



Recognition of abnormal patterns in industrial processes with variable window size via convolutional neural networks and AdaBoost

Ahmed Maged^{1,3} · Min Xie^{1,2}

Received: 22 July 2021 / Accepted: 27 December 2021 / Published online: 20 January 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

In industrial settings, it is inevitable to encounter abnormal patterns monitoring a process. These patterns point out manufacturing faults that can lead to significant internal and external failure costs unless treated promptly. Thus, detecting such abnormalities is of utmost importance. Machine learning algorithms have been widely applied to this problem. Nevertheless, the existing control chart pattern recognition (CCPR) method can only deal with a fixed input size rather than dealing with different input sizes according to the actual production needs. In order to tackle this problem, an original CCPR method relying on convolutional neural network (CNN) named as VIS-CNN is proposed. Signal resizing is performed using resampling methods, then CNN is used to extract the abnormal patterns in the dataset. Five different input sizes are generated for model training and testing. The optimal hyperparameters, as well as the best structure of the used CNN are obtained using Bayesian Optimization. Simulation results show that the correct recognition rate of the VIS-CNN is 99.78%, based on different window size control charts. Furthermore, we address the issue of the mixed CCP and provide a modified scheme to achieve high recognition ratio for 8 mixed patterns on top of 6 standard patterns. The modified scheme includes wavelet noise reduction and Adaptive Boosting. A case study on metal galvanization process is presented to show that the method has potential applications in the industrial environment.

Keywords CCPR · Mixed CCP · CNN · Deep learning · Bayesian optimization · Control chart pattern recognition

Introduction

In manufacturing, a process is considered in-control if only random causes affect its operation. On the other hand, an unstable process encounters other sources of variation, recognized as abnormal variations. The two mentioned sorts of variations can be differentiated using various monitoring tools (e.g., adaptive control system and control chart). Once some points surpass a threshold or an unnatural pattern is detected, the monitored manufacturing process is declared to

be out of control. When dealing with continuous variables, the abnormal control chart patterns (CCPs) can be linked with unambiguous causes that unfavorably disturb the manufacturing processes and point to specific machine problems. For example, trend patterns might indicate wear, thermal distortion of crucial parts of a machine tool, or operator fatigue (Addeh et al., 2018). In comparison, shift patterns might occur due to changes in operators, materials, or equipment. Moreover, periodic variation in the power supply could lead to cyclic patterns (Hachicha & Ghorbel, 2012).

While finding the out-of-control points can be recognized with no trouble by the quality practitioners, identifying the unnatural patterns depends on the experience level of quality control personnel. To detect unnatural patterns, many scholars have proposed supplementary runs-rules (Mehmood et al., 2019; Shongwe, 2020; Zhang et al., 2017). Nevertheless, Zan et al. (2020b) indicated that there is no one-to-one mapping relation between supplementary rules and abnormal patterns. Additionally, employing too many rules may not be applicable for real-time monitoring and will inevitably cause multiple false alarms (Ranaee & Ebrahimzadeh, 2011). On

✉ Ahmed Maged
amaged2-c@my.cityu.edu.hk

Min Xie
minxie@cityu.edu.hk

¹ Department of Advanced Design and Systems Engineering, City University of Hong Kong, Kowloon, Hong Kong

² Shenzhen Research Institute, City University of Hong Kong, Shenzhen, China

³ Department of Mechanical Engineering, Benha Faculty of Engineering, Benha University, Benha, Egypt

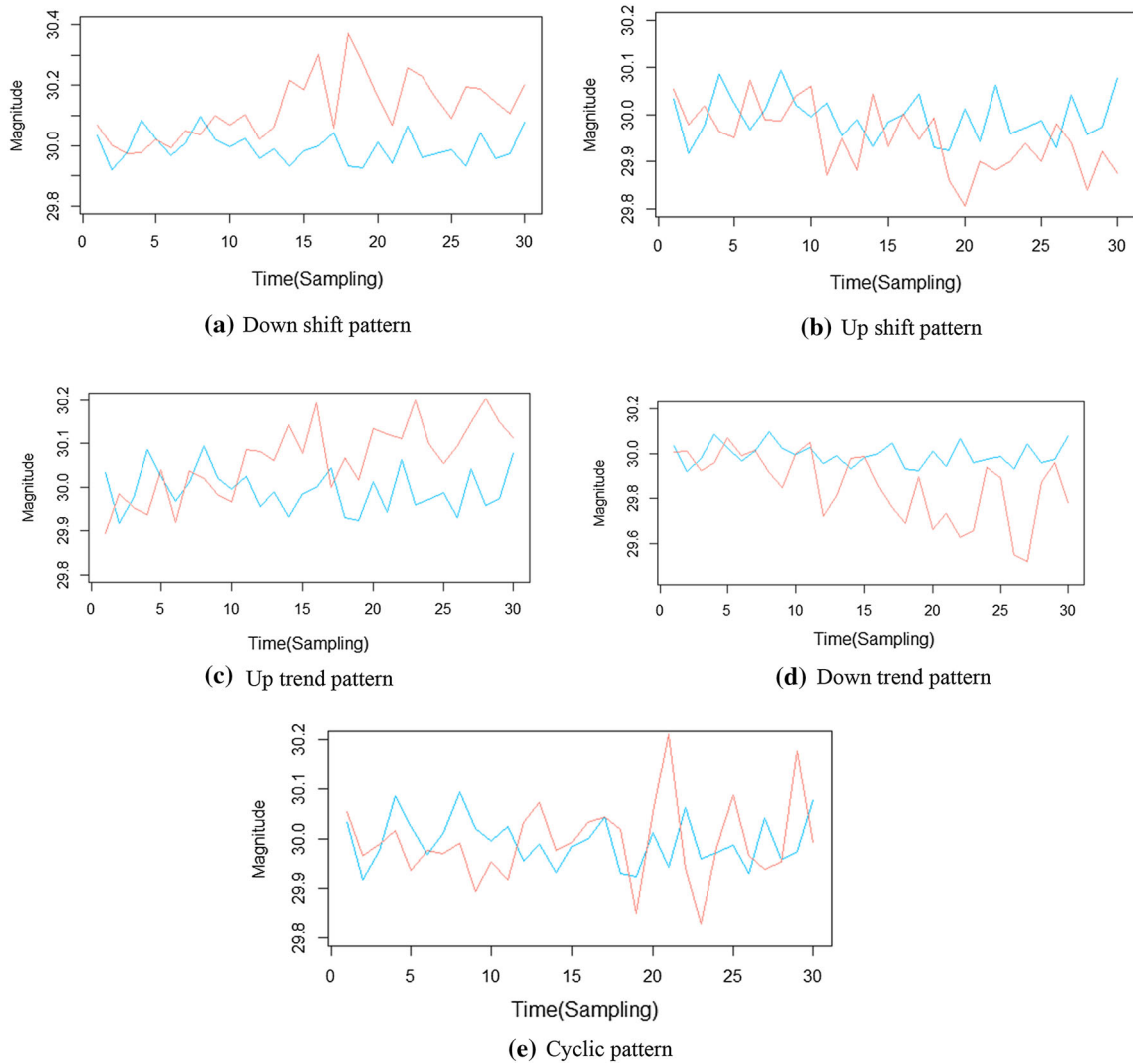


Fig. 1 Illustration of the normal (blue) and abnormal control chart patterns (red) (Color figure online)

Table 1 Parameters for generating the CCPs

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

the other hand, the presence of abnormal patterns can be extended to other applications beyond manufacturing; they can appear, for instance, as outbreaks in healthcare (Santiago & Smith, 2013) or variances in a supply chain (Costantino et al., 2015). Owing to the deficiencies of supplementary rules and continuously increasing requirements for intelligent manufacturing, more and more interest is noticed in

evolving accurate and automatic CCP Recognition (CCPR) algorithms. Hence, the CCPR problem is formulated as a pattern recognition problem and tackled by various machine learning models to be used. Hachicha and Ghorbel (2012) presented a systematic literature review on CCPR studies until 2010. They discussed about 120 research papers pub-

lished in well-recognized journals. In their review, most of the articles were based on artificial neural network (ANN).

Conversely, recently, more advanced methods with complicated frameworks have been adopted in order to improve the classification accuracy and the applicability of CCPR. Support Vector Machines (SVM) has been widely applied in the CCPR, achieving good results. Khormali and Addeh (2016) used a multiclass SVM-based classifier. In the presented method, type-2 fuzzy *c*-means (T2FCM) clustering algorithm was used to enhance the SVM system. Moreover, they used cuckoo optimization algorithm (COA) to optimize the hyperparameters of the classifier. Zhou et al. (2018) proposed a classification scheme of Fuzzy Support Vector Machine (FSVM) with hybrid kernel function. Meanwhile, the hyperparameters were optimized using genetic algorithm (GA). The results of the practical cases manifest that the proposed method has application potential for solving the problem of control chart interpretation in the real situations. Zaman and Hassan (2019) presented a hybrid method using a combination of fuzzy *c*-mean (FCM) and ANFIS. The comparison showed that the FCM–ANFIS method achieves comparable classification accuracy for eight types of investigated CCPs.

With the popularity of deep neural networks (DNNs), numerous academics have started to adopt it for CCPR. Miao and Yang (2019) were the first to use convolutional neural network (CNN) for CCPR based on extracted statistical and shape features from raw data. Dissimilar to traditional neural networks, the removal of full-connects between layers significantly decreases the CNN algorithms' complexity (Khan et al., 2017). Later, Zan and et al. (2020b) showed that CNN could learn optimal features automatically with no need to apply complex features extraction. Furthermore, the 1 Dimension CNN (1D-CNN) did well when a real dataset from production environment was used on top of the improved automation of the quality control in the process. Lu et al. (2020) extended the idea of CCPR to monitor machining conditions instead of output quality. Experiments were conducted for two machining processes with different cutting parameters. The outcomes revealed the applicability of CCPR for condition monitoring for the machining process under different machining environments. Fuqua and Razzaghi (2020) discussed the issue of imbalanced data in the CCPR problem. They developed a Cost-Sensitive CNN (CSCNN) for the imbalanced CCPR problem. Zan and et al. (2020a) used multilayer Bi-LSTM for CCPR and a similar problem named histogram pattern recognition (HPR), where seven histogram patterns are used for anomaly detection of the process.

One problem that is usually encountered when monitoring any process in order to detect abnormal control chart patterns

is the lack of ability to deal with the changing production rates. For the sake of clarification, consider a situation where the output of the production line is changing every hour or every shift while the network is trained to deal with fixed input size, let say 60 units/minute. This means that if the throughput yield of the production line is 40, it cannot be processed by the network. For the currently available methods, the network has to be retrained in order to change the input size, which will be inconvenient in many cases. To the best of our knowledge, all research works have been conducted using CNNs considered detecting CCPs for input signals with fixed length. Moreover, a common problem appears when more than one pattern exists at the same time in a problem referred to as mixed CCPs. Little research has been done on the mixed CCPs, and to the best of our knowledge, no one has considered all the variations and combinations between all the patterns that could potentially exist (8 mixed patterns) on top of the variable input size that we consider here. The different variations and combinations can be very similar in shape, and previously developed models can be easily confused between them. On the other hand, most of the available literature found on DNNs for CCPR determines the optimal hyperparameters as well as the network structure based on trial and error experiments with no proof of the optimality of the output.

In this paper, we aim to address the abovementioned three challenges in one framework based on CNNs. The contributions that we present in this work are as follows; first, we propose a CNN with Variable Input Size (VIS-CNN) for the CCPR problem in order to be able to monitor a signal with variable length. To the best of our knowledge, this is the first time VIS-CNN has been designed for the CCPR problem. Second, a modified VIS-CNN is presented in order to achieve high efficiency when dealing with mixed CCPs. Third, since hand-tuning is an arduous procedure to reproduce as it is based on trial and error, we optimize the hyperparameters as well as the architecture of our proposed VIS-CNN model using Bayesian optimization (BO). The paper is structured as follows; section “[Methodology](#)” describes the simulation process, the standard CNN, our proposed VIS-CNN algorithm, and the performance evaluation. Section “[VIS-CNN for standard CCP](#)” presents computational results and comparisons for the VIS-CNN, while section “[VIS-CNN for mixed CCP](#)” shows the modified VIS-CNN for mixed CCPs and its performance evaluation. Section “[Case study](#)” presents a case study, and section “[Conclusion](#)” concludes the paper and provides future directions for research.

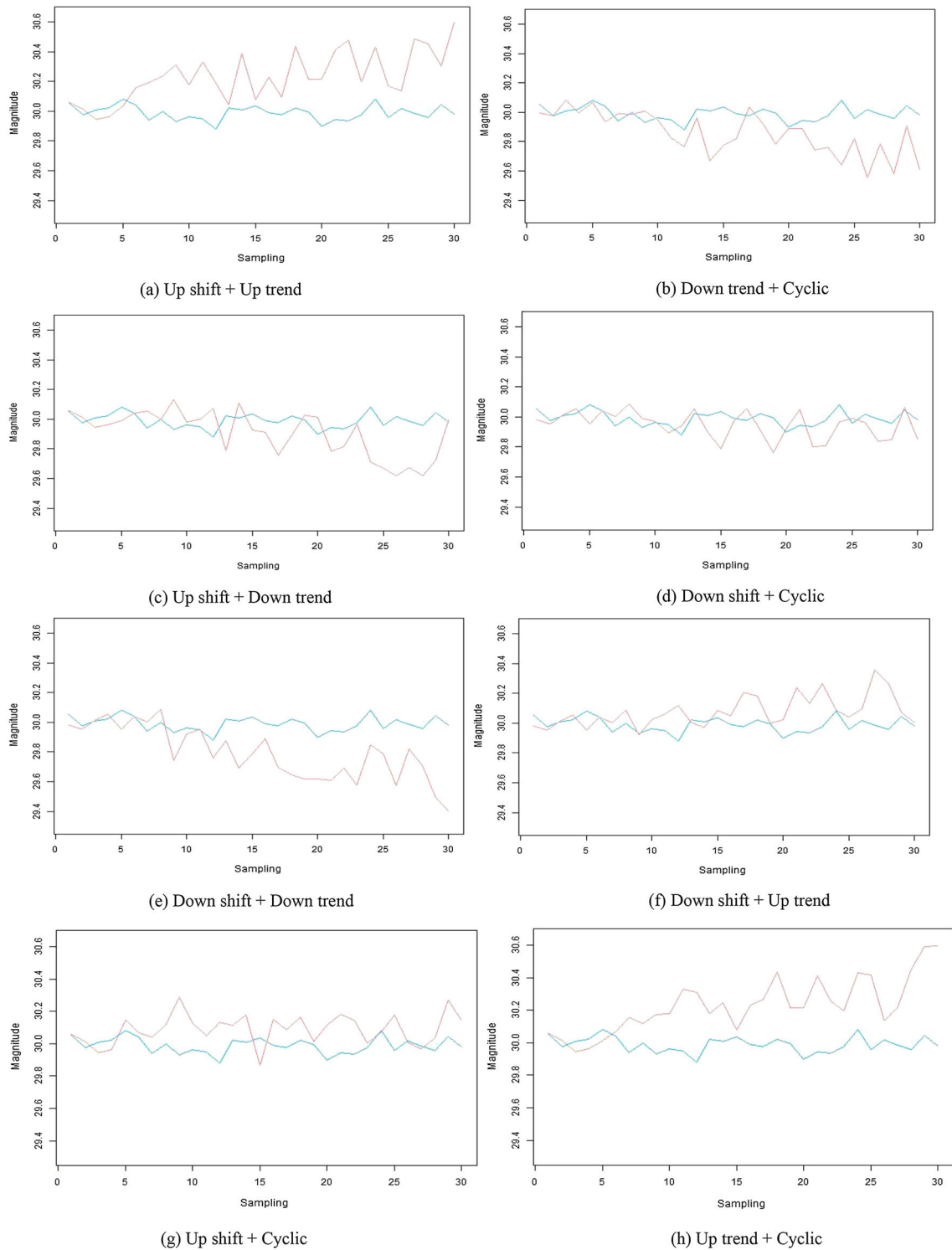


Fig. 2 Illustration of the normal (blue) and mixed abnormal control chart patterns (red) (Color figure online)

Methodology

Data simulation

Essentially, there are six different CCPs in any production process, namely, normal (NOR) pattern, up and down shift

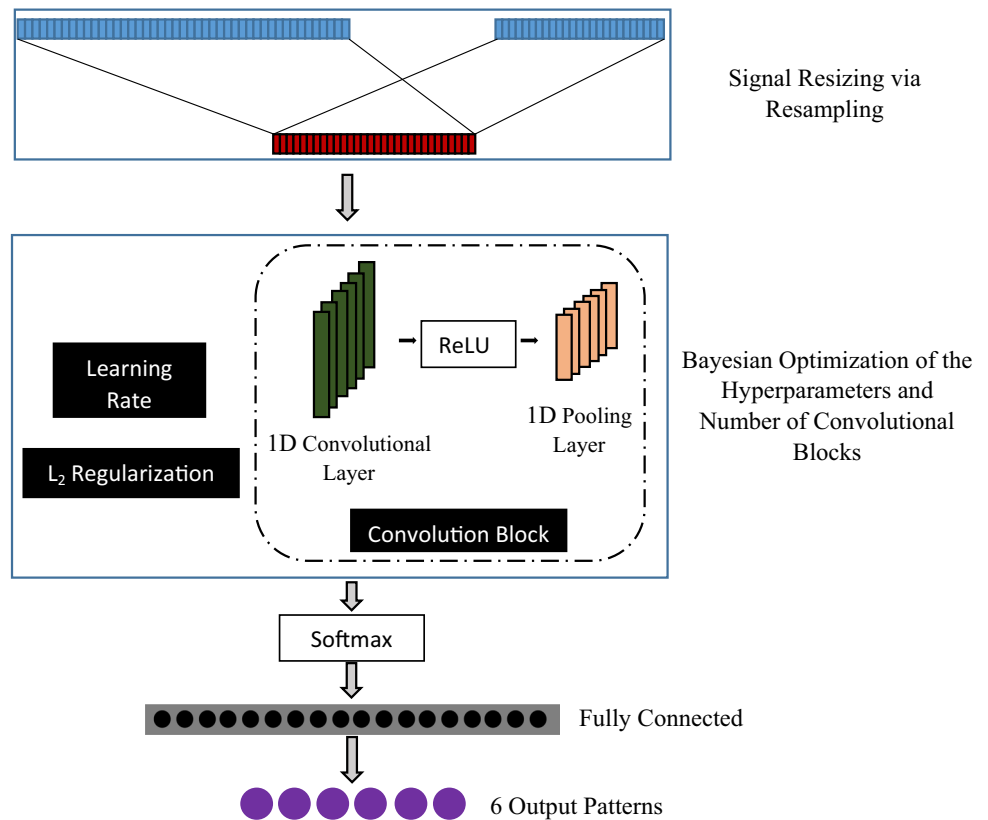
(US and DS) patterns, up and down trend (UT and DT) patterns, and finally, cyclic (CYC) pattern. The six patterns are illustrated in Fig. 1. where the abnormal pattern is shown in red and normal patterns in blue.

We use Monte Carlo simulation to provide a large number of the CCPs for the classification algorithm. The input signal

Table 2 Parameters for generating mixed CCPs

Pattern	Mathematical representation	Parameter value/range
US + UT	$y(t) = \mu + x(t) + v_1 \times s + v_2 \times d \times t$	$\mu = 30, \sigma = 0.05, s \in [1.5\sigma, 3\sigma], d \in [0.1\sigma, 0.3\sigma]$
US + DT	$y(t) = \mu + x(t) + v_1 \times s - v_2 \times d \times t$	$\mu = 30, \sigma = 0.05, s \in [1.5\sigma, 3\sigma], d \in [0.1\sigma, 0.3\sigma]$
DS + UT	$y(t) = \mu + x(t) - v_1 \times s + v_2 \times d \times t$	$\mu = 30, \sigma = 0.05, s \in [1.5\sigma, 3\sigma], d \in [0.1\sigma, 0.3\sigma]$
DS + DT	$y(t) = \mu + x(t) - v_1 \times s - v_2 \times d \times t$	$\mu = 30, \sigma = 0.05, s \in [1.5\sigma, 3\sigma], d \in [0.1\sigma, 0.3\sigma]$
US + CYC	$y(t) = \mu + x(t) + v_1 \times s \times t + v_2 \times \alpha \times \sin(2\pi t/\omega)$	$\alpha \in [1.5\sigma, 4\sigma], \omega \in \{4, 5, 6, 7, 8\}$
DS + CYC	$y(t) = \mu + x(t) - v_1 \times s \times t + v_2 \times \alpha \times \sin(2\pi t/\omega)$	$\mu = 30, \sigma = 0.05, s \in [1.5\sigma, 3\sigma], \alpha \in [1.5\sigma, 4\sigma], \omega \in \{4, 5, 6, 7, 8\}$
UT + CYC	$y(t) = \mu + x(t) + v_1 \times d \times t + v_2 \times \alpha \times \sin(2\pi t/\omega)$	$\mu = 30, \sigma = 0.05, d \in [0.1\sigma, 0.3\sigma], \alpha \in [1.5\sigma, 4\sigma], \omega \in \{4, 5, 6, 7, 8\}$
DT + CYC	$y(t) = \mu + x(t) - v_1 \times d \times t + v_2 \times \alpha \times \sin(2\pi t/\omega)$	$\mu = 30, \sigma = 0.05, d \in [0.1\sigma, 0.3\sigma], \alpha \in [1.5\sigma, 4\sigma], \omega \in \{4, 5, 6, 7, 8\}$

Fig. 3 The structure of the proposed CCPR method



of the CCPR problem has three main components $y(t) = \mu + x(t) + d(t)$. A constant term represents the process mean μ , random term $x(t)$ that follows normal distribution $x(t) \sim N(0, \sigma)$ with $\mu = 0$ and σ standard deviation to represent the natural variability in the process. Finally, a function to model each specific abnormal pattern $d(t)$ where $d(t) = 0$ for the NOR pattern. For the US and DS patterns, $d(t) = \pm v \times s$ 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; positive sign is used for the US pattern, while negative sign is used for the DS pattern. Similarly, in

the case of the UT and DT patterns, $d(t) = \pm v \times d \times t$ 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. Lastly, for the CYC pattern, $d(t) = v \times \alpha \times \sin(2\pi t/\omega)$ where α is the amplitude of a cycle, and ω is the period of a cycle. The formulation and used parameters/ranges of CCPs are summarized in Table 1.

It is worth mentioning that in some literature, the parameter setting lacked randomness (e.g., parameter setting range was too narrow or using fixed values) when generating simu-

Fig. 4 1D signal resampling

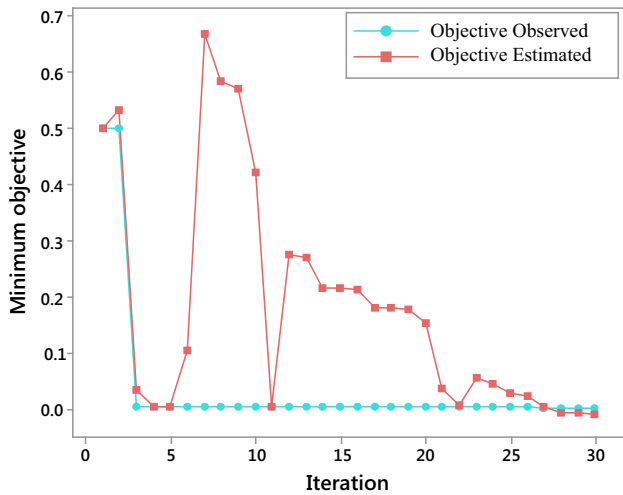
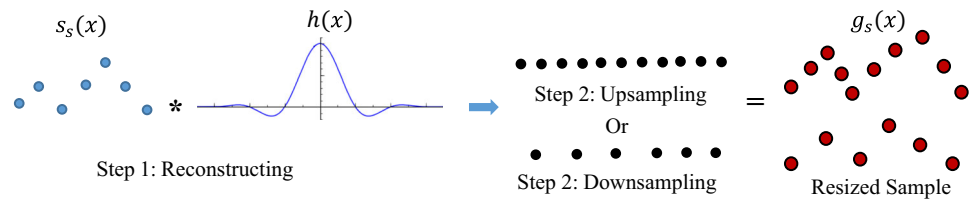


Fig. 5 Optimization process: minimum objective tracing

lation data (Zan et al., 2020b) which contradicts the real-life CCPs. In addition, in the majority of previous literature, about 60 sampling points were used as an input size for whatever model used (Zan et al., 2020b). This could ensure a more considerable difference between different CCP types; however, it lowers the detection ability of the model. To imitate actual manufacturing processes, we tend to make complicated changes to the monitored object. The mean μ is set to 30, and the standard deviation σ is set to 0.05. The rest of the CCP parameters are selected randomly within a specific range. For the slope, $d \sim U(0.1\sigma, 0.3\sigma)$, the magnitude of the shift $s \sim U(1.5\sigma, 3.0\sigma)$, the amplitude of cyclic patterns $\alpha \sim U(0.1\sigma, 0.3\sigma)$, and the period of cycle $\omega \sim U(4, 8)$. Also, to provide a more realistic resemblance to the real-world situations, the starting point of the abnormal pattern $v \sim U(4, 10)$. This means that every window of abnormal

pattern starts with in-control data and eventually includes only out-of-control vectors regardless of the window size. This indeed increased the difficulty of the CCPR because of the large randomness of parameters.

On the other hand, in real-life situations, there is no guarantee that CCP will appear in a sole manner. The traditional CCPR methods in such cases will only detect one pattern at a time, which can be misleading. Hence, we simulate various mixed CCP to be used along with the six traditional CCPs which are shown in Fig. 2. We also show the equations used for the mixed patterns and their parameters in Table 2. Similar to the simulation for the traditional CCPs, the μ is set to 30, and the σ is set to 0.05. Also, $d \sim U(0.1\sigma, 0.3\sigma)$, $s \sim U(1.5\sigma, 3.0\sigma)$, $\alpha \sim U(0.1\sigma, 0.3\sigma)$, and $\omega \sim U(4, 8)$ and independent from each other.

Proposed pattern recognition method for standard CCP

We use CNN as the central core of the proposed algorithm. The advantage of using CNN in the context of the CCPR problem is that it can automatically detect significant features with no need for any domain expertise. CNN consists of six elementary layers, namely, the input layer, the convolutional layer, activation function layer, pool layer, fully-connected layer, and the output layer.

The combination of convolutional and pooling layers performs feature extraction while the classification occurs at the fully connected layer (Xia et al., 2018). The weights and biases of a convolutional layer are arranged as a sequence of convolutional kernels known as filters). The convolution of multiple input feature maps and multiple convolutional kernels yields to an output feature map as $x_j^l = f$

Table 3 Optimized structure and parameters of the VIS-CNN

Layer No	Name	Parameters
1	Image input	$30 \times 1 \times 1$
2	Convolution	$24 \times 1 \times 1$ convolutions with stride [1 1], 16 filters
3	ReLU	–
4	Max pooling	12×1 max pooling with stride [1 1]
5	Fully connected	6 fully connected layer
6	Softmax	–
7	Classification	–

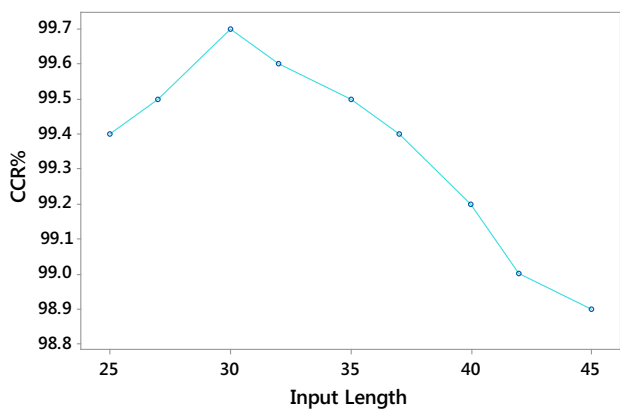


Fig. 6 Comparison of different input sizes

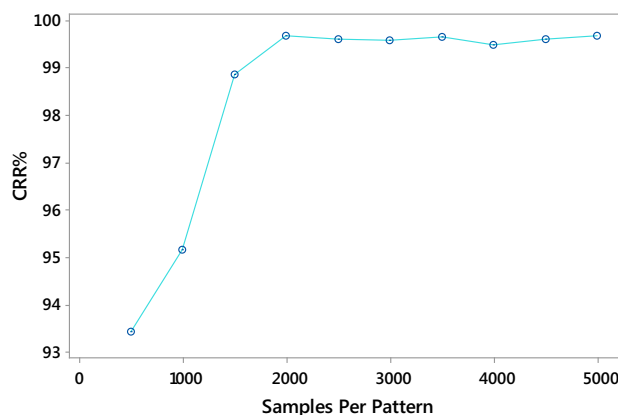


Fig. 7 Effect of different number of training samples on the performance

$(\sum_{i=1}^{D_{l-1}} x_i^{l-1} * \omega_{ij}^l + b_j^l)$, $j = 1, 2, \dots, D_l$ where l represents the layer number, D is the number of feature maps, ω is a convolutional filter between each consecutive layers, x_j^l represents the j th output feature map, b is the bias, and f is the activation function. The most common nonlinearity activation functions are Sigmoid function $f(x) = 1/(1 + e^{-x})$ and Rectified Linear Units (ReLU) function $f(x) = \max(0, x)$. On the other hand, the pooling layer reduces the dimension of the feature maps through downsampling $x_j^l = \text{down}(x_j^{l-1})$, $j = 1, 2, \dots, D_l$ where the x_j^l here is the j th output of the pooling map and down is the pooling function. In this paper we use max pooling where the maximum value of the sub-sampling region is taken as a new feature. The output of the pooling layer is then turned into a single vector, and then weights are applied to predict the correct label in the fully connected layer. The last layer of the CNN is that the output layer represents the number of pattern types to be identified. Usually, the activation function of the output layer is a Softmax function, which is presented as $f(x_i) = e^{x_i} / \sum_{j=1}^N e^{x_j}$. Finally, the goal of the training phase algorithm is to estimate the parameters (ω_{ij}^l and b_j^l) which minimize the cost function. In this paper, the CNN used has inputs and outputs of a width of 1.

The proposed CCPR method is shown in Fig. 3; at first, the input signal is resized to match the input of the CNN using a resampling technique. In real applications, the length of the monitored data might be changed per unit time due to the production needs, especially in sensor-based real-time monitoring. Therefore, this paper proposes a VIS-CNN to tackle the CCPR regardless of the input size. We further continue by determining the best structure of the used CNN by defining the number of convolutional blocks and optimal hyperparameters using BO. The suggested VIS-CNN can take raw data as an input and provide end-to-end recognition.

The raw CCP data, will be divided into training, validation, and test sets, are generated as described in the “Data simula-

tion” section. The weights and biases of a CNN structure are optimized and adjusted by Back Propagation to minimize the objective cost function using the training set. The test set is then used to validate the optimized CNN. The performance measure used is the correct recognition rate (CRR), represented as:

$$CRR = \text{correctly identified patterns} / \text{total number of patterns}$$

At last, the proposed VIS-CNN was compared to other existing methods for the CCPR problem.

Image resizing using resampling

In order to use a variable-sized signal, we will resize the input signal using a resampling method. Unlike other methods, resampling can resize an image without losing information or adding noise. Consider a signal represented as a 1D image in our case, $s(x)$ that is ideally sampled with a summation of delta functions or *comb* where

$$\text{comb}(x) = \sum_{n=-\infty}^{\infty} \delta(x - n) \tag{1}$$

and

$$\delta(x - n) = \begin{cases} +\infty, & x = n \\ 0, & x \neq n \end{cases} \tag{2}$$

Then, the sampled signal could be expressed as

$$S_s(x) = s(x) \cdot \frac{1}{X} \text{comb}\left(\frac{x}{X}\right) = \sum_n s(nX) \delta(x - nX) \tag{3}$$

Resizing an image by resampling (upsampling or downsampling) can be described as a two-step process. First, the continuous signal is reconstructed using some low-pass filter, $h(f)$, via a spatial-domain convolution to pass the baseband

Fig. 8 Confusion matrix for the CCPR using VIS-CNN and raw data

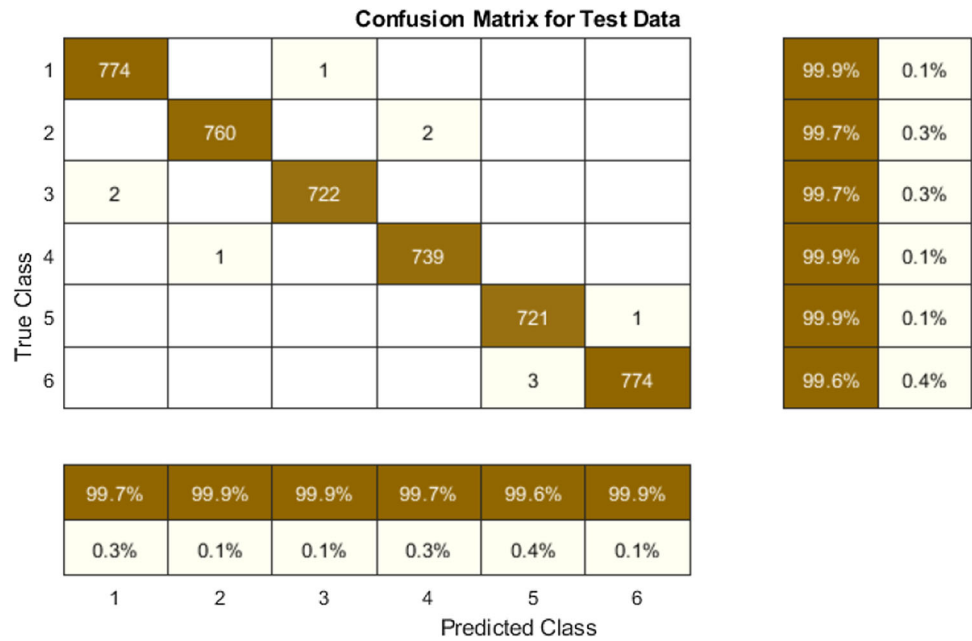


Table 4 Performance of different machine learning methods for CCPR

Reference	Input data type	Classifier	Correct recognition ratio %
Ranaee and Ebrahimzadeh (2013)	Shape and statistical features	MLP	99.15
Kao et al. (2016)	Independent component analysis	SVM	98.94
Zhou et al. (2018)	Shape and statistical features	FSVM	99.28
Addeh et al. (2018)	Shape and statistical features	Bees-RBF	99.63
Zan et al. (2020b)	Raw data	1D-CNN	99.30
Zan et al. (2020a)	Raw data	Bi-LSTM	99.26
Aziz kalteh and Babouei (2020)	Shape and statistical features	CWOA-ANFIS	99.77
Proposed method	Raw data	VIS-CNN	99.78

replicate of the sampled signal spectrum. As illustrated in Fig. 4, the reconstructed continuous signal is then resampled with the new sampling frequency, the reconstructed continuous signal and the resampled signal are expressed, respectively, as

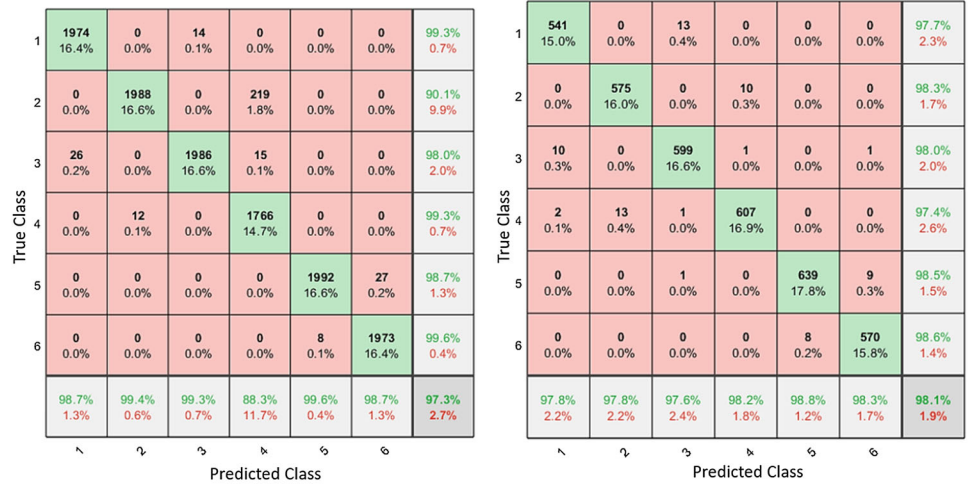
$$g(x) = (s_s(x) * h(x)) = \sum_n s(nX)h(x - nX) \tag{4}$$

$$g_s(x) = g(x) \cdot \frac{1}{X'} \text{comb}\left(\frac{x}{X'}\right) = \sum_n g(nX')\delta(x - nX') \tag{5}$$

In general, resampling can be achieved through various interpolation methods. Image interpolation tries to achieve the best estimation of a pixel’s intensity using neighboring pixel values on proximity basis. Interpolation is above all essen-

tial when resampling the image to meet the specifications or present the final image with no visual loss. The distance between samples of the original signal is divided into several intervals. Then, interpolation is used to estimate the values of the resampled points. Image interpolation can generally be achieved through three methods: nearest neighbor, bilinear interpolation, or bicubic interpolation (Antoniou, 2016). Since each method has its own merits and challenges, more discussion on the choice of appropriate method is provided by Hemanth and Balas (2019). For a 1D signal, linear interpolation is widely adopted for ease of calculation and good efficiency; hence we use it here. Note that the underlying assumption that it is always possible to distinguish between low and high-frequency signals (Lévesque, 2014).

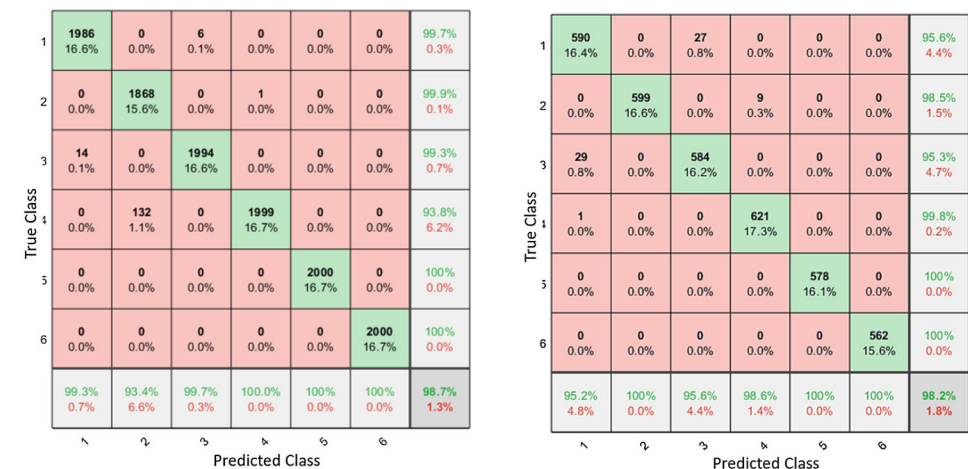
Fig. 9 Correct classification percentage of VIS-CNN (left matrices) vs 1D-CNN (right matrices) for various input sizes



(a) input size 25

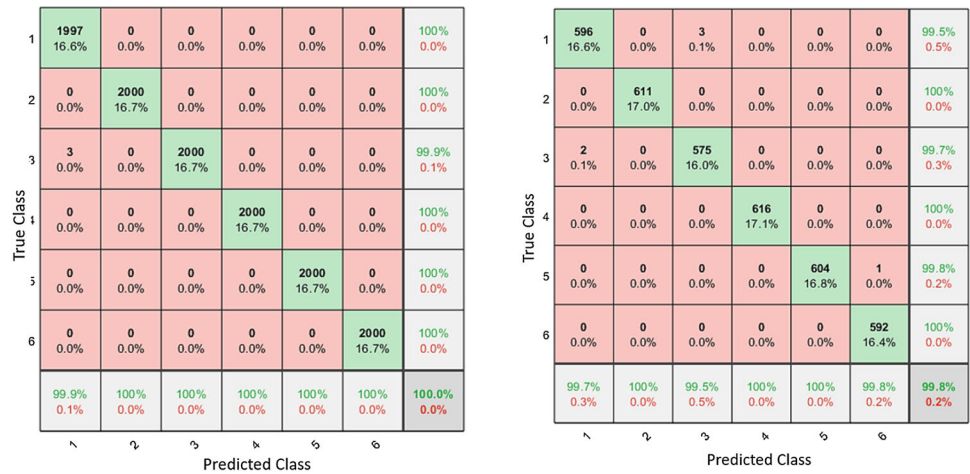


(b) input size 30

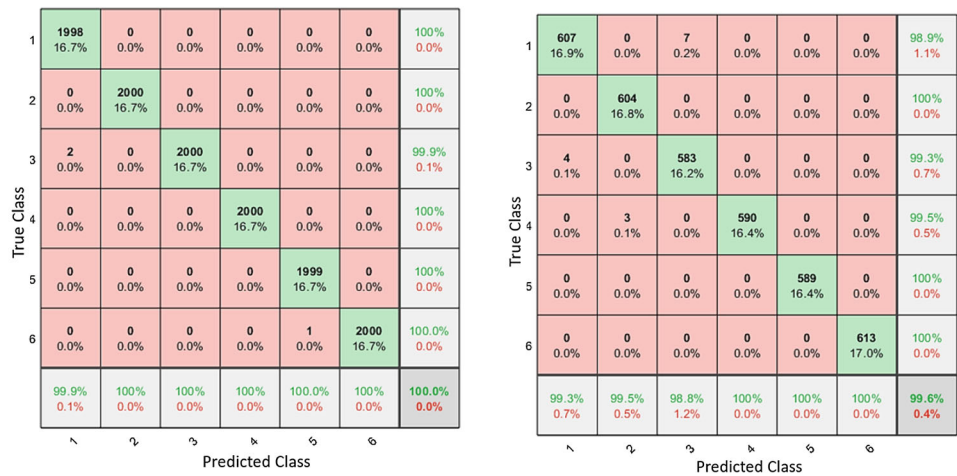


(c) input size 35

Fig. 9 continued



(d) input size 40



(e) input size 45

Tuning of hyperparameters using Bayesian optimization

In this paper, we want to find the best performing hyperparameters of a CNN measured on a validation set. Hyperparameters, in contrast to model parameters, are set by the operating engineer before training by trial and error. Hyperparameters optimization can be represented in equation form as $\theta^* = \arg \min_{x \in X} E_V(\bar{\theta})$ where $E_V(\bar{\theta})$ represents the objective function evaluated on the validation set, θ^* is the set of hyperparameters that yield the lowest value of the score, and where $\bar{\theta}$ is a vector of the considered hyperparameters. Optimizing hyperparameters is usually very expensive due to the need to retrain the model for every set of new hyperparameters and then calculate the validation metric. BO can be advantageous over other methods due to its ability to minimize the number of training cycles (Victoria & Maragatham, 2020). In Bayesian optimization (BO), the objective function is repre-

sented as a probabilistic model. Intuitively, instead of merely using the local gradient and Hessian approximations, we take advantage of all the available information from the previous evaluations. This way, computations are done to determine the next point to evaluate. In general, BO has the ability to find the extrema of complicated nonconvex functions with relatively small number of evaluations (Brochu et al., 2010). Assume that $E_V(\bar{\theta})$ is drawn from a prior that follows Gaussian Process (GP), i.e., $E_V(\bar{\theta}) \sim N(0, K)$, where $E_V(\bar{\theta})$ has some Gaussian noise with zero mean and standard deviation of σ and its kernel is given as

$$K = \begin{bmatrix} k(\bar{\theta}_1, \bar{\theta}_1) & \dots & k(\bar{\theta}_1, \bar{\theta}_t) \\ \vdots & \ddots & \vdots \\ k(\bar{\theta}_t, \bar{\theta}_1) & \dots & k(\bar{\theta}_t, \bar{\theta}_t) \end{bmatrix} + \sigma_{noise}^2 I$$

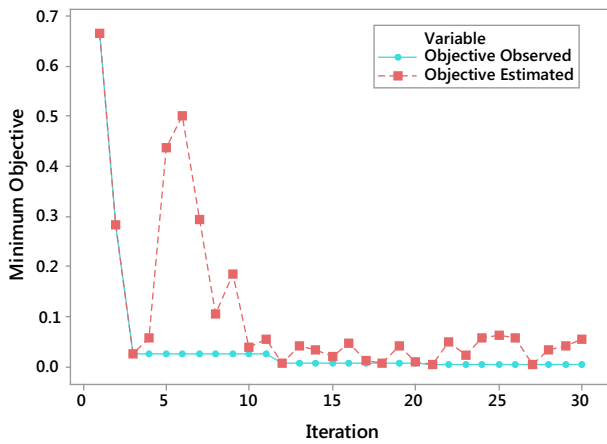


Fig. 10 Optimization process for the Bi-LSTM network: minimum objective tracing

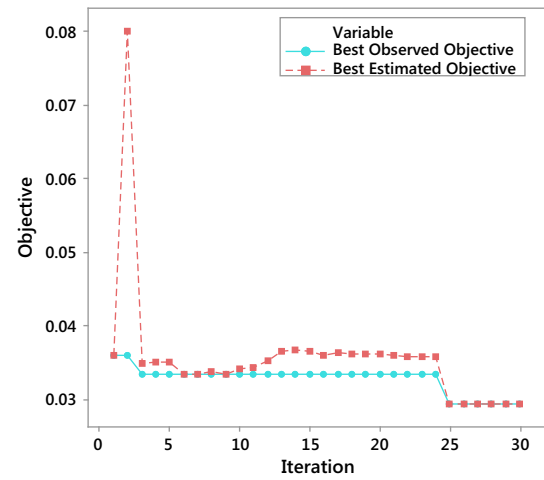


Fig. 12 Optimization process for the modified VIS-CNN: minimum objective tracing

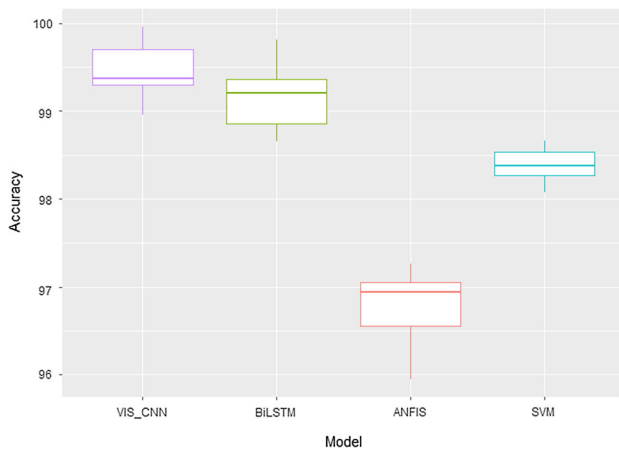


Fig. 11 Classification accuracy for the four models under variable input

where $k(\bar{\theta}_1, \bar{\theta}'_1)$ is the covariance function and σ_{noise}^2 is the variance of the Gaussian noise. Assume the observations from the preceding iterations denoted as $D_{1:t}$ and are in the form as $\{\bar{\theta}_{1:t}, E_{1:t}^V\}$ where $E_{1:t}^V = E_V(\bar{\theta}_{1:t})$.

Denote $\bar{\theta}_{1+t}$ as the next point to evaluate and the value of the function at $\bar{\theta}_{1+t}$ as $E_{1+t}^V = E_V(\bar{\theta}_{1+t})$. Under the GP prior, $E_{1:t}^V$ and E_{1+t}^V are known to be jointly Gaussian, and the predictive distribution can be obtained (Rasmussen & Nickisch, 2010):

$$E_{t+1}^V | D_{1:t} \sim N\left(\mu(\bar{\theta}_{t+1}), \sigma^2(\bar{\theta}_{t+1}) + \sigma_{noise}^2\right) \tag{6}$$

where

$$\mu(\bar{\theta}_{t+1}) = k^T (K + \sigma_{noise}^2 I)^{-1} E_{1:t}^V, \tag{7}$$

$$\sigma^2(\bar{\theta}_{t+1}) = k(\bar{\theta}_{t+1}, \bar{\theta}_{t+1}) - k^T (K + \sigma_{noise}^2 I)^{-1} k, \tag{8}$$

and

$$k = [k(\bar{\theta}_{t+1}, \bar{\theta}_1)k(\bar{\theta}_{t+1}, \bar{\theta}_2) \cdots k(\bar{\theta}_{t+1}, \bar{\theta}_t)]^T \tag{9}$$

Consequently, the predictive posterior distribution $E_{t+1}^V | D_{1:t}$ can be characterized by its predictive mean function $\mu(\bar{\theta}_{t+1})$ and predictive variance function $\sigma^2(\bar{\theta}_{t+1})$, which solely depend on the choice of the covariance function $k(\bar{\theta}, \bar{\theta}')$. In this study, the ARD Matern 5/2 covariance function is used (Snoek et al., 2012).

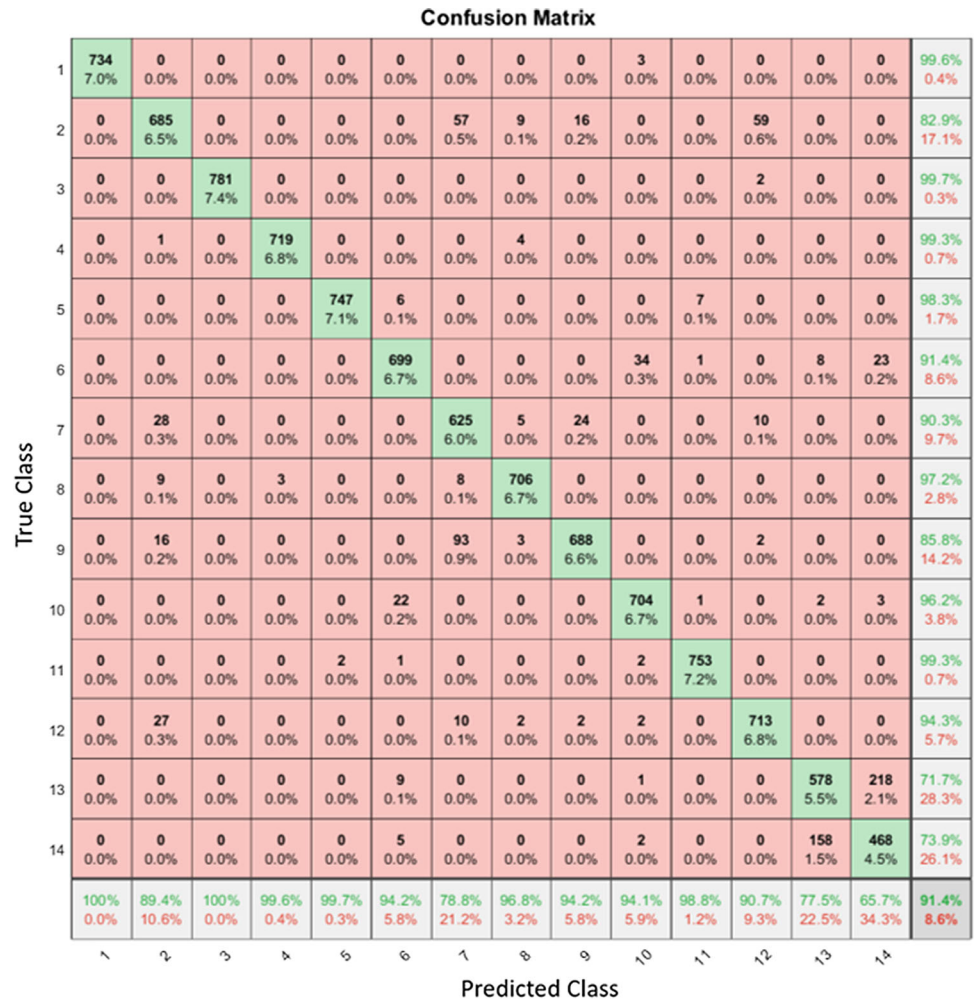
$$k(\bar{\theta}, \bar{\theta}') = \gamma_0 \left(1 + \sqrt{5 \sum_{l=1}^L (\bar{\theta}_l - \bar{\theta}'_l)^2 / \gamma_l^2} + \frac{5}{3} \sum_{l=1}^L (\bar{\theta}_l - \bar{\theta}'_l)^2 / \gamma_l^2 \right) e^{-\sqrt{5 \sum_{l=1}^L (\bar{\theta}_l - \bar{\theta}'_l)^2 / \gamma_l^2}} \tag{10}$$

where γ_0 is covariance amplitude and γ_l is the characteristic length scale, which are the hyperparameters of the GP. Note that these hyperparameters are distinct from those being subjected to the overall Bayesian optimization. The most commonly advocated approach is to use a point estimate of these parameters by optimizing the marginal likelihood under the GP (Snoek et al., 2012). As stated earlier, BO tends to allocate more computations on determining the next point $\bar{\theta}_{t+1}$ to evaluate, and it does so by using an acquisition function created from the above-discussed predictive posterior distribution. Acquisition functions balance between discovering new areas in the objective space and exploiting areas that are already identified to have auspicious values. The acquisition function is based on Expected Improvement (EI) over the best-expected value $\mu_{best} = \arg \min_{\bar{\theta}_j \in \bar{\Theta}_{1:t}} \mu(\bar{\theta}_j)$,

Table 5 Optimized structure and parameters of the modified VIS-CNN

Layer No	Name	Parameters
1	Image input	$30 \times 1 \times 1$
2	Convolution	$35 \times 1 \times 1$ convolutions with stride [1 1], 45 filters
3	ReLU	
2	Convolution	$35 \times 1 \times 1$ convolutions with stride [1 1], 45 filters
3	ReLU	
4	Max pooling	12×1 max pooling with stride [1 1]
5	Fully connected	12 fully connected layer
6	Softmax	
7	Classification	

Fig. 13 Correct classification percentage of VIS-CNN



which has a closed-form solution under the GP (Snoek et al., 2012) assumption as follows:

$$a_{EI}(\bar{\theta}_{t+1}) = \sigma(\bar{\theta}_{t+1})[Z\Phi(Z) + \phi(Z)] \tag{11}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are CDF and PDF of the standard normal, respectively, and $Z = \frac{\mu_{best} - \mu(\bar{\theta}_{t+1})}{\sigma(\bar{\theta}_{t+1})}$. Therefore, the point to be evaluated in the next iteration is determined so that it

maximizes the acquisition function. It is noted that, unlike the original unknown objective function, $a_{EI}(\cdot)$ in Eq. (11) can be cheaply sampled to be maximized. Steps for the BO adopted in this paper is described as follows:

1. Define the $\bar{\theta}$ domain of hyperparameters to search over
2. Calculate the predictive mean and predictive variance functions $\mu(\bar{\theta}_{t+1})$ and $\sigma^2(\bar{\theta}_{t+1})$ using Eqs. (7) and (8)

using the chosen ARD Matérn 5/2 covariance function in Eq. (10)

3. Find $\bar{\theta}_{t+1}$ by optimizing the acquisition function over the $\bar{\theta}_{t+1} = \arg \max_{\bar{\theta}} a_{EI}(\bar{\theta} | D_{1:t})$
4. Evaluate the validation error $E_V(\bar{\theta}_{t+1})$ through the deep learning model with $\bar{\theta}_{t+1}$ determined in step 3
5. A history consisting of (score, hyperparameters) pairs used by the algorithm to update GP

VIS-CNN for standard CCP

We generated 2000 samples of each data length (25, 30, 35, 40, and 45) for the six patterns. The dataset was randomly divided into three parts, of which 70% of samples were used to train VIS-CNN, 15% of samples for validation, and the rest were used for testing. The data is all resampled and resized to have a length of 30. To this point, the structure and hyperparameters of the network have not been determined yet. BO is then used to obtain the optimal hyperparameters.

We choose the variables to be optimized and the corresponding ranges to search using BO. First, we optimize the network section depth. This parameter controls the number of convolutional blocks in the network. We set the searching range between 1 and 4. Then, we search for the best learning rate, note that it may depend on the data as well as the network structure. The searching set is between 0.01 and 1. We use L_2 regularization in order to avoid overfitting, and we search the space of regularization strength to find a good value. Adaptive Moment Estimation (Adam) is used for the training process.

The values of the observed and estimated objective functions of the optimization process are illustrated in Fig. 5. It can be seen that function evaluation ends with 30 iterations because of the model saturation. At the end of the 27th iteration, the minimum observed objective is achieved, which will later be used to construct the best model.

Table 7 gives the values for the hyperparameters (best values are bolded) obtained from the validation set over the 30 iterations. The section depth is found to be 1, which implies that only one convolutional layer is enough for achieving the highest accuracy. The optimal network structure and its parameters are shown in Table 3.

Influence of resampling size and number of training samples on the recognition performance

Although the developed model can deal with variable input sizes, the used CNN can only have a fixed input size. Hence, we need to find the optimal output size of the resampling process, which will be the CNN's input. To select the most convenient input size, we performed a set of simulations whose results are shown in Fig. 6. Using the same training

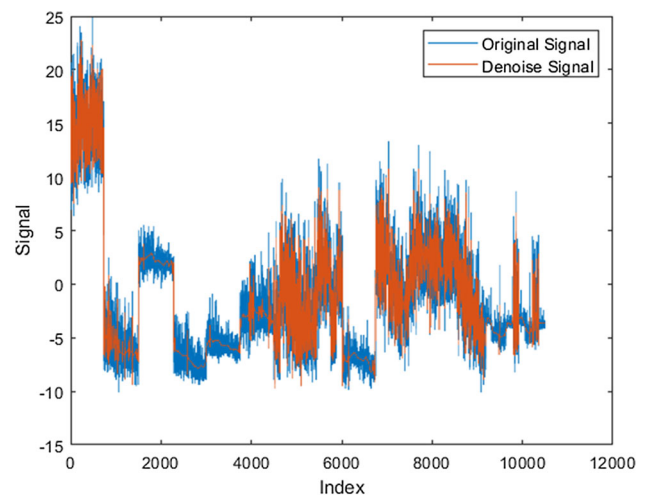


Fig. 14 Example of the noise in the extracted feature for the original and denoised signal

set, the size of the resized signal is tested for various sizes. We use the Bayesian optimized CNN for all of the trials. As shown in Fig. 6, the best-resized signal input of the network is 30, and the corresponding CRR is 99.7%.

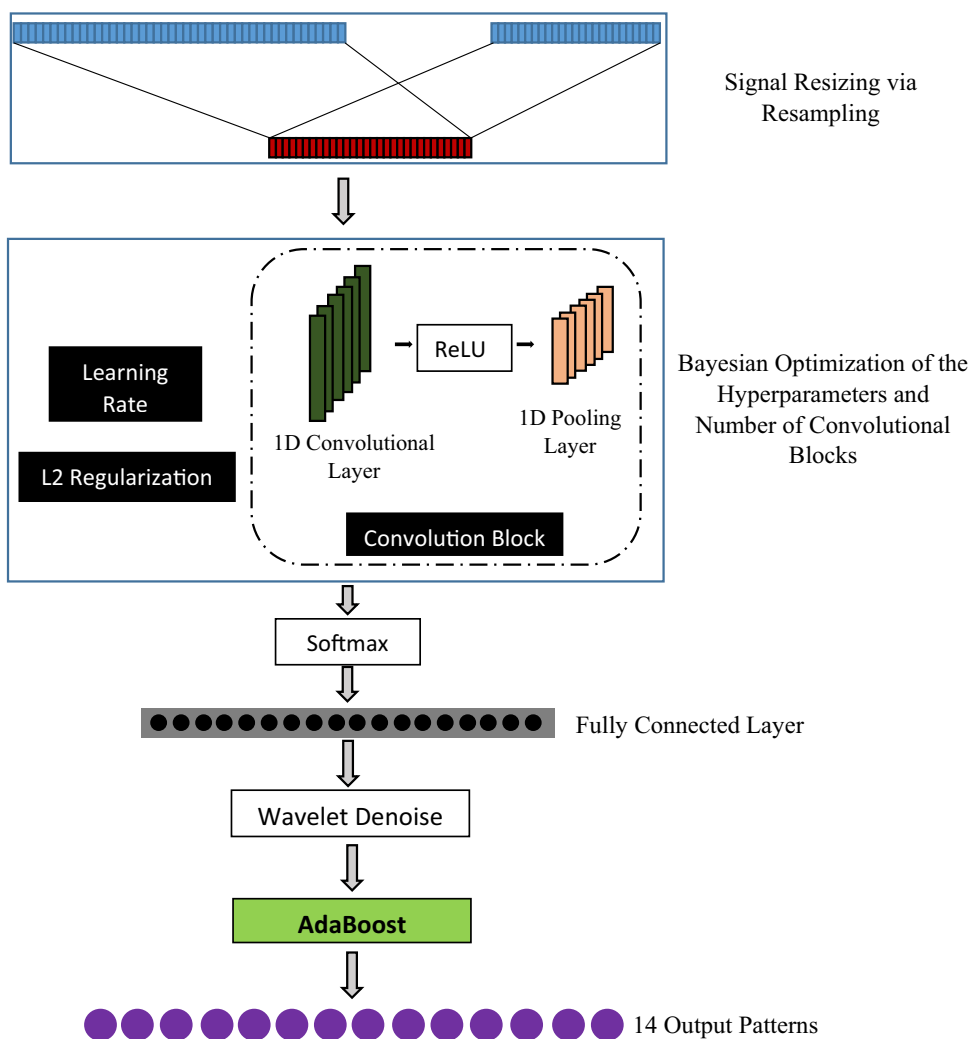
On the other hand, different number of training samples per pattern has been explored in order to define the best number of samples that can achieve the required accuracy. As shown in Fig. 7, the number of samples of each CCP was respectively set to a range between 500 and 5000 with a step of 500. It can be observed that starting from 2000 training samples per pattern, no significant change can be noticed in the CRR.

Performance evaluation of VIS-CNN

In order to evaluate the classifier performance more intuitively, the confusion matrix is used. The diagonal values represent the count of the number of correctly recognized patterns, while the values around the diagonal denote the number of misclassifications. For the VIS-CNN, the confusion matrix is shown in Fig. 8. The CRR of the VIS-CNN compared to other recent CPPR literature is shown in Table 4. The results showed that the VIS-CNN has the highest CRR (99.78%) compared to the other methods. Since we classify six faults, the number from 1 to 6 represents respectively the following patterns CYC, DT, NOR, DS, US, and UT.

To assess the proposed method's performance further, a comparative experiment of VIS-CNN and 1D-CNN proposed by Zan et al. (2020b) in terms of various input sizes is conducted, as shown in Fig. 9. The 1D-CNN used by Zan et al. (2020b) was shown to be superior over multi-layer perceptron (MLP) in cases when the input was either the raw data or the feature set. The feature set included the statistical

Fig. 15 The new structure of the VIS-CNN with AdaBoost



features (mean, standard deviation, skewness, and kurtosis) and shape features (S, NC1, NC2, APLM, APSL). In addition, the 1D-CNN had almost a similar performance as the method proposed by Miao and Yang (2019), where CNN was used with shape and statistical features as inputs. Hence, we choose Zan et al. (2020b) for comparison. It is important to note that the structure of the VIS-CNN is different from the 1D-CNN proposed by Zan et al. (2020b) in two main aspects. First, the VIS-CNN structure can deal with variable input size due to the incorporation of the image resizing process, which is not the case in the 1D-CNN where the network needs to be retrained for each different input size. Secondly, BO showed that the optimal structure for the VIS-CNN has one convolutional layer which is simpler and less computationally expensive than the 1D-CNN which has two convolutional layers.

As depicted in Fig. 9, when the window size is 25, the image has to be enlarged. The VIS-CNN has CRR = 97.3%

less than the 1D-CNN that has CRR = 98.1%. When the window size is increased to be 30, the VIS-CNN has a higher CRR = 99.6% than the 1D-CNN with CRR = 99.3%. For 35 window size, still, the VIS-CNN outperformed the 1D-CNN with CRR = 99.7% for the former and 98.2% for the latter. For 40 and 45 window sizes, the VIS-CNN has the highest CRR of 100% and 99.8 and 99.8 for the 1D-CNN, respectively. It worthy to mention that the training time of the VIS-CNN is between 7 to 13 min on a personal computer with an i7-8700 CPU@3.20-GHz CPU, 16 GB RAM and 64-bit operating system. The VIS-CNN is advantageous over the 1D-CNN since only one training process is needed for any input size.

In order to show the performance of the VIS-CNN against other methods in terms of its capability to deal with the variable window size we consider substituting the 1D-CNN with SVM and ANFIS and Bi-LSTM. For a fair comparison, we are dealing with raw input data. In addition, the same method

Fig. 16 Correct classification percentage of VIS-CNN with AdaBoost

Confusion Matrix

True Class	1	733 7.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	2	0 0.0%	755 7.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	99.6% 0.4%
	3	0 0.0%	0 0.0%	780 7.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	722 6.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	748 7.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.9% 0.1%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	742 7.1%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.7% 0.3%
	7	0 0.0%	8 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	787 7.5%	7 0.1%	39 0.4%	0 0.0%	0 0.0%	4 0.0%	0 0.0%	0 0.0%	0 0.0%	93.1% 6.9%
	8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.0%	721 6.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.6% 0.4%
	9	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	691 6.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.6% 0.4%
	10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	744 7.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	761 7.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	12	1 0.0%	1 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	781 7.4%	0 0.0%	0 0.0%	0 0.0%	99.5% 0.5%
	13	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	633 6.0%	32 0.3%	0 0.0%	95.0% 5.0%
	14	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	113 1.1%	680 6.5%	0 0.0%	85.5% 14.5%
			99.9% 0.1%	98.6% 1.4%	99.9% 0.1%	100% 0.0%	99.9% 0.1%	100% 0.0%	99.2% 0.8%	98.9% 1.1%	94.7% 5.3%	99.5% 0.5%	99.9% 0.1%	99.4% 0.6%	84.9% 15.1%	95.5% 4.5%	97.9% 2.1%
		1	2	3	4	5	6	7	8	9	10	11	12	13	14		
		Predicted Class															

used for resampling is used when training and testing the SVM and ANFIS based models. In contrast, this is not the case for the Bi-LSTM since it already has the ability to perform inference on variable lengths. In addition, a dataset with 2000 samples of each pattern is simulated where 70% percent was used for the training, and 30% was used for testing.

For the Bi-LSTM we choose the variables to be optimized and the corresponding ranges to search using BO. First, we optimize the number of Bi-LSTM layers where the searching range is between 1 and 4. Then, we search for the best learning rate and L2 regularization in the interval 0.01–1 and 0–0.1, respectively. Adam is used for the training process. The values of the observed and estimated objective functions of the optimization process are illustrated in Fig. 10, where the optimal parameters are achieved at the end of the 21st iteration, the minimum observed objective is achieved. The best model is achieved with 2 layers with an initial learning rate of 0.021 and L2 regularization of 0.0036. The Bi-LSTM network achieved a high accuracy of 99.14%, which is very competitive to VIS-CNN; however it took more time than the 1D-CNN due to its sequential nature.

On the other hand, for the SVM, three types of kernels were considered here; namely Linear, Quadratic and Gaussian kernels. For each kernel, box constraint and kernel scale are optimized. The optimal box constraint and kernel scale are found to be 3 and 4.31 for the SVM with Linear kernel with corresponding accuracy of 95.04%. While, the SVM with Gaussian kernel showed 97.9% accuracy through 1.17 and 1.4 optimal box constraint and kernel scale, respectively. Finally, the highest accuracy was obtained by the Quadratic kernel-based SVM with an accuracy of 98.34%. The optimal box constraint and kernel scale are found to be 0.466 and 5.798, respectively.

For the ANFIS, several membership functions are available however, the initial trials showed that it does not play a vital role. Hence we consider the Gaussian membership function since only two parameters represent it. Jang (1993) indicated that the ANFIS algorithm could perform classification between two labels; hence, six ANFIS models were trained and tested for each pattern using the IF–THEN rules. In other words, each ANFIS model gives a result if the specified pattern is observed or not. For the training and testing, a

Table 6 Performance of different machine learning methods for the CCPR for mixed patterns

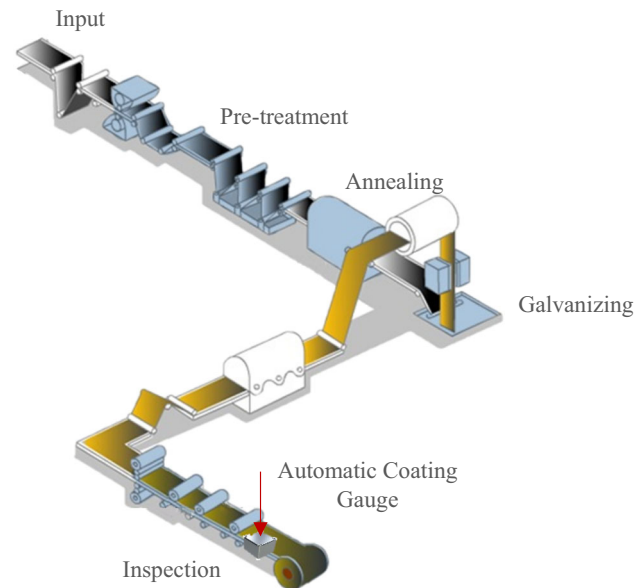
References	Input data type	CRR for 4 mixed patterns %	CRR for 8 mixed patterns %
Zhang and Cheng (2015)	Shape and statistical features	97.6	–
Kadakadiyavar et al. (2020)	Raw data	97.22	–
Proposed method	Raw data	99.5	97.9

hybrid method where backpropagation is used for the parameters associated with the input membership functions, while least squares estimation is used for the parameters associated with the output membership functions. Then the fuzzy inference system is generated using subtractive clustering (Yager & Filev, 1994). The optimal initial step size is found to be 0.018. The final achieved accuracy is 96.7%. Finally, we report the results from 10 replicates of the inference for the VIS-CNN, Bi-LSTM, ANFIS and Quadratic based SVM in Fig. 11. As it can be seen, The VIS-CNN showed the highest performance, followed by the Bi-LSTM. One important conclusion is that relying only on raw data for ANFIS and SVM in the CCPR problem does not achieve good performance; instead, statistical and shape features should be used.

In standard SPC charts, analysis could be done to measure the two types of error committed by the chart (Type I error and Type II error). Type I error could be explained as when it is inferred that a process is out of control when it is actually in control whilst Type II error is exhibited when the process is seen to be in control when it is indeed out-of-control. In our study, we performed 100 simulations, and we found out that the normal pattern was detected to be abnormal with an average of 2.11, where the normal pattern appeared with an average of 724.52 in the testing set, hence Type I error 0.0029. Similarly, we can estimate Type II error to be 0.0021 since the model only. On the other hand, the traditional control chart usually is designed to have a 0.0027 Type I error (370.37 in-control average run length (Maged et al., 2021)).

VIS-CNN for mixed CCP

We generated 28,000 samples of each data length (25, 30, 35, 40, and 45) for the 14 patterns. Similar to what we did for the case of standard CCP, the dataset was random divided into 70% for the training set, 15% samples for validation, and 15% for the testing set. BO is also used to obtain the optimal hyperparameters of the network. We choose the same variables and ranges to be optimized using BO. Figure 12 shows the best observed and estimated values of the objective function and Table 8 shows the hyperparameters for each iteration (best values are bolded). The best parameters for the network are shown in Table 5.

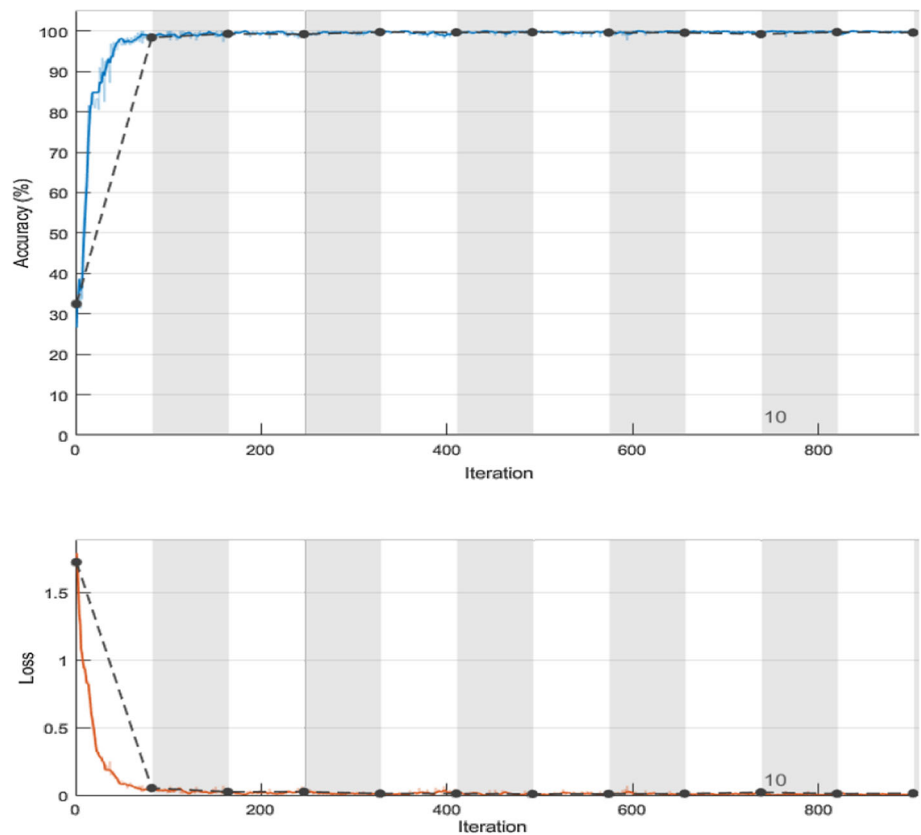
**Fig. 17** Sequence of processes used to produce galvanized cold rolled coils

After the training process, we test the model, and the resultant CRR is 91.4%. The confusion matrix is shown in Fig. 13. For the CCPR problem, this accuracy is quite low and needs to be enhanced.

To further provide more precise classification, we utilize CNN as a feature extractor to obtain features from the input data then employ an ensemble method named Adaptive Boosting (AdaBoost) as a classifier in the higher level of the network. AdaBoost can combine multiple weak classifiers, which in this case are binary decision trees, into a strong classifier. A tree boosting model output \hat{y}_i with K trees is defined as $\hat{y}_i = \sum_{k=1}^K f_k(x_i)$, $f_k \in F$ where $F = \{f(x) = \omega_q(x)\}$ ($q: \mathbb{R}^m \rightarrow T$, $\omega \in \mathbb{R}^T$) is the space of classification trees. f_k divides the trees into structure part and leaf weights part represented respectively as q and ω . T symbolizes the number of leaves in the tree. The objective function of the model can be represented as $O = \sum_i c(\hat{y}_i, y_i) + \sum_k l(f_k)$ where $c(\cdot)$ is the cost function representing the distance between the prediction \hat{y}_i and the object y_i and $l(\cdot)$ is the regularization term.

By examining the extracted features from CNN, which are the input to the AdaBoost, too much noise is observed. For example, we show the noise exhibited in one sample

Fig. 18 Accuracy and validation error for the dataset



of the training set of the boosting algorithm in Fig. 14. Hence, we add a new stage for denoising using wavelet denoiser. The basic idea behind wavelet denoising is that the wavelet transforms signal and image features in a few large-magnitude wavelet coefficients. Wavelet coefficients that are small in value are typically noise and shall be removed without affecting the signal or image quality. After we threshold the coefficients, we reconstruct the data using the inverse wavelet transform. We use wavelet denoise method called empirical Bayesian method with a Cauchy prior as described by Johnstone and Silverman (2004).

The new structure for the CCPR for mixed patterns is illustrated in Fig. 15 where the two new stages are added. Evidently, the new structure could successfully increase the classification accuracy rates from 91.4 to 97.9% as shown in Fig. 16. These observations notably demonstrate the efficacy and realism of the proposed CCPR method.

The CRR of the proposed VIS-CNN with AdaBoost for mixed CPPs is compared to other recent literature is shown in Table 6. Previous literature has only considered 4 mixed patterns, hence we compare the proposed algorithm for the same 4 mixed patterns where it is found that the proposed algorithm has the highest CRR with 99.5%, note that both the methods provided by Zhang and Cheng (2015) and Kadakadiyavar et al. (2020) cannot deal with variable input size and the used fixed input size is 30.

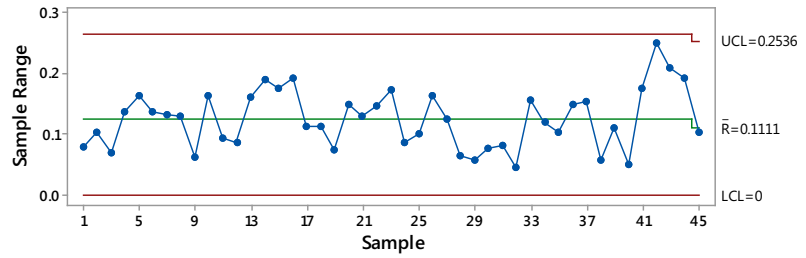
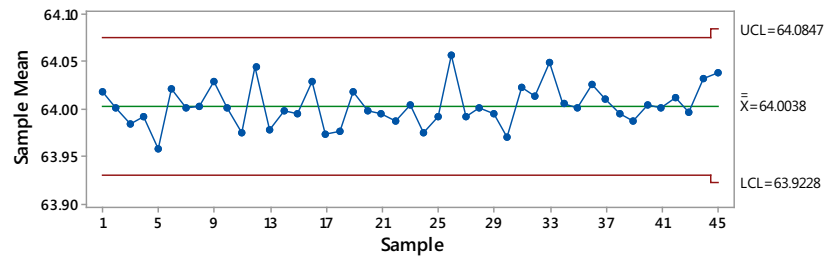
Case study

This section shows the usage of the proposed method for monitoring the thickness of galvanized metal sheets. The coating thickness is an essential requirement as it directly relates to the effectiveness of corrosion protection. As illustrated in Fig. 17, the sheets pass through a pretreatment process at first to remove all the rolling oil and contaminations from the surface. The material properties of the sheets are then enhanced through recrystallization during the annealing process. The resulting sheets are galvanized until the desired coating thickness and weight is achieved.

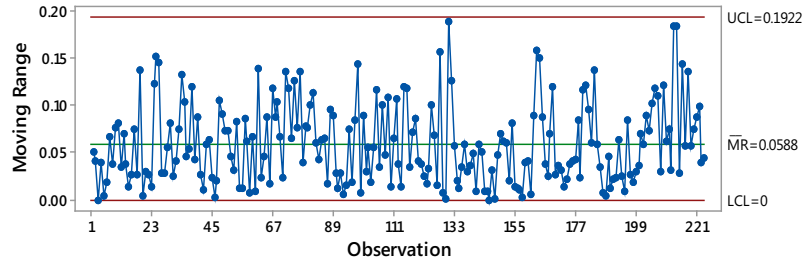
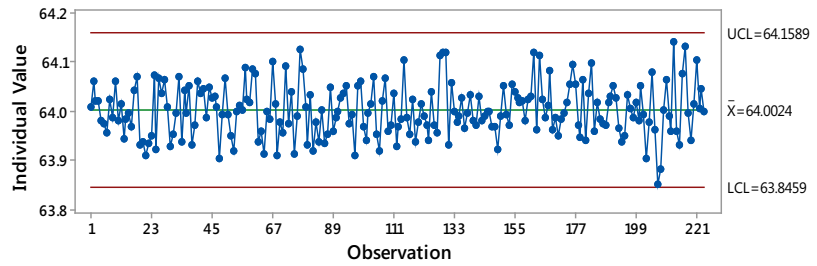
Continuous measuring of coating thickness requires a special measurement system. The installed measuring system uses the magnetic induction principle. An excitation current in a probe generates a low-frequency magnetic field that spreads throughout the magnetizable base material. The non-magnetic coating deteriorates the magnetic field. The reduction in its strength is used to determine the coating's thickness. As metal sheets move down the conveyor, an automatic coating thickness gauge is installed at the line's exit.

The production line rolls sheets at speeds between 30–100 cm/s depending on the required thickness. This means that the sizes of the output samples can change multiple times over the shift, making the traditional CCPR methods inadequate since they may need to be retrained for

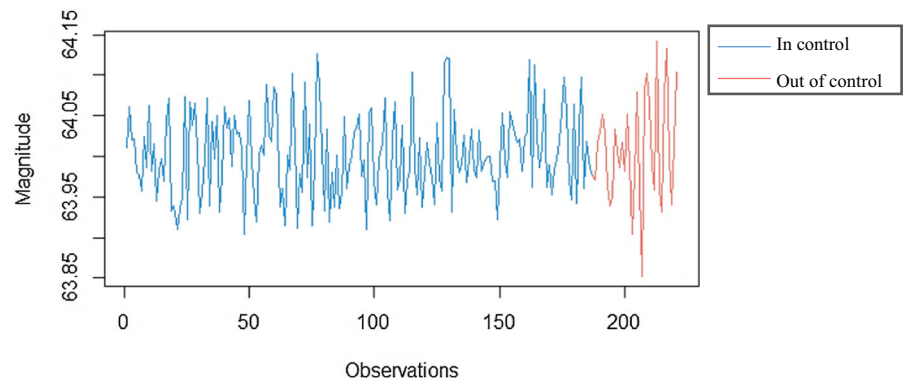
Fig. 19 Process monitoring using control chart and CCPR



(a) *X* and *R* control chart (subgroup size 5)

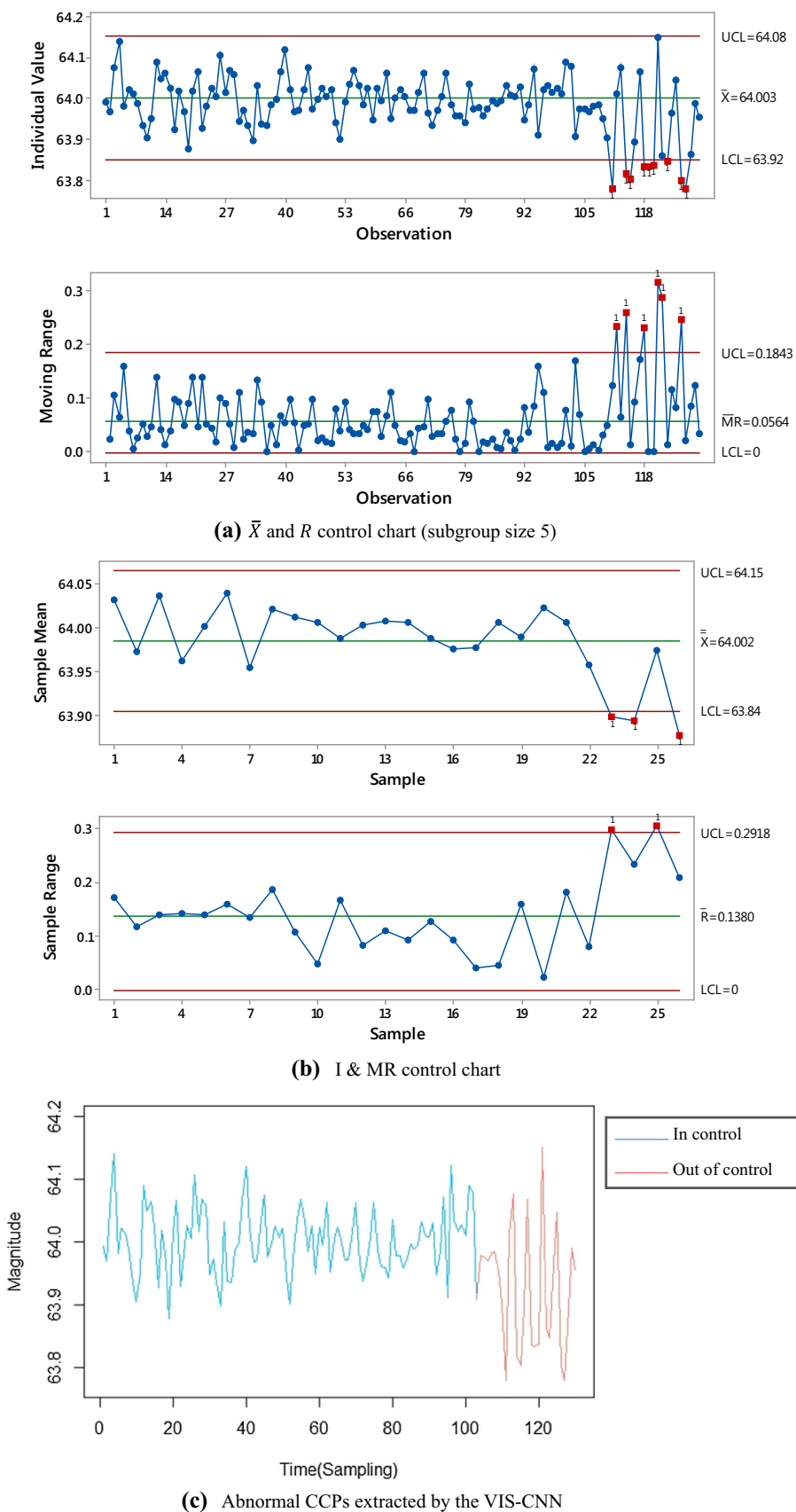


(b) I & MR control chart



(c) Abnormal CCPs extracted by the VIS-CNN

Fig. 20 Process monitoring using control chart and CCPR for mixed CCP



each rolling speed, while the proposed method can resize the sample size for each speed. The raw collected data from the process had to be preprocessed before it was feed to the system due to the outliers and missing values. Instead of removing the outliers entirely we used winsorization by limiting extreme values in the data to reduce the effect of possibly spurious outliers. At the same time, missing values are replaced by the mean value due to its relatively low computational effort. Similar to the number of subgroups in simulation experiments, the reference window size of CCPR is also set as 30 where the in-control mean of the process is 64, with a standard deviation of 0.06. The accuracy and loss error of the training on both the training and validation sets is shown in Fig. 18. It is observed that no over-fitting has been encountered on the validation set in the training procedure. Note that we use the modified VIS-CNN for monitoring all 14 patterns for monitoring all the samples. Also, it is essential to mention that the training time of the modified VIS-CNN is similar to the one in the simulation experiments, which is less than 8 min. As we mentioned before, only one training procedure is required for the network to monitor any input size, note that the prediction can be performed very fast *in-situ*.

Traditional monitoring techniques such as Shewhart charts (including Nelson rules) did not detect any pattern, and the process was declared to be in control, as shown in Fig. 19a, representing \bar{X} and R charts and 02(b), representing the individual chart and MR charts. The proposed CCPR system presented in Fig. 19c detects an uptrend pattern in point 187. This implies that the thickness of the coating layer has been subtly and steadily increasing. A change in the thickness of the coating layer could lead to an unwanted increase or decrease in the weight and change in the sheet properties. The increase in thickness of the coating layer implies that the dipping speed has been decreasing. Analyzing the process, it was found that some internal parts of the mechanism controlling the dipping speed needed to be changed. This illustrates that the CCPR method based on VIS-CNN can correctly identify the traditional CCP appears in a production process. Another pattern was detected which was a combination of cyclic and downtrend patterns. In this process, such a combination can appear if some vibration is encountered, causing the coating layer not to stick properly on the metal sheet. From Fig. 20, we can see that the pattern has been detected by the SPC control charts later than the CCPR system when it was detected at sample 103. The CCPR system can also be advantageous over SPC since it can identify the detected pattern rather than just alarming out-of-control points.

Conclusion

This paper presents a VIS-CNN model to study the well-known CCPR problem in industrial environment. We have mainly tackled the literature gap of developing CCPR classification for variable input size. In addition, in order to increase the efficiency of the network, we optimized the hyperparameters and structure of the VIS-CNN using BO. Furthermore, we include the mixed CCP and provide a modified scheme to achieve a high recognition ratio of 8 mixed patterns on top of the original 6 patterns. The modified scheme includes wavelet noise reduction and Adaptive Boosting. The modified scheme increased significantly enhanced the performance of recognizing the 14 patterns. To demonstrate our algorithm's advantages, an extensive experimental study has been conducted using both simulated and real-world datasets. The results of the proposed VIS-CNN have been compared with the performance of the existing CNN algorithms in terms of the CRR as well as other algorithms from the literature.

This research lays down the foundation for several future lines of research. For example, we have shown the efficacy of the VIS-CNN and modified VIS-CNN on standard and mixed abnormal patterns; however, a sequel research work may consider investigating newer patterns and developing deeper VIS-CNN models. In addition, the developed method should be updated in order to be cost-sensitive where one abnormal pattern may be costlier than other. Finally, a new method that can deal with autocorrelated data should be developed.

Acknowledgements The authors would like to thank the reviewers and the editor for their constructive comments on an earlier version of the paper. This work is supported by National Natural Science Foundation of China (71971181 and 72032005) and by Research Grant Council of Hong Kong (11203519, 11200621). It is also funded by Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA) and Hong Kong Institute of Data Science (Project 9360163).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

See Tables 7 and 8.

Table 7 Observed and estimated objective functions and corresponding hyperparameters

Iter	Objective	Best observed objective	Best estimated objective	Section depth	Initial learning rate	L ₂ Regularization	Evaluation result
1	0.50022	0.50022	0.50022	1	0.33817	1.15E−09	Best
2	0.83111	0.50022	0.53321	2	0.38789	3.24E−10	Accept
3	0.004	0.004	0.032283	3	0.014392	8.39E−09	Best
4	0.83356	0.004	0.0040512	3	0.93109	0.003202	Accept
5	0.83111	0.004	0.0040767	2	0.071964	3.44E−09	Accept
6	0.83156	0.004	0.10271	1	0.9657	1.23E−08	Accept
7	0.834	0.004	0.66651	3	0.017367	1.86E−09	Accept
8	0.005778	0.004	0.58392	1	0.01778	0.008442	Accept
9	0.45089	0.004	0.56914	3	0.014384	0.001203	Accept
10	0.17111	0.004	0.42048	3	0.025446	1.01E−10	Accept
11	0.002667	0.002667	0.0026732	3	0.014147	7.22E−10	Best
12	0.004667	0.002667	0.27543	2	0.01566	1.59E−10	Accept
13	0.834	0.002667	0.26794	4	0.084313	4.20E−09	Accept
14	0.002667	0.002667	0.21541	4	0.011041	1.36E−05	Accept
15	0.66867	0.002667	0.2148	1	0.17453	7.50E−09	Accept
16	0.002667	0.002667	0.21211	1	0.034091	1.06E−09	Accept
17	0.005556	0.002667	0.18008	3	0.01466	1.84E−10	Accept
18	0.50022	0.002667	0.18076	1	0.24567	5.90E−08	Accept
19	0.83422	0.002667	0.17698	3	0.83685	2.48E−07	Accept
20	0.002444	0.0024444	0.15239	1	0.01155	1.00E−09	Best
21	0.003111	0.0024444	0.035833	1	0.010047	8.22E−10	Accept
22	0.002667	0.0024444	0.0059573	1	0.011919	8.08E−10	Accept
23	0.834	0.0024444	0.055756	2	0.036713	1.32E−09	Accept
24	0.006889	0.0024444	0.04365	2	0.029879	8.91E−10	Accept
25	0.004	0.0024444	0.027967	2	0.012849	5.65E−10	Accept
26	0.83533	0.0024444	0.023248	2	0.12412	1.17E−09	Accept
27	0.001777	0.0017778	0.0023458	1	0.013502	1.023e−08	Best
28	0.003333	0.0017778	− 0.0073915	3	0.010032	5.24E−10	Accept
29	0.834	0.0017778	− 0.0058137	4	0.33275	3.35E−10	Accept
30	0.003778	0.0017778	− 0.009824	2	0.01004	1.04E−10	Accept

Table 8 Observed and estimated objective functions and corresponding hyperparameters for modified VIS-CNN

Iter	Objective	Best observed objective	Best estimated objective	Section depth	Initial learning rate	L ₂ Regularization	Evaluation result
1	0.036111	0.036111	0.036111	1	0.016482	2.83E−10	Best
2	0.91544	0.036111	0.080143	2	0.91209	1.36E−10	Accept
3	0.033556	0.033556	0.034863	1	0.01145	0.00021663	Best
4	0.035444	0.033556	0.035052	1	0.02024	7.22E−10	Accept
5	0.92033	0.033556	0.035064	4	0.033447	1.25E−07	Accept
6	0.91533	0.033556	0.033485	1	0.99966	2.16E−10	Accept
7	0.91811	0.033556	0.033553	1	0.98536	0.00713	Accept
8	0.92033	0.033556	0.03377	2	0.3502	0.0051795	Accept
9	0.91544	0.033556	0.033421	1	0.99759	0.00083884	Accept
10	0.035111	0.033556	0.034199	1	0.014318	7.54E−08	Accept
11	0.035889	0.033556	0.034385	1	0.010001	2.03E−07	Accept
12	0.037778	0.033556	0.035308	1	0.014119	1.48E−07	Accept
13	0.040556	0.033556	0.036653	1	0.014164	1.66E−07	Accept
14	0.039	0.033556	0.036848	1	0.034095	1.07E−08	Accept
15	0.036	0.033556	0.036522	1	0.013069	2.16E−07	Accept
16	0.036222	0.033556	0.036068	1	0.010032	5.26E−08	Accept
17	0.038556	0.033556	0.036435	1	0.010001	2.70E−08	Accept
18	0.038444	0.033556	0.036316	1	0.025081	2.89E−09	Accept
19	0.036444	0.033556	0.036321	1	0.025703	4.14E−09	Accept
20	0.038111	0.033556	0.036282	1	0.024989	5.33E−09	Accept
21	0.91811	0.033556	0.036054	3	0.38261	0.0037216	Accept
22	0.043333	0.033556	0.035872	1	0.022979	2.79E−09	Accept
23	0.036111	0.033556	0.035826	1	0.010107	6.54E−07	Accept
24	0.057222	0.033556	0.035847	1	0.13232	1.03E−10	Accept
25	0.029444	0.029444	0.02951	2	0.01139	0.00039916	Best
26	0.036778	0.029444	0.029428	3	0.010036	1.26E−10	Accept
27	0.913	0.029444	0.02943	4	0.97711	1.25E−10	Accept
28	0.047778	0.029444	0.029442	1	0.074786	1.27E−09	Accept
29	0.073111	0.029444	0.029375	2	0.05709	1.02E−10	Accept
30	0.040111	0.029444	0.029375	1	0.049308	5.63E−10	Accept

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*, 79, 202–216. <https://doi.org/10.1016/j.isatra.2018.04.020>
- Antoniou, A. (2016). *Digital signal processing*. McGraw-Hill.
- Azizkalteh, A., & Babouei, S. (2020). Control chart patterns recognition using ANFIS with new training algorithm and intelligent utilization of shape and statistical features. *ISA Transactions*, 102, 12–22. <https://doi.org/10.1016/j.isatra.2019.12.001>
- Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint [arXiv:1012.2599](https://arxiv.org/abs/1012.2599).
- Costantino, F., Di Gravio, G., Shaban, A., & Tronci, M. (2015). SPC forecasting system to mitigate the bullwhip effect and inventory variance in supply chains. *Expert Systems with Applications*, 42(3), 1773–1787. <https://doi.org/10.1016/j.eswa.2014.09.039>
- Fuqua, D., & Razzaghi, T. (2020). A cost-sensitive convolution neural network learning for control chart pattern recognition. *Expert Systems with Applications*, 150, 113275. <https://doi.org/10.1016/j.eswa.2020.113275>
- Hachicha, W., & Ghorbel, A. (2012). A survey of control-chart pattern-recognition literature (1991–2010) based on a new conceptual classification scheme. *Computers & Industrial Engineering*, 63(1), 204–222. <https://doi.org/10.1016/j.cie.2012.03.002>
- Hemanth, D. J., & Balas, V. E. (2019). *Telemedicine technologies: Big data*. Academic Press.
- Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665–685.
- Johnstone, I. M., & Silverman, B. W. (2004). Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences. *The Annals of Statistics*, 32(4), 1594–1649.
- Kadakadiyavar, S., Ramrao, N., & Singh, M. K. (2020). Efficient mixture control chart pattern recognition using adaptive RBF neural network. *International Journal of Information Technology*, 12(4), 1271–1280. <https://doi.org/10.1007/s41870-019-00381-z>

- 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.
- Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., & Togneri, R. (2017). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 3573–3587.
- Khormali, A., & Addeh, J. (2016). A novel approach for recognition of control chart patterns: Type-2 fuzzy clustering optimized support vector machine. *ISA Transactions*, 63, 256–264. <https://doi.org/10.1016/j.isatra.2016.03.004>
- Lévesque, L. (2014). Nyquist sampling theorem: Understanding the illusion of a spinning wheel captured with a video camera. *Physics Education*, 49(6), 697.
- Lu, Z., Wang, M., & Dai, W. (2020). A condition monitoring approach for machining process based on control chart pattern recognition with dynamically-sized observation windows. *Computers & Industrial Engineering*, 142, 106360. <https://doi.org/10.1016/j.cie.2020.106360>
- Maged, A., Haridy, S., Khoo, M. B. C., & Shamsuzzaman, M. (2021). Time between events monitoring for imperfect maintained systems with application to a robotic system. *Journal of Statistical Computation and Simulation*. <https://doi.org/10.1080/00949655.2021.1993224>
- Mehmood, R., Lee, M. H., Hussain, S., & Riaz, M. (2019). On efficient construction and evaluation of runs rules-based control chart for known and unknown parameters under different distributions. *Quality and Reliability Engineering International*, 35(2), 582–599. <https://doi.org/10.1002/qre.2423>
- Miao, Z., & Yang, M. (2019). *Control chart pattern recognition based on convolution neural network*. Paper presented at the smart innovations in communication and computational sciences, Singapore.
- Ranaee, V., & Ebrahimzadeh, A. (2011). Control chart pattern recognition using a novel hybrid intelligent method. *Applied Soft Computing*, 11(2), 2676–2686. <https://doi.org/10.1016/j.asoc.2010.10.016>
- 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. <https://doi.org/10.1007/s10044-011-0246-6>
- Rasmussen, C. E., & Nickisch, H. (2010). Gaussian processes for machine learning (GPML) toolbox. *The Journal of Machine Learning Research*, 11, 3011–3015.
- Santiago, E., & Smith, J. (2013). Control charts based on the exponential distribution: Adapting runs rules for the t chart. *Quality Engineering*, 25(2), 85–96.
- Shongwe, S. C. (2020). On the design of nonparametric runs-rules schemes using the Markov chain approach. *Quality and Reliability Engineering International*, 36(5), 1604–1621. <https://doi.org/10.1002/qre.2648>
- Snoek, J., Larochelle, H., & Adams, R. P. (2012). *Practical bayesian optimization of machine learning algorithms*. Paper presented at the advances in neural information processing systems.
- Victoria, A. H., & Maragatham, G. (2020). Automatic tuning of hyperparameters using Bayesian optimization. *Evolving Systems*. <https://doi.org/10.1007/s12530-020-09345-2>
- 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>
- Yager, R. R., & Filev, D. P. (1994). Generation of fuzzy rules by mountain clustering. *Journal of Intelligent & Fuzzy Systems*, 2(3), 209–219.
- Zaman, M., & Hassan, A. (2019). Improved statistical features-based control chart patterns recognition using ANFIS with fuzzy clustering. *Neural Computing and Applications*, 31(10), 5935–5949. <https://doi.org/10.1007/s00521-018-3388-2>
- Zan, T., Liu, Z., Su, Z., Wang, M., Gao, X., & Chen, D. (2020a). Statistical process control with intelligence based on the deep learning model. *Applied Sciences*, 10(1), 308.
- Zan, T., Liu, Z., Wang, H., Wang, M., & Gao, X. (2020b). Control chart pattern recognition using the convolutional neural network. *Journal of Intelligent Manufacturing*, 31(3), 703–716.
- Zhang, M., & Cheng, W. (2015). Recognition of mixture control chart pattern using multiclass support vector machine and genetic algorithm based on statistical and shape features. *Mathematical Problems in Engineering*.
- Zhang, Y., Shang, Y., Gao, N., & Wang, Q. (2017). Monitoring pre-specified changes in linear profiles using control charts with supplementary runs rules. *Communications in Statistics - Simulation and Computation*, 46(9), 7249–7263. <https://doi.org/10.1080/03610918.2016.1235190>
- Zhou, X., Jiang, P., & Wang, X. (2018). Recognition of control chart patterns using fuzzy SVM with a hybrid kernel function. *Journal of Intelligent Manufacturing*, 29(1), 51–67. <https://doi.org/10.1007/s10845-015-1089-6>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.