fully connected layer, M features are gained, expressed by the following:

$$M = H_{l-1} \times W_{l-1} \times G_{l-1} \tag{7}$$

where M represents the number of features, and G_{l-1} means the number of input feature maps of the lth fully connected layer.

In the training process, the backpropagation algorithm is utilized to optimize the weights and biases (k_{ij}^l, a_j^l, $\boldsymbol{\omega}_0$ and $\boldsymbol{b}_0$) to minimize the value of the loss function. The mean square error (MSE) is suitable for processing time series [29], and this paper chooses the MSE loss function, given as follows:

$$E(w, b) = \frac{\sum_{i=1}^{n} (y - y')^2}{n} \tag{8}$$

where y represents the real value, and y' is the predicted value.

2.2. Support Vector Machine

SVM is an important classification algorithm in machine learning with the advantages of simplicity, prediction and generalization [30]. Establishing an optimal hyperplane is the main idea, which maximizes the distance between the two classes of samples and the hyperplane [31]. The schematic diagram of the SVM is shown in Figure 1.

SVM is widely applied to solve binary classification problems. The process of binary classification is given as follows.

For a binary classification problem, suppose that the training samples are $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, where $x_i \in R^n$, $y_i \in \{+1, -1\}$. The samples are divided into two categories through a hyperplane according to the sample features. The expression of hyperplane is given by the following:

$$\omega^T x + b = 0 \tag{9}$$

where ω is the normal vector of the hyperplane, and b indicates a relative bias value.

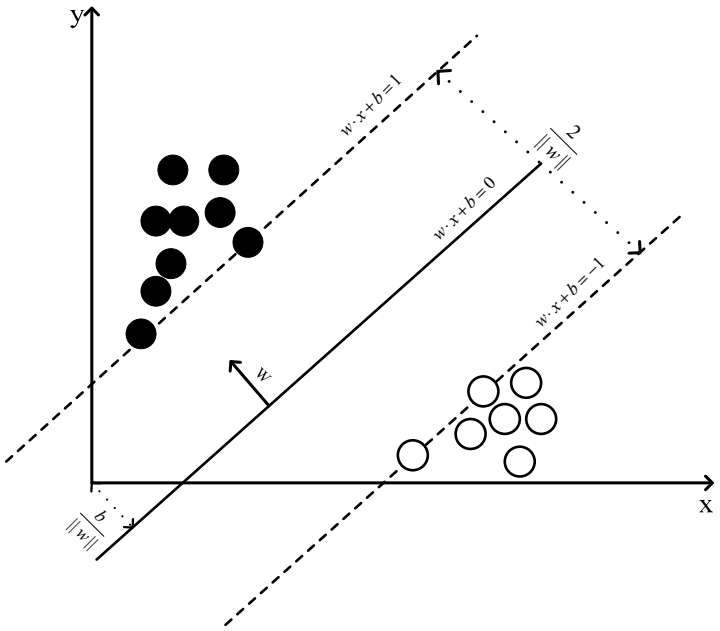

Figure 1. The schematic diagram of SVM.

For non-separable samples in linear space, a kernel function is used to map the input feature vector from the raw space to a higher-dimensional space. Common kernel functions include the Radial Basis Function (RBF), Laplace, Bessel and Tanh. This study chose to use the RBF, which is described by the following:

$$k(X, Y) = \exp(-\vartheta||X - Y||^2) \tag{10}$$

where the function $k(X, Y)$ is a non-negative monotonic function for any fixed $\vartheta > 0$.

The hyperplane of high-dimensional space is given by the following:

$$f(x) = \omega^T \varphi(x) + b = \sum_{i=1}^{m} \vartheta_i y_i k(X, Y) + b \tag{11}$$

The hyperplane vector that separates samples into two classes is called the "support vector" of a binary classification problem. To obtain the optimal classification hyperplane, the distance must be maximized between two types of support vectors and the hyperplane.

The distance can be expressed by the following:

$$\tau = \max \frac{2}{||\omega||} \text{ or } \tau = \min \frac{||\omega||^2}{2} \tag{12}$$

Additionally, a slack variable, ε_i, is introduced to obtain the SVM.

$$\min \frac{||\omega||^2}{2} + C \sum_{i=1}^{n} \varepsilon_i, \varepsilon_i \geq 0 \tag{13}$$

where C represents the penalty coefficient.

3. The Proposed 1DCNN-GWOSVM Scheme

A novel scheme to recognize CCPs is proposed for autocorrelated processes, and Figure 2 shows the flowchart. In detail, the 1DCNN is employed to extract features from autocorrelated CCP inputs, while the SVM optimized by the GWO algorithm is utilized as the classifier to recognize which type the input CCP belongs to. For the sake of simplicity,

this paper refers to the proposed scheme simply as the 1DCNN-GWOSVM scheme. The proposed scheme is implemented in three stages.

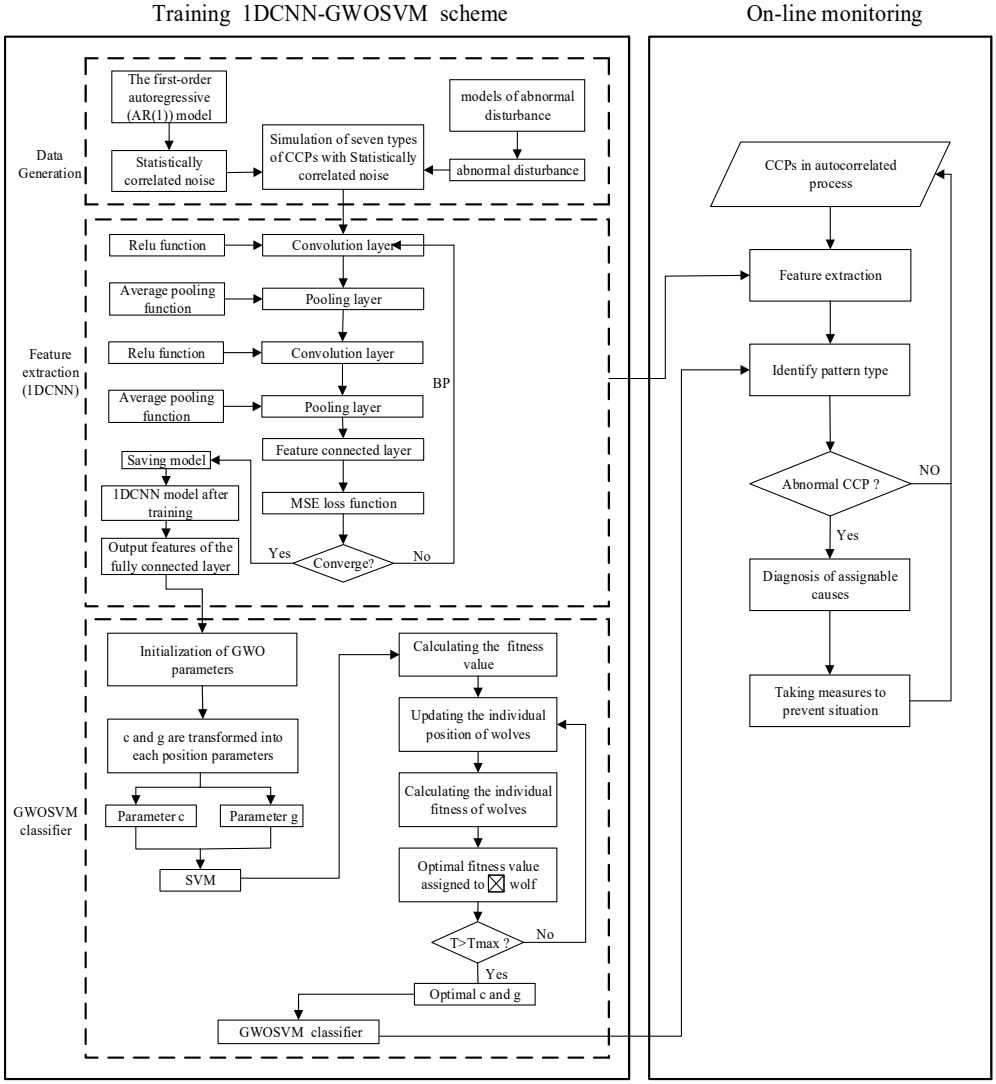

Figure 2. The flowchart of the 1DCNN-GWOSVM scheme.

Stage 1: Data Generation. As discussed in one review [32], due to the lack of fully documented public databases, 41 out of 44 papers evaluated the performance of the CCPR models by simulated data, while only 3 papers implemented real data in their studies. In this study, a large number of CCP samples are generated by the Monte-Carlo method, which is usually applied to generate CCPs both for training and testing. In this study, the process mean, inherent noise and abnormal disturbances components were utilized to generate the data points for the seven CCPs [21]. The specific expressions are given by the following:

$$Y_t = \mu + W_t + G_t \tag{14}$$

where Y_t represents the observation at time, t; μ indicates the process mean; W_t represents the inherent noise at time, t; and G_t represents an abnormal disturbance.

The inherent noise is assumed to follow the AR(1) model, which is given by the following:

$$W_t = \alpha W_{t-1} + \tau_t \tag{15}$$

where W_{t-1} indicates the inherent noise at time, $t-1$; α is autoregressive coefficient; τ_t is a white noise and obeys the standard normal distribution; and $\tau_t \sim N(0,1)$.

Based on Equations (1) and (2), the expressions of different typical CCPs are defined below. The NOR pattern is expressed by the following:

$$Y_t = \mu + W_t \tag{16}$$

The CYC pattern is expressed by the following:

$$Y_t = \mu + W_t + \gamma_1 \times \sin\left(\frac{2\pi t}{\gamma_2}\right) \tag{17}$$

where γ_1 and γ_2 represent the amplitude and period of a cycle, respectively.

The UT and DT patterns are expressed by the following:

$$Y_t = \mu + W_t \pm \gamma_3 \times t \tag{18}$$

where γ_3 is the slope of a trend, the UT pattern employs the mark "+" and the DT pattern applies the mark "−".

The US and DS patterns are expressed by the following:

$$Y_t = \mu + W_t \pm s \times \gamma_4 \tag{19}$$

where s represents the shift position, γ_4 indicates the shift magnitude, the US pattern generated employs the mark "+" and the DS pattern applies the mark "−".

The SYS pattern is expressed by the following:

$$Y_t = \mu + W_t + \gamma_5 \times (-1)^t \tag{20}$$

where γ_5 represents the systematic departure.

In this study, eight datasets at different autocorrelation levels were generated to validate the performance of the proposed 1DCNN-GWOSVM scheme.

Stage 2: Extract features. The CCPs generated from Stage 1 served as the inputs of the 1DCNN model. Two convolution layers and two pooling layers were alternatively used to select, extract and optimize the features of CCPs. The fully connected layer was used to expand and splice together the features that were extracted from the convolution and pooling layers. The ReLU activation function was used in the convolution layer, while the average pooling function was utilized in the pooling layer. In order to make the extracted features more prominent, the weights, biases and parameters of the convolution and pooling layers were optimized by the backpropagation algorithm. The extracted features were then output in the fully connected layer. The structure diagram of the 1DCNN feature extraction is shown in Figure 3.

Stage 3: Build the GWOSVM classifier. The SVM is employed as the classifier, and the features extracted from Stage 2 serve as the inputs. For the SVM classifier, the penalty term and the kernel function parameter g have a great impact on its classification performance. The GWO algorithm [33] is a new intelligence optimization algorithm inspired by the predatory activity of grey wolves, such as hunting, encircling and attacking their prey. Its advantages are its strong convergence, its few parameters and its readiness for implementation, accomplished through its simulation of social hierarchy and predatory behavior. Therefore, the parameters of SVM are optimized by the GWO algorithm to gain the optimal parameters, abbreviated as the GWOSVM classifier.

The proposed 1DCNN-GWOSVM scheme aims to recognize a CCP and identify its type in actual manufacturing processes. Identifying the type of abnormal CCP greatly narrows the search scope of the abnormal cause. The type of each CCP corresponds to possible causes [20,34], which are shown in Table 1.

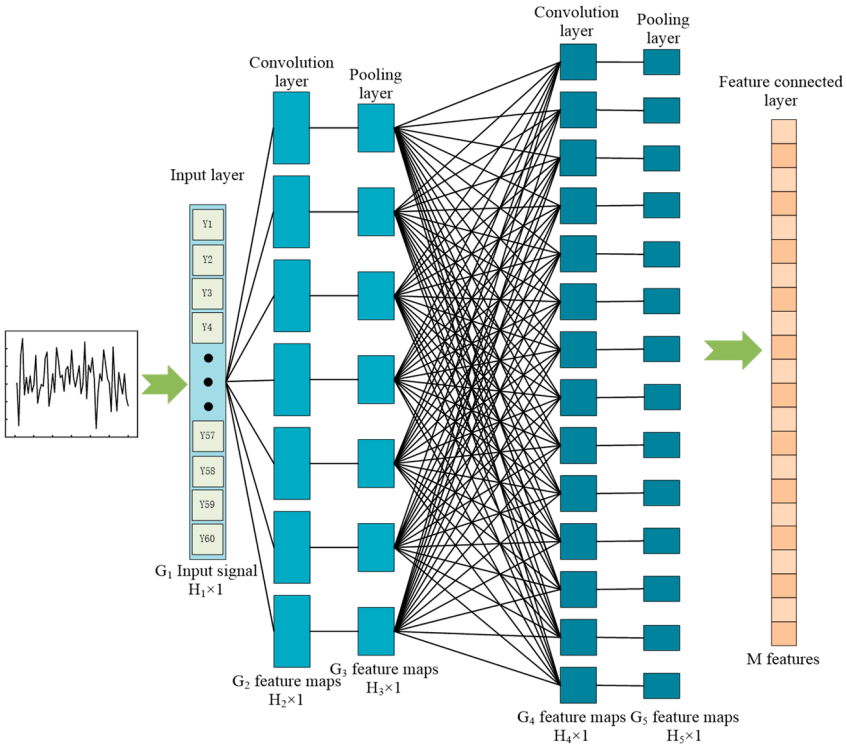

Figure 3. The structure diagram of the 1DCNN feature extraction.

Table 1. The type of CCPs and the possible causes.

The Type of CCPs	Possible Causes
CYC	The change of system environment, the fluctuation of power generation equipment and the rotation of operators periodically.
UT/DT	Fatigue operation, tools wear and equipment performance degradation.
US/DS	The failure of equipment parts, the transformations of raw materials, the machine setting and the introduction of new workers.
SYS	Difference between test sets; the output is checked in the rotation from different production lines.

4. Experimental Results and Discussions

To verify the availability and practicability of the proposed method, a series of simulation experiments are conducted in this section. In this study, the experimental data were simulated based on the Monte-Carlo method. All of the experiments were coded in MATLAB and implemented on a personal computer with a 2.50 GHz CPU and 8 GB memory. The package LibSVM in MATLAB was employed to classify the CCP patterns. The recognition accuracy of the test samples was used as the evaluation criterion, and the recognition accuracy of each pattern was intuitively represented by the confusion matrix.

4.1. Simulating CCP Samples in Autocorrelated Processes

According to the generation process and mathematical expression of CCPs outlined in Section 3, many samples are generated to validate the performance of the proposed scheme. This paper considers the CCP samples generated at different autocorrelation levels, including the autoregressive coefficients $\alpha \in (-1, -0.7]$, $\alpha \in (-0.7, -0.5]$, $\alpha \in (-0.5, -0.3]$, $\alpha \in (-0.3, 0]$, $\alpha \in (0, 0.3]$, $\alpha \in (0.3, 0.5]$, $\alpha \in (0.5, 0.7]$ and $\alpha \in (0.7, 1]$.

The actual manufacturing process is a complex system. To ensure the facticity of the simulated samples, the parameters of CCPs are set in a large range, as described in Reference [21]. Without the loss of generality, μ is set to 0. There is also no abnormal

disturbance in the NOR pattern. The parameters and equations of seven CCPs are given in Table 2, and assume the parameters follow uniform distribution. The shift point is randomly selected between 15 and 45, while the observation window length is 60. Figure 4 shows the seven types of CCPs. Training a 1DCNN model requires the use of many samples to avoid overfitting, and in this study, a total of 49,000 samples (42,000 for training samples; 7000 for testing samples) were generated to construct training and testing datasets.

Table 2. The parameters of seven CCPs.

Pattern	Window Length	Equations	Parameter Value
NOR	$N = 60$	$Y_t = \mu + W_t$	$\mu = 0$
CYC		$Y_t = \mu + W_t + \gamma_1 \times \sin\left(\frac{2\pi t}{\gamma_2}\right)$	$0.1 \leq \gamma_1 \leq 3.0$ $3 \leq \gamma_2 \leq 16$
UT		$Y_t = \mu + W_t + \gamma_3 \times t$	$0.01 \leq \gamma_3 \leq 0.3$
DT		$Y_t = \mu + W_t - \gamma_3 \times t$	$0.01 \leq \gamma_3 \leq 0.3$
US		$Y_t = \mu + W_t + s \times \gamma_4$	$0.1 \leq \gamma_4 \leq 3.0$
DS		$Y_t = \mu + W_t - s \times \gamma_4$	$0.1 \leq \gamma_4 \leq 3.0$
SYS		$Y_t = \mu + W_t + \gamma_5 \times (-1)^t$	$0.1 \leq \gamma_5 \leq 3.0$

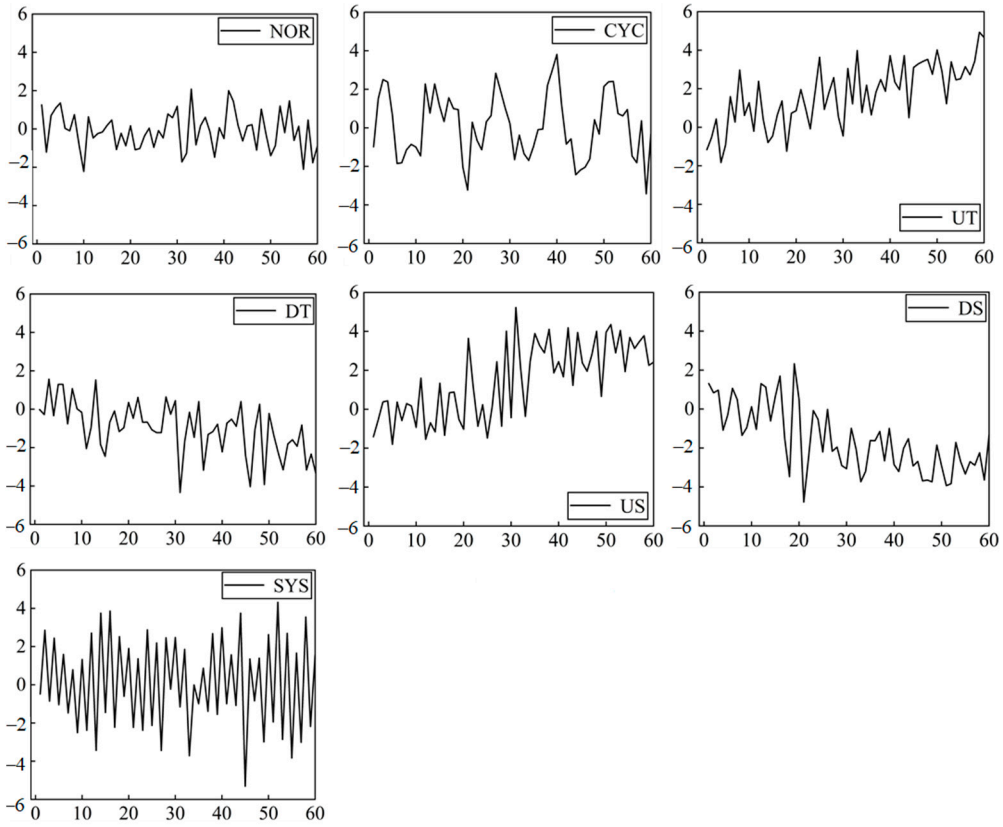

Figure 4. The seven types of CCPs.

4.2. The 1DCNN Feature Extraction

To ensure the availability of the 1DCNN feature extraction, it is necessary to train an integral 1DCNN. The 1DCNN structure adopted herein is similar to the classic LeNet5 model. We conducted a number of simulation experiments to determine the 1DCNN model's optimal structure parameters, which are given in Table 3.

Table 3. Structural parameters of the 1DCNN.

Parameters	Specific Value
Number and size of the input layer	$G_1 \times H_1 \times W_1 = 1 \times 60 \times 1$
Size of the first convolutional kernel	$h_2 \times w_2 = 11 \times 1$
Number and size of the first convolution layer feature map	$G_2 \times H_2 \times W_2 = 7 \times 50 \times 1$
Number and size of the first pooling layer subsample map	$G_3 \times H_3 \times W_3 = 7 \times 25 \times 1$
Size of the second convolutional kernel	$h_4 \times w_4 = 18 \times 1$
Number and size of the second convolution layer feature map	$G_4 \times H_4 \times W_4 = 14 \times 8 \times 1$
Number and size of the second pooling layer subsample map	$G_5 \times H_5 \times W_5 = 14 \times 4 \times 1$
Number of features	$M = 56$

Firstly, the input of 1DCNN is time-series data. Since the output of the univariate time series is also a univariate time series after convolution calculation, the first convolution layer applies seven convolution kernels, also known as filters, to extract features. The average pooling function is used to further extract the features and reduce the data complexity. Then, the convolution and pooling layers are used to further extract features. Finally, the fully connected layer is used to link the extracted features into one-dimensional feature vectors. In this paper, to show the effect of the 1DCNN feature extraction, we utilize the t-SNE algorithm to visualize raw data and extracted features, as shown in Figures 5 and 6. Comparing Figure 5 with Figure 6, it can be observed that the overlap degree of the seven CCPs is very high in Figure 5. Using the 1DCNN model for feature extraction is more conducive to the classification of CCPs, as the overlap degree of various CCPs has significantly decreased, as shown in Figure 6. It shows that the 1DCNN model performs well in the feature extraction.

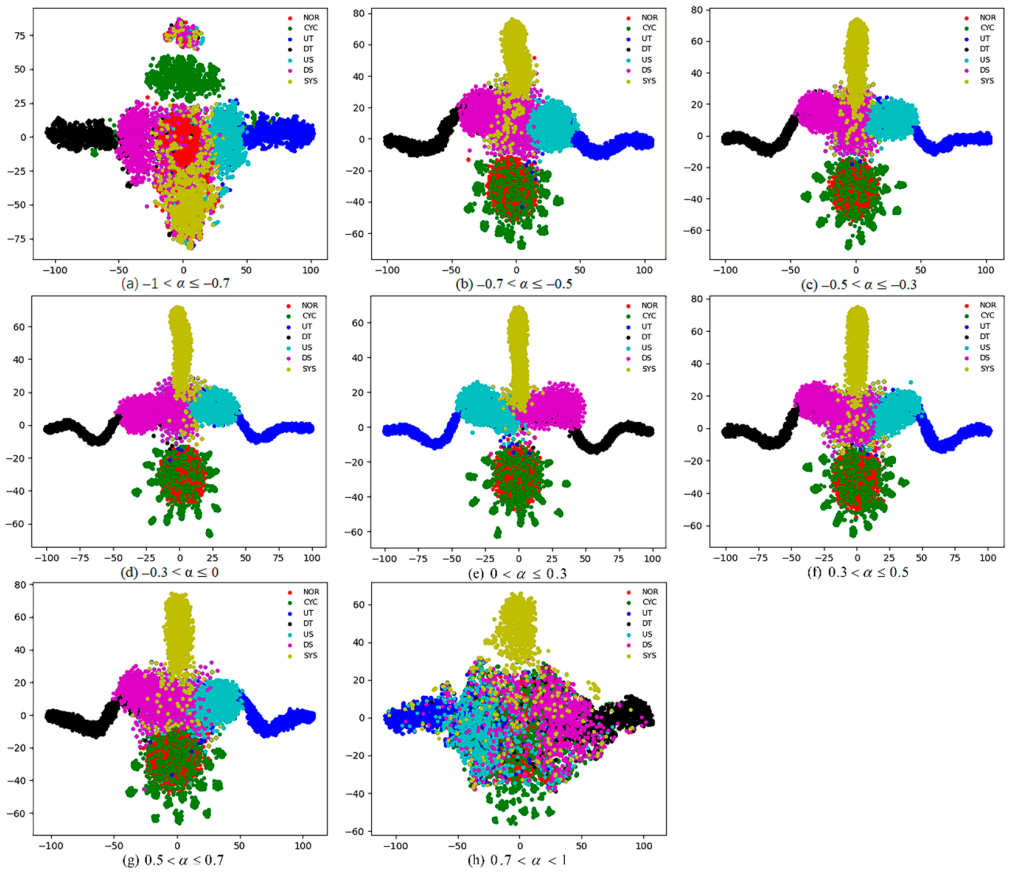

Figure 5. Visualization of the raw data at different α levels.

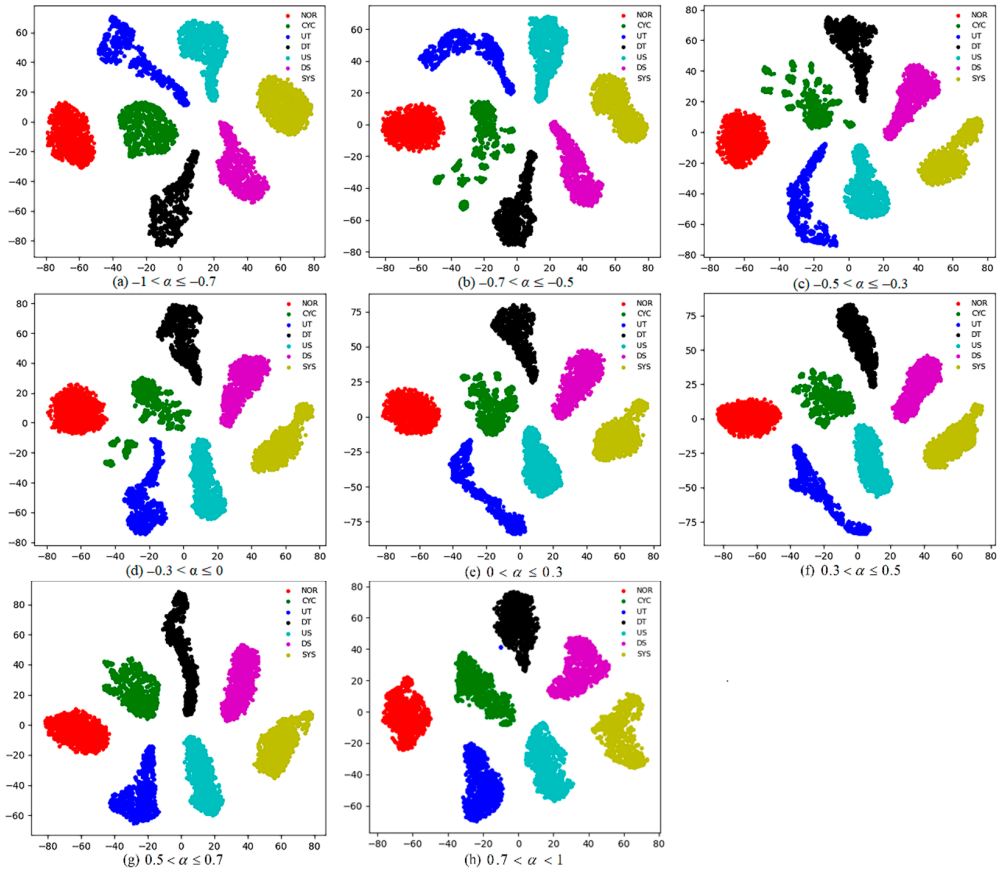

Figure 6. Visualization of the features at different α levels.

4.3. Training and Testing GWOSVM

After extracting features from the raw data, seven CCPs are classified by SVM. The extracted features are used to train and test the SVM classifier. Selecting reasonable parameters is helpful for the SVM classifier to enhance recognition accuracy, so this study used the GWO algorithm to select the optimal parameters.

Initially, 5600 feature vectors (800 samples for each pattern) were employed to train SVM. The GWO algorithm was applied to optimize the penalty factor and kernel function parameter, and to ultimately select the optimal value. The parameters of the GWO algorithm are given in Table 4. Then, 700 feature vectors (100 samples for each pattern) were applied to test the performance of the GWOSVM classifier. To obtain steady test accuracy, the tenfold cross-validation technique is used in this paper, with an average of 10 results used as the recognition accuracy.

Table 4. Parameter setting of the GWO optimization algorithm.

Algorithm	Group	Iteration	*c*	*g*
GWO	20	50	[0.01, 100]	[0.01, 100]

4.4. Results

The recognition performance of the proposed scheme was investigated at different α levels. The confusion matrix [35] was employed to intuitively express the recognition effect. Its diagonal values indicate the correct classification of diverse CCPs, and the other values indicate the percentage of the wrong classification. The confusion matrices of different α levels are shown in Figure 7.

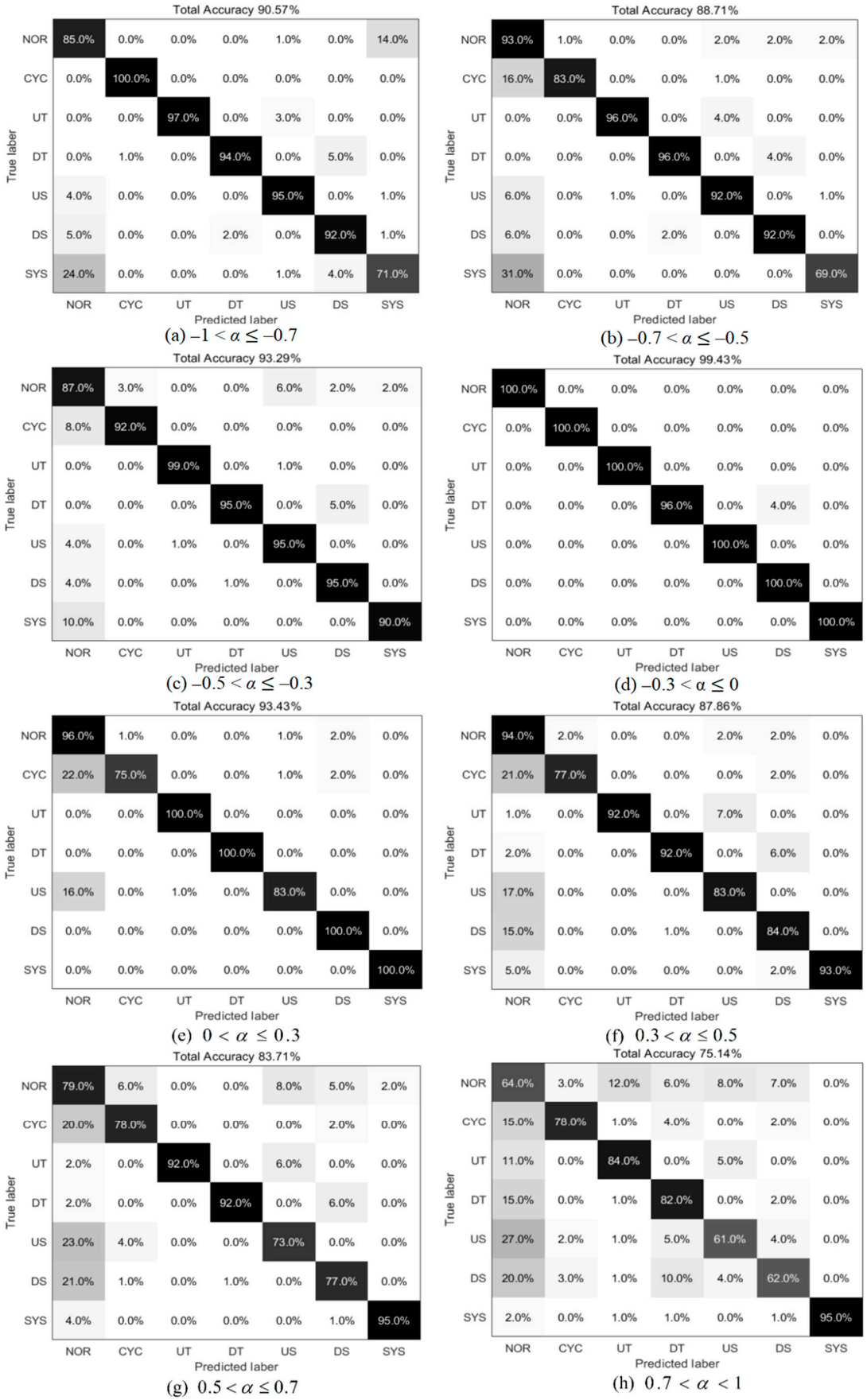

Figure 7. The confusion matrices of different α levels.

From Figure 7d, it can be found that the total accuracy is 99.43% at an autocorrelation level of $-0.3 < \alpha \leq 0$. The accuracy of the DT pattern is 96%, and the other 4% is wrongly divided into the DS pattern, with the recognition accuracy of the rest of the CCPs reaching 100%. Figure 7h shows that the lowest accuracy is 75.14% at a high autocorrelation level of $0.7 < \alpha < 1$. The US and DS patterns have the lowest accuracy.

To show the performance of the proposed scheme in terms of α levels, the overall recognition accuracy is present in Table 5, which reveals that the overall accuracy is 89.02% considering all φ levels. The UT and DT trend patterns have higher recognition accuracies of 95.25% and 92.88%, respectively. The US pattern yields the lowest recognition accuracy.

Table 5. The overall accuracy considering all the α levels.

CCP	Overall	NOR	CYC	UT	DT	US	DS	SYS
Accuracy	89.02%	87.24%	85.25%	95.25%	92.88%	85.25%	88.25%	89.38%

4.5. Discussions

The study on the CCPR in autocorrelated processes is worthwhile. Therefore, to verify the availability of the proposed method, five experiments were conducted. Table 6 gives some details of the input and classifier of six experiments in which statistical features (mean, standard deviation, skewness, kurtosis, mean-square value, median and range) [1] and shape features (SB, ACLPI, PSMLSC, AASL, REVE, SRANGE and ABDPE) [36,37] are utilized.

Table 6. Some details of input and classifier of six experiments.

No.	Input	Classifier
Experiment 1	Raw data	SVM
Experiment 2	7 statistical features	SVM
Experiment 3	7 statistical features and 7 shape features	SVM
Experiment 4	Features extracted by the 1DCNN	SVM
Experiment 5	Features extracted by the 1DCNN	Softmax
The proposed scheme	Features extracted by the 1DCNN	GWOSVM

Table 7 gives the recognition effects of the proposed scheme and other experiments at different α levels. Comparing Experiments 1–4, it can be found that different features have a remarkable influence on the recognition results. The features extracted by the 1DCNN can better express the features of CCPs, and this paper demonstrates the 1DCNN model's high performance regarding feature extraction. When comparing the results of Experiment 4, Experiment 5 and the proposed scheme, it can be proved that the GWOSVM classifier is better than the traditional Softmax and SVM classifiers.

Table 8 gives the overall accuracy of five experimental methods, the literature and the proposed scheme considering all the α levels. The results show that the recognition accuracy of the proposed scheme is 89.02%, which is a suboptimal performance likely caused by having different datasets in the respective training and testing models. As mentioned in De and Pham [21], datasets are generated by PGS and NO-PGS methods, with different datasets used to train and test the model. Because the PGS method discards at least a quarter of the initial patterns, the range of samples is artificially narrowed compared with the samples generated by NO-PGS. They show that 24.26%, 35.96% and 44.43% of the samples are discarded, while the corresponding recognition accuracies are 89.52%, 89.52% and 90.03%, respectively. When NO-PGS datasets are used to train and test the same model, the recognition accuracy is only 82.54%. This indicates that the difference is too significant to be ignored between the PGS and NO-PGS datasets. In this work, the NO-PGS datasets are used to train and test the 1DCNN-GWOSVM scheme, and the recognition accuracy is 89.02%, a recognition accuracy clearly higher than other methods trained and tested by

the NO-PGS datasets. Thus, the proposed scheme has more advantages in recognizing a diverse range of patterns.

Table 7. The recognition effects of the proposed scheme and other experiments at different α levels.

No. \ α	$-1 < \alpha \leq -0.7$	$-0.7 < \alpha \leq -0.5$	$-0.5 < \alpha \leq -0.3$	$-0.3 < \alpha \leq 0$
Experiment 1	61.57%	79.86%	83.14%	83.86%
Experiment 2	59.29%	72.71%	74%	75%
Experiment 3	68.29%	73.43%	79%	78.71%
Experiment 4	78.43%	82%	85.14%	85.29%
Experiment 5	88.47%	89.10%	89.57%	91.20%
The proposed scheme	90.57%	88.71%	93.29%	99.43%
	$0 < \alpha \leq 0.3$	$0.3 < \alpha \leq 0.5$	$0.5 < \alpha \leq 0.7$	$0.7 < \alpha < 1$
Experiment 1	84.86%	80.71%	73.71%	52.43%
Experiment 2	76.57%	70.14%	64.57%	46.86%
Experiment 3	79%	74.71%	58.86%	50.43%
Experiment 4	83.71%	80.57%	76.57%	67.57%
Experiment 5	88.37%	84.10%	82.66%	73.34%
The proposed scheme	93.43%	87.86%	83.71%	75.14%

Table 8. Comparison of the overall accuracy.

Comparison	Datasets	Overall Accuracy
Experiment 1	NO-PGS	75.02%
Experiment 2		67.39%
Experiment 3		70.30%
Experiment 4		79.91%
Experiment 5		85.85%
Reference [21]	PGS	90.03%
Reference [21]	NO-PGS	82.54%
The proposed scheme		89.02%

5. Conclusions

In this paper, a novel scheme based on the 1DCNN and GWOSVM was proposed to recognize seven types of CCPs in autocorrelated processes, in which the 1DCNN model was applied to extract important features, and GWOSVM was used as a classifier. The initial datasets were generated through the Monte-Carlo method, and then a series of experiments were conducted to prove the validity and correctness of the proposed scheme. The experiment results show that the overall accuracy is 89.02%, considering all the α levels, proving the proposed scheme's good performance. The comparison experiments indicate that the proposed method is superior to the other five methods and, in fact, has a wider generality and applicable range compared to the findings in the literature.

In summary, the following conclusions can be drawn:

(1) The 1DCNN is a more effective method to obtain optimal features than one in which features are extracted manually.
(2) The GWO algorithm is an effective method to enhance the recognition accuracy of the SVM classifier.
(3) In contrast to the Softmax classifier, the GWOSVM classifier performs well in terms of recognition accuracy.
(4) The scheme based on 1DCNN feature extraction and GWOSVM classification has a reliable recognition ability at different autocorrelation levels.
(5) Because this proposed scheme is limited to univariate statistics in autocorrelated processes, developing multivariate SPC is a direction for future work.

Author Contributions: Conceptualization, C.W. and H.H.; methodology, H.H. and C.L.; software, H.H.; validation, C.L., P.Z. and Y.D.; formal analysis, C.W.; investigation, Y.D.; data curation, P.Z.; writing—original draft preparation, C.W. and H.H.; writing—review and editing, C.L., P.Z. and Y.D.; visualization, H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the National Key Research and Development Plan under Grant No. 2020YFB1713600, Natural Science Foundation of Gansu Province under Grant 20JR5RA432 and the Program for Hongliu Excellent and Distinguished Young Scholars in Lanzhou University of Technology.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the editors and anonymous referees for their insightful comments that significantly improved this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hassan, A.; Baksh, M.S.N.; Shaharoun, A.M.; Jamaluddin, H. Improved SPC Chart Pattern Recognition Using Statistical Features. *Int. J. Prod. Res.* **2003**, *41*, 1587–1603. [CrossRef]
2. Kim, J.; Abdella, G.M.; Kim, S.; Al-Khalifa, K.N.; Hamouda, A.M. Control Charts for Variability Monitoring in High-Dimensional Processes. *Comput. Ind. Eng.* **2019**, *130*, 309–316. [CrossRef]
3. Peres, F.A.P.; Fogliatto, F.S. Variable Selection Methods in Multivariate Statistical Process Control: A Systematic Literature Review. *Comput. Ind. Eng.* **2018**, *115*, 603–619. [CrossRef]
4. De la Torre-Gutierrez, H.; Pham, D. A Control Chart Pattern Recognition System for Feedback-Control Processes. *Expert Syst. Appl.* **2019**, *138*, 112826. [CrossRef]
5. Montgomery, D.C. *Introduction to Statistical Quality Control*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
6. Wu, C.; Liu, F.; Zhu, B. Control Chart Pattern Recognition Using an Integrated Model Based on Binary-Tree Support Vector Machine. *Int. J. Prod. Res.* **2015**, *53*, 2026–2040. [CrossRef]
7. Western Electric Company. *Statistical Quality Control Handbook*, 2nd ed.; Western Electric Company: Indianapolis, IN, USA, 1956.
8. Cheng, C.-S.; Huang, K.-K.; Chen, P.-W. Recognition of Control Chart Patterns Using a Neural Network-Based Pattern Recognizer with Features Extracted from Correlation Analysis. *Pattern Anal. Appl.* **2015**, *18*, 75–86. [CrossRef]
9. Alexander, S.M.; Jagannathan, V. Advisory System for Control Chart Selection. *Comput. Ind. Eng.* **1986**, *10*, 171–177. [CrossRef]
10. Bag, M.; Gauri, S.K.; Chakraborty, S. An Expert System for Control Chart Pattern Recognition. *Int. J. Adv. Manuf. Technol.* **2012**, *62*, 291–301. [CrossRef]
11. Al-Assaf, Y. Recognition of Control Chart Patterns Using Multi-Resolution Wavelets Analysis and Neural Networks. *Comput. Ind. Eng.* **2004**, *47*, 17–29. [CrossRef]
12. Khormali, A.; Addeh, J. A Novel Approach for Recognition of Control Chart Patterns: Type-2 Fuzzy Clustering Optimized Support Vector Machine. *ISA Trans.* **2016**, *63*, 256–264. [CrossRef]
13. Zhang, M.; Yuan, Y.; Wang, R.; Cheng, W. Recognition of Mixture Control Chart Patterns Based on Fusion Feature Reduction and Fireworks Algorithm-Optimized MSVM. *Pattern Anal. Appl.* **2020**, *23*, 15–26. [CrossRef]
14. Chinas-Sanchez, P.; Lopez-Juarez, I.; Vazquez-Lopez, J.A.; El Kamel, A.; Navarro-Gonzalez, J.L. Stable and Unstable Pattern Recognition Using D-2 and SVM: A Multivariate Approach. *Mathematics* **2021**, *9*, 10. [CrossRef]
15. Bo, Z.; Beibei, L.; Yuwei, W.; Shengran, Z. Recognition of Control Chart Patterns in Auto-Correlated Process Based on Random Forest. In Proceedings of the 2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering, Hsinchu, Singapore, 8–9 February 2018; pp. 53–57.
16. Guh, R.-S. Real-Time Pattern Recognition in Statistical Process Control: A Hybrid Neural Network/Decision Tree-Based Approach. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2005**, *219*, 283–298. [CrossRef]
17. Wang, C.-H.; Guo, R.-S.; Chiang, M.-H.; Wong, J.-Y. Decision Tree Based Control Chart Pattern Recognition. *Int. J. Prod. Res.* **2008**, *46*, 4889–4901. [CrossRef]
18. Ranaee, V.; Ebrahimzadeh, A. Control Chart Pattern Recognition Using Neural Networks and Efficient Features: A Comparative Study. *Pattern Anal. Appl.* **2013**, *16*, 321–332. [CrossRef]
19. Ranaee, V.; Ebrahimzadeh, A.; Ghaderi, R. Application of the PSO-SVM Model for Recognition of Control Chart Patterns. *ISA Trans.* **2010**, *49*, 577–586. [CrossRef]
20. Aziz kalteh, A.; Babouei, S. Control Chart Patterns Recognition Using ANFIS with New Training Algorithm and Intelligent Utilization of Shape and Statistical Features. *ISA Trans.* **2020**, *102*, 12–22. [CrossRef]
21. De la Torre Gutierrez, H.; Pham, D.T. Identification of Patterns in Control Charts for Processes with Statistically Correlated Noise. *Int. J. Prod. Res.* **2018**, *56*, 1504–1520. [CrossRef]
22. Chiu, C.C.; Chen, M.-K.; Lee, K.-M. Shifts Recognition in Correlated Process Data Using a Neural Network. *Int. J. Syst. Sci.* **2001**, *32*, 137–143. [CrossRef]

23. Psarakis, S.; Papaleonida, G.E.A. SPC Procedures for Monitoring Autocorrelated Processes. *Qual. Technol. Quant. Manag.* **2007**, *4*, 501–540. [CrossRef]
24. Cheng, H.-P.; Cheng, C.-S. Denoising and Feature Extraction for Control Chart Pattern Recognition in Autocorrelated Processes. *Int. J. Signal Imaging Syst. Eng.* **2008**, *1*, 115–126. [CrossRef]
25. Lin, S.-Y.; Guh, R.-S.; Shiue, Y.-R. Effective Recognition of Control Chart Patterns in Autocorrelated Data Using a Support Vector Machine Based Approach. *Comput. Ind. Eng.* **2011**, *61*, 1123–1134. [CrossRef]
26. Yang, W.-A.; Zhou, W. Autoregressive Coefficient-Invariant Control Chart Pattern Recognition in Autocorrelated Manufacturing Processes Using Neural Network Ensemble. *J. Intell. Manuf.* **2015**, *26*, 1161–1180. [CrossRef]
27. Shao, Y.E.; Chang, P.-Y.; Lu, C.-J. Applying Two-Stage Neural Network Based Classifiers to the Identification of Mixture Control Chart Patterns for an SPC-EPC Process. *Complexity* **2017**, *2017*, 2323082. [CrossRef]
28. Zan, T.; Liu, Z.; Wang, H.; Wang, M.; Gao, X. Control Chart Pattern Recognition Using the Convolutional Neural Network. *J. Intell. Manuf.* **2020**, *31*, 703–716. [CrossRef]
29. Zhao, B.; Lu, H.; Chen, S.; Liu, J.; Wu, D. Convolutional Neural Networks for Time Series Classification. *J. Syst. Eng. Electron.* **2017**, *28*, 162–169. [CrossRef]
30. Fadel, S.; Ghoniemy, S.; Abdallah, M.; Abu, H. Investigating the Effect of Different Kernel Functions on the Performance of SVM for Recognizing Arabic Characters. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 446–450. [CrossRef]
31. Madzarov, G.; Gjorgjevikj, D.; Chorbev, I. A Multi-Class SVM Classifier Utilizing Binary Decision Tree. *Informatica* **2009**, *33*, 233–242.
32. Garcia, E.; Penabaena-Niebles, R.; Jubiz-Diaz, M.; Perez-Tafur, A. Concurrent Control Chart Pattern Recognition: A Systematic Review. *Mathematics* **2022**, *10*, 934. [CrossRef]
33. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
34. Lesany, S.A.; Fatemi Ghomi, S.M.T.; Koochakzadeh, A. Development of Fitted Line and Fitted Cosine Curve for Recognition and Analysis of Unnatural Patterns in Process Control Charts. *Pattern Anal. Appl.* **2019**, *22*, 747–765. [CrossRef]
35. Sokolova, M.; Lapalme, G. A Systematic Analysis of Performance Measures for Classification Tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
36. Gauri, S.K. Control Chart Pattern Recognition Using Feature-Based Learning Vector Quantization. *Int. J. Adv. Manuf. Technol.* **2010**, *48*, 1061–1073. [CrossRef]
37. Pham, D.T.; Wani, M.A. Feature-Based Control Chart Pattern Recognition. *Int. J. Prod. Res.* **1997**, *35*, 1875–1890. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.