Dynamic Process Monitoring Using Machine Learning Control Charts

Xiulin Xie and Peihua Qiu

Abstract Machine learning methods have been widely used in different applications, including process control and monitoring. For handling statistical process control (SPC) problems, the existing machine learning approaches have some limitations. For instance, most of them are designed for cases in which in-control (IC) process observations at different time points are assumed to be independent and identically distributed. In practice, however, serial correlation almost always exists in the observed sequential data, and the longitudinal pattern of the process to monitor could be dynamic in the sense that its IC distribution would change over time (e.g., seasonality). It has been well demonstrated in the literature that control charts could be unreliable to use when their model assumptions are invalid. In this chapter, we modified some representative existing machine learning control charts using nonparametric longitudinal modeling and sequential data decorrelation algorithms. The modified machine learning control charts can well accommodate time-varying IC process distribution and serial data correlation. Numerical studies show that their performance are improved substantially for monitoring different dynamic processes.

Keywords Control chart · Data correlation · Dynamic processes · Machine learning · Seasonality · Statistical process control

1 Introduction

Statistical process control (SPC) provides a major tool for online monitoring of sequential processes [16, 22, 26]. Most conventional SPC charts are designed for detecting process distributional shifts under the assumptions that process observations at different time points are independent and identically distributed (i.i.d.) with a parametric (e.g., Normal) distribution when the process under monitoring is in-control (IC). In practice, however, observed data of a sequential process are often

X. Xie · P. Qiu (✉)
Department of Biostatistics, University of Florida, 2004 Mowry Road, Gainesville, FL 32610, USA
e-mail: pqiu@ufl.edu

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023

K. P. Tran (ed.), *Artificial Intelligence for Smart Manufacturing*, Springer Series in Reliability Engineering, https://doi.org/10.1007/978-3-031-30510-8_4

serially correlated, and dynamic in the sense that their IC distribution varies over time. This chapter focuses on online monitoring of dynamic processes with serially correlated data.

In the SPC literature, many control charts have been developed, which can be roughly classified into the following four categories: Shewhart, cumulative sum (CUSUM), exponentially weighted moving average (EWMA), and change-point detection (CPD) charts (cf., [17, 24, 30, 31]). As mentioned above, early control charts are designed mainly for cases when the observed IC data are i.i.d. and parametrically distributed. After SPC finds more and more applications for disease surveillance, environmental monitoring, business management, and many others, the conventional model assumptions mentioned above are rarely valid in these applications. It has been well demonstrated in the literature that control charts would be unreliable to use in cases when one or more of their model assumptions are invalid (e.g., [4, 20, 27]). So, some recent SPC research has considered cases when the IC process distribution does not have a parametric form (e.g., [10, 27]), process observations are serially correlated (e.g., [5, 29, 38]), or the IC process distribution is time-varying (e.g., [28, 36]).

In recent years, machine learning methods have been under rapid development (e.g., [1, 8, 14]). Since an SPC problem can be regarded as a binary class classification problem, in which each process observation needs to be classified into either the IC or the out-of-control (OC) status during sequential process monitoring, some machine leaning methods using both the IC and OC historical data have been used for process monitoring in the SPC literature. For instance, support vector machine (SVM), linear discriminant analysis (LDA), and k-nearest neighbors (KNN) methods have been employed for various process monitoring problems (i.e., [21, 39]). However, unlike the conventional classification problem, most SPC applications only involve IC training data before online process monitoring. To overcome this difficulty, some machine leaning algorithms, such as KNN, SVM, and random forest (RF), have been adapted to develop control charts using the one-class classification, artificial contrast, real-time contrast, and some other novel ideas (e.g., [12, 21, 34]). An attractive feature of these control charts based on machine learning algorithms is that they usually do not impose restrictive model assumptions explicitly. However, most of them require the implicit assumptions that process observations at different observation times are independent and identically distributed in order to define their decision rules properly. Therefore, such machine learning approaches have much room for improvement.

In Xie and Qiu [37], we modified some representative existing machine learning control charts so that the modified charts can properly accommodate serial correlation in process observations. However, these modified charts still assume the IC process distribution to be time-independent. Thus, they cannot be used in cases when the IC distribution is actually time-varying. In this chapter, we further modify the representative existing machine learning control charts considered in Xie and Qiu [37] so that the modified charts can accommodate both serial data correlation and time-varying IC process distribution, by using nonparametric longitudinal modeling and sequential data decorrelation algorithms. More specifically, in a modified control

chart, an IC dataset is required to obtain an initial estimate of the IC longitudinal pattern of the dynamic process under monitoring using a nonparametric longitudinal modeling approach. Then, at the current time point during online process monitoring, the observed data are first standardized using the estimated IC longitudinal pattern and then decorrelated with all historical data using a sequential data decorrelation algorithm. Next, a machine learning control chart is applied to the standardized and decorrelated data for making a decision whether the process has a distributional shift at the current time point or not. Numerical studies show that the modified machine learning control charts are substantially improved for monitoring different dynamic processes after such a modification.

The remaining parts of the chapter are organized as follows. In Sect. 2, some representative existing machine learning control charts are briefly described. In Sect. 3, the proposed modification for certain machine learning control charts are described in detail. Some simulation studies are presented in Sect. 4 to evaluate their numerical performance. A real-data example to demonstrate the application of the modified machine learning control charts is discussed in Sect. 5. Finally, some remarks conclude the chapter in Sect. 6.

2 Some Representative Machine Learning Control Charts

In this section, we introduce some representative recent machine learning control charts. Assume that $\mathbf{X} = (X_1, X_2, \ldots, X_p)'$ is a vector of $p \geq 1$ numerical quality characteristics to monitor about a sequential process, and its observation at time n is $\mathbf{X}_n = (X_{n1}, X_{n2}, \ldots, X_{np})'$. To online monitor the sequential process $\{\mathbf{X}_n, n \geq 1\}$, an initial IC dataset $\mathcal{X}_{IC} = \{\mathbf{X}_{-m_0+1}, \mathbf{X}_{-m_0+2}, \ldots, \mathbf{X}_0\}$ of size m_0 is assumed to be available in advance for all methods.

2.1 Control Chart Based on Artificial Contrasts

To solve a classification problem by a supervised machine learning method, a training dataset containing observations of both classes (e.g., IC and OC) is required. However, in many SPC applications, we only have an IC dataset before online process monitoring, and no OC process observations would be available in advance. To overcome this difficulty, Tuv and Runger [34] proposed the idea of artificial contrast. By this idea, artificial data are generated from a given distribution (e.g., Uniform) to represent the off-target data from the process. More specifically, for individual variables X_l, their artificial contrasts are generated independently from uniform distributions whose ranges are the same as those of X_l values in the IC dataset, for $l = 1, 2, \ldots, p$. Then, these artificial observations can be used as OC observations. By generating an artificial OC dataset, it converts the process monitoring problem to a supervised learning problem so that any machine learning classifiers, such as SVM and RF, can

be used. However, this type of control charts are basically Shewhart charts, since the decision at a given time point during online process monitoring only relied on the observation at that time point. To overcome these limitations, Hu and Runger [19] suggested a modification by using the ideas of generalized likelihood ratio test and EWMA. To make a decision about the status of the process under monitoring at the current time point n, the log likelihood ratio of observed data $\mathbf{X}_n$ is first calculated as $l_n = \log\left[\hat{p}_1(\mathbf{X}_n)\right] - \log\left[\hat{p}_0(\mathbf{X}_n)\right]$, for $n \geq 1$, where $\hat{p}_1(\mathbf{X}_n)$ and $\hat{p}_0(\mathbf{X}_n)$ are the estimated probabilities of $\mathbf{X}_n$ in each class obtained by the RF classifier. Then, they considered the following univariate EWMA charting statistic (cf., [30]):

$$E_n = \lambda l_n + (1 - \lambda)E_{n-1}, \text{ for } n \geq 1, \tag{1}$$

where $\lambda \in (0, 1]$ is a weighting parameter. The chart gives a signal of process mean shift at time n if

$$E_n > h_{AC}, \tag{2}$$

where h_{AC} is a control limit. The chart (1)–(2) is called AC chart hereafter to represent "artificial contrast."

For the AC chart (1)–(2), its control limit h_{AC} can be determined by a 10-fold cross-validation (CV) procedure to achieve a given value of the IC average run length (ARL), denoted as ARL_0. More specifically, 90% of the IC dataset $\mathcal{X}_{IC}$ and the artificial contrast dataset is first used to train the RF classifier. Then, a bootstrap sample can be drawn with replacement from the remaining 10% of the IC dataset, and the chart (1)–(2) with a given h_{AC} can be applied to the bootstrap sample. The run length (RL) value, defined to be the number of observation times from the beginning of process monitoring to the signal time, can then be recorded. Finally, the above procedure can be repeated for V times, and the average of the corresponding V values of RL can be used as the estimate of the ARL_0. Then, h_{AC} can be searched so that a given level of ARL_0 is reached. In this searching process, the bisection algorithm (Qiu [26], this chapter) or its modifications [9] can be used.

2.2 Control Chart Based on Real Time Contrasts

The classifier in the method AC is trained only one time using the IC dataset $\mathcal{X}_{IC}$ and the artificial OC dataset, which may not represent the actual off-target process observations well in a given application. To overcome this limitation, Deng et al. [12] suggested the so-called real-time contrast (RTC) method. The RTC method treats the process monitoring problem as a real-time classification problem, in which process observations in the IC dataset and those within a moving window of the current time point form a training dataset, with the former as IC observations and the latter as OC observations. More specifically, a dataset with N_0 observations, which is denoted as $\boldsymbol{S}_0$, is first randomly selected from the IC dataset $\mathcal{X}_{IC}$. Then, during online process monitoring, process observations in a window of the current observation time point

n are treated as OC data and denoted as $\boldsymbol{S}_n = \{\mathbf{X}_{n-w+1}, \mathbf{X}_{n-w+2}, \ldots, \mathbf{X}_n\}$, where w is the window size. Then, the RF classifier can be retrained sequentially overtime using the training dataset that combines $\boldsymbol{S}_0$ and $\boldsymbol{S}_n$. As discussed in Deng et al. [12], there could be several possible charting statistics based on the RF algorithm. As in their simulation studies, the average estimated classification rate in the dataset $\boldsymbol{S}_0$ can be used as the charting statistic, which is defined to be

$$R_n = \sum_{i=-m_0+1}^{0} \hat{p}_0^{(n)}(\mathbf{X}_i) I(\mathbf{X}_i \in \boldsymbol{S}_0)/N_0, \text{ for } n \geq 1, \tag{3}$$

where $\hat{p}_0^{(n)}(\mathbf{X}_i)$ are the estimated probabilities of $\mathbf{X}_i$ in the IC class obtained by the RF classifier trained at time n, and $I(u)$ is the indicator function that equals 1 when u is "true" and 0 otherwise. The chart gives a signal at time n if

$$R_n > h_{RTC}, \tag{4}$$

where h_{RTC} is a control limit of the RTC chart.

The control limit of the RTC chart (3)–(4) can be determined by the following bootstrap procedure suggested by Deng et al. [12]. First, we draw with replacement a sample from the IC dataset after the observations in $\boldsymbol{S}_0$ are excluded. Then, the chart (3)–(4) with the control limit h_{RTC} is applied to the bootstrap sample to obtain a RL value. This bootstrap re-sampling procedure is repeated for B = 1,000 times, and the average of the B values of RL is used to approximate the ARL_0 value for the given h_{RTC}. Finally, h_{RTC} can be searched by a numerical algorithm so that the assumed ARL_0 value is reached.

2.3 Control Chart Based on Support Vector Machine

Even though the RTC chart based on the RF classifier is useful and can be applied to a variety of monitoring problems, its charting statistic takes discrete values, which makes it less effective in some cases. As an alternative, He et al. [18] proposed a distance-based control chart. It uses the SVM framework to measure the distance between the support vectors and real time observations in $\boldsymbol{S}_n$. As discussed in He et al. [18], the distance from a sample of process observations to the boundary surface defined by the support vectors can be either positive or negative. They suggested transforming the distance using the following standard logistic function:

$$g(d) = \frac{1}{1 + \exp(-d)}.$$

Then, the following average value of the transformed distances from individual observations in $\boldsymbol{S}_n$ to the boundary surface can be defined to be the charting statistic:

$$M_n = \sum_{j=n-w+1}^{n} g(d(\mathbf{X}_i))/w, \text{ for } n \geq 1, \tag{5}$$

where $d(\mathbf{X}_i)$ is the distance from the observation $\mathbf{X}_i$ to decision boundary determined by the SVM classifier obtained at time n. The chart gives a signal at time n if

$$M_n > h_{SVM}, \tag{6}$$

where h_{SVM} is the control limit of the chart. The chart (5)–(6) is denoted as DSVM hereafter, to reflect the fact that it is a Distance-based control chart using SVM. The control limit of DSVM can be determined by a bootstrap procedure, similar to the one described above for the RTC chart.

In the above DSVM chart (5)–(6), the SVM algorithm needs to be implemented, and there are several qualititess involved that need to be selected in advance, including the kernel function and the penalty parameter [11]. In SVM, one of the most commonly used kernel functions is the Gaussian radial basis function (RBF), which is defined as: for any two observations $\mathbf{X}_i, \mathbf{X}_j$,

$$G(\mathbf{X}_i, \mathbf{X}_j) = \exp\left(\frac{\|\mathbf{X}_i - \mathbf{X}_j\|^2}{\sigma^2}\right),$$

where σ^2 is the spread parameter. He et al. [18] suggested using the above RBF as the kernel function with $\sigma^2 > 2.8$. They also suggested choosing the penalty parameter to be 1 for training SVM.

2.4 Control Chart Based on the KNN Classification

Another machine learning control chart, proposed by Sukchotrat et al. [33], is based on the KNN data description procedure. This chart is denoted as KNN hereafter. The charting statistic of KNN is defined as the average distance between a given observation $\boldsymbol{X}_n$ and its k nearest observations in the IC dataset $\mathcal{X}_{IC}$, and it is defined as follows:

$$C_n^2 = \sum_{j=1}^{k} \|\mathbf{X}_n - N_j(\mathbf{X}_n)\|/k, \text{ for } n \geq 1, \tag{7}$$

where $N_j(\mathbf{X}_n)$ is the jth nearest neighboring observation of $\mathbf{X}_n$ in the IC dataset $\mathcal{X}_{IC}$, and $\|\cdot\|$ is the Euclidean distance. Then, for online process monitoring, the process is declared to be OC at a given time n if the charting statistic C_n^2 of the related process observation exceeds the control limit h_{KNN}.

In the above KNN chart, the control limit h_{KNN} can be determined by the following bootstrap procedure: (i) a total of $B = 1,000$ bootstrap samples are obtained from

the training dataset by the random sampling procedure with replacement and each bootstrap sample has the same size as the training dataset, (ii) the C_n^2 values defined in (7) of the individual observations in the bootstrap sample can be computed, (iii) the $(1-\alpha)$th percentile of all C_n^2 values can be computed from each bootstrap sample, and (iv) h_{KNN} is chosen to be the mean of the B such percentiles.

3 Suggested Modified Machine Learning Control Charts for Dynamic Process Monitoring

For many longitudinal processes, their distributions could change over time, even when their performance is considered to be IC. One example is about sequential monitoring of environmental variables, such as air temperature and various pollutant levels. These variables usually have seasonal variation. To monitor such dynamic processes, the machine learning control charts introduced in the previous section are obviously inappropriate to use because they require the IC process distribution to be unchanged over time. Recently, Xie and Qiu [36] suggested a new method for dynamic process monitoring. The basic idea of that method is to specify a time period as a baseline time period, estimate the regular longitudinal pattern of the quality variables in that period, and then compare the future performance of the process under monitoring with its performance in the baseline time period. In this section, a procedure for estimating the regular longitudinal pattern is first discussed in detail. Then, the suggested modification of some representative machine learning control charts for monitoring dynamic processes using is discussed.

3.1 Estimation of the Regular Multivariate Longitudinal Pattern

The time period of the initial IC dataset $\mathcal{X}_{IC}$ is set as a baseline time interval, and the IC dataset is assumed to follow the nonparametric longitudinal model:

$$\mathbf{X}_j = \boldsymbol{\mu}_j + \boldsymbol{\epsilon}_j, \qquad \text{for } j = -m_0+1, -m_0+2, \ldots, 0, \tag{8}$$

where $\boldsymbol{\mu}_j = (\mu_{j1}, \mu_{j2}, \ldots, \mu_{jp})'$ is the mean of $\mathbf{X}_j$, and $\boldsymbol{\epsilon}_j$ is the p-dimensional zero-mean error term. In Model (8), the covariance structure is described by $\text{Cov}(\boldsymbol{\epsilon}_j, \boldsymbol{\epsilon}_{j^*})$, for any $j, j^* \in [-m_0+1, 0]$. Furthermore, it is assumed that the serial correlation among the IC process observations is stationary, and the serial correlation exists only when two observations are within $b_{\max} > 0$ in their observation indices. More specifically, it is assumed that $\boldsymbol{\gamma}(s) = \text{Cov}(\boldsymbol{\epsilon}_j, \boldsymbol{\epsilon}_{j+s})$ only depends on s when j changes, and $\boldsymbol{\gamma}(s) = \mathbf{0}$ when $s > b_{\max}$. The above assumptions should be reasonable in many applications.

To obtain an initial estimate of $\boldsymbol{\mu}_j$, we can compute the local linear kernel (LLK) smoothing estimates of all components of $\boldsymbol{\mu}_j$ (cf., [35]). In matrix notation, let $\mathbf{W} = (X_{-m_0+1,1}, \ldots, X_{0,1}, \ldots, X_{-m_0+1,p}, \ldots, X_{0,p})'$, $\mathbf{Z}_j = \left[(1, -m_0+1-j)', \ldots, (1, -j)'\right]'$, and $\mathbf{K}_j = \text{diag}\{K(\frac{i-j}{h_l}), i = -m_0+1, -m_0+2, \ldots, 0, l = 1, 2, \ldots, p\}$, where $K(\cdot)$ is a kernel function and $\{h_l, l = 1, 2, \ldots, p\}$ are bandwidths. Then, the initial estimate of $\boldsymbol{\mu}_j$, for $j = -m_0+1, -m_0+2, \ldots, 0$, can be obtained by the following LLK smoothing procedure:

$$\min_{\boldsymbol{\beta} \in R^{2p}} \left[\mathbf{W} - (I_{p\times p} \otimes \mathbf{Z})\boldsymbol{\beta}\right]' \mathbf{K}_j \left[\mathbf{W} - (I_{p\times p} \otimes \mathbf{Z})\boldsymbol{\beta}\right], \tag{9}$$

where $\otimes$ denotes the Kronecker product, $I_{p\times p}$ is the $p \times p$ identity matrix, and $\boldsymbol{\beta} = (\beta_{01}, \beta_{11}, \ldots, \beta_{0p}, \beta_{1p})'$ are coefficients. The solution of (9) has the expression

$$\widehat{\boldsymbol{\beta}} = [(I_{p\times p} \otimes \mathbf{Z}_j)'\mathbf{K}_j(I_{p\times p} \otimes \mathbf{Z}_j)]^{-1}(I_{p\times p} \otimes \mathbf{Z}_j)'\mathbf{K}_j\mathbf{W}.$$

Then, the initial estimate of $\boldsymbol{\mu}_j$, for $j = -m_0+1, -m_0+2, \ldots, 0$, is given by:

$$\widehat{\boldsymbol{\mu}}_j = \widehat{\boldsymbol{\beta}}'(I_{p\times p} \otimes \boldsymbol{\xi}_1), \tag{10}$$

where $\boldsymbol{\xi}_1 = (1, 0)'$. In the above LLK procedure, the kernel function $K(\cdot)$ is usually chosen to be the Epanechnikov kernel function, i.e., $K(u) = \frac{3}{4}(1-u^2)I(|u| \leq 1)$, because of its good properties [13]. For the bandwidths $\{h_l, l = 1, 2, \ldots, p\}$, it has been well discussed in the literature that the conventional cross-validation (CV) procedure would not perform well when process observations at different time points are serially correlated, since the CV procedure cannot properly distinguish the data correlation structure from the data mean function (e.g., [2, 23]). Thus, we suggest choosing them using the following modified cross-validation (MCV) procedure that was originally suggested by Brabanter et al. [7] for handling bandwidth selection in a univariate regression setup with correlated data. By this approach, the bandwidths $\{h_l, l = 1, \ldots, p\}$ can be chosen by minimizing the following MCV score:

$$\text{MCV}(h_1, h_2, \ldots, h_p) = \frac{1}{m_0} \sum_{j=-m_0+1}^{0} \left(\mathbf{X}_j - \widehat{\boldsymbol{\mu}}_{-j}\right)' \left(\mathbf{X}_j - \widehat{\boldsymbol{\mu}}_{-j}\right),$$

where $\widehat{\boldsymbol{\mu}}_{-j}$ is the leave-one-out estimate of $\boldsymbol{\mu}_j$ by (10) when the observation $\mathbf{X}_j$ is excluded in the computation and when the kernel function $K(\cdot)$ is modified to be

$$K_\varepsilon(u) = \frac{4}{4 - 3\varepsilon - \varepsilon^3} \begin{cases} \frac{3}{4}(1-u^2)I(|u| \leq 1), & \text{when } |u| \geq \varepsilon, \\ \frac{3(1-\varepsilon^2)}{4\varepsilon}|u|, & \text{when } |u| < \varepsilon, \end{cases}$$

where $\varepsilon \in (0, 1)$ is a small constant. The modified kernel function $K_\varepsilon(u)$ equals 0 at $u = 0$ and is small around $u = 0$, to diminish the impact of data autocorrelation on bandwidth selection.

As mentioned above, two original process observations are allowed to be correlated if their observation times are within $b_{\max}$ apart and the serial correlation is assumed to be stationary. Then, the covariance matrices $\boldsymbol{\gamma}(s)$, for $0 \le s \le b_{\max}$, can be estimated by the following moment estimates:

$$\widehat{\boldsymbol{\gamma}}(s) = \frac{1}{m_0 - s} \sum_{j=-m_0+1}^{-s} \left(\mathbf{X}_{j+s} - \widehat{\boldsymbol{\mu}}_{j+s}\right)\left(\mathbf{X}_j - \widehat{\boldsymbol{\mu}}_j\right)', \qquad \text{for } 0 \le s \le b_{\max}.$$

3.2 Dynamic Process Monitoring

Next, we discuss online monitoring of the p-dimensional dynamic process with the observations $\{\mathbf{X}_n, n \ge 1\}$. When the process is IC, it is assumed that it follows the regular longitudinal pattern described by Model (8) in the sense that

$$\mathbf{X}_n = \boldsymbol{\mu}_n + \boldsymbol{\epsilon}_n, \qquad \text{for } n \ge 1, \tag{11}$$

where $\boldsymbol{\mu}_n = \boldsymbol{\mu}_{n^*}$, n^* is an integer in $[-m_0 + 1, 0]$, $n = n^* + Tm_0$, $T \ge 1$ is an integer, and the error term $\boldsymbol{\epsilon}_n$ has the same covariance structure as that in Model (8).

Then, for monitoring dynamic processes using machine learning control charts, we suggest first standardizing the observed data at the current time point using the estimated IC longitudinal pattern in (10), and then decorrelating the observed data with historical data. After a proper data standardization and decorrelation of the observed data, a machine learning control chart can be used to make a decision whether the process is IC or not at the current time point. The modified machine learning control charts for monitoring dynamic processes with serial data correlation can then be summarized below.

Proposed Dynamic Process Monitoring Scheme using Machine Learning Control Charts

Step 1 **Initial Estimation**: Obtain the initial estimates $\{\widehat{\boldsymbol{\mu}}_j, -m_0 + 1 \le j \le 0\}$ and $\{\widehat{\boldsymbol{\gamma}}(s), 0 \le s \le b_{\max}\}$ from the initial IC data $\mathcal{X}_{IC}$, as discussed in Sect. 3.1.

Step 2 **Data Standardization and Decorrelation**: At the current time point n, if $n = 1$, then define the standardized observation to be

$$\mathbf{e}_1^* = [\widehat{\boldsymbol{\gamma}}(0)]^{-1/2} \left(\mathbf{X}_1 - \widehat{\boldsymbol{\mu}}_1\right).$$

Otherwise, the estimated covariance matrix of $(\mathbf{X}'_{n-b}, \mathbf{X}'_{n-b+1}, \ldots, \mathbf{X}'_n)'$ is defined to be

$$\widehat{\boldsymbol{\Sigma}}_{n,n} = \begin{pmatrix} \widehat{\boldsymbol{\gamma}}(0) & \cdots & \widehat{\boldsymbol{\gamma}}(b) \\ \vdots & \ddots & \vdots \\ \left[\widehat{\boldsymbol{\gamma}}(b)\right]' & \ldots & \widehat{\boldsymbol{\gamma}}(0) \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{\Sigma}}_{n-1,n-1} & \widehat{\boldsymbol{\Sigma}}_{n-1,n} \\ \widehat{\boldsymbol{\Sigma}}'_{n-1,n} & \widehat{\boldsymbol{\gamma}}(0) \end{pmatrix},$$

where $b = \min(n-1, b_{\max})$. Then, the decorrelated and standardized observation at time n is defined to be

$$\mathbf{e}_n^* = \widehat{\mathbf{D}}_n^{-1/2}\left[-\widehat{\boldsymbol{\Sigma}}'_{n-1,n}\widehat{\boldsymbol{\Sigma}}_{n-1,n-1}^{-1}\widehat{\mathbf{e}}_{n-1} + (\mathbf{X}_n - \widehat{\boldsymbol{\mu}}_n)\right],$$

where $\widehat{\mathbf{D}}_n = \widehat{\boldsymbol{\gamma}}(0) - \widehat{\boldsymbol{\Sigma}}'_{n-1,n}\widehat{\boldsymbol{\Sigma}}_{n-1,n-1}^{-1}\widehat{\boldsymbol{\Sigma}}_{n-1,n}$, and $\widehat{\mathbf{e}}_{n-1} = [(\mathbf{X}_{n-b} - \widehat{\boldsymbol{\mu}}_{n-b})', (\mathbf{X}_{n-b+1} - \widehat{\boldsymbol{\mu}}_{n-b+1})', \ldots, (\mathbf{X}_{n-1} - \widehat{\boldsymbol{\mu}}_{n-1})']'$.

Step 3 **Decision-Making**: Apply a machine learning control chart to the decorrelated and standardized data $\{\mathbf{e}_n^*, n \geq 1\}$ to see whether a signal is triggered.

4 Simulation Studies

In this section, we investigate the numerical performance of the four existing machine learning control charts AC, RTC, DSVM and KNN described in Sect. 2, in comparison with their two modified versions. The first modified version of these four control charts are denoted as AC-D-WOC, RTC-D-WOC, DSVM-D-WOC and KNN-D-WOC, where "D" indicates that the Dynamic nature of the process under monitoring is considered in the chart, and "WOC" represents 'WithOut considering serial Correlation. The second modified version of these four control charts are denoted as AC-D-C, RTC-D-C, DSVM-D-C and KNN-D-C, where the last letter "C" denotes serial Correlation has been considered. This modified version is discussed in Sect. 3.2. In all simulation examples, the nominal ARL_0 values of all charts are fixed at 200. The number of quality characteristics is set to be $p = 5$, and the parameter $b_{\max}$ is chosen to be 15. Regarding the IC process distribution and the IC serial data correlation, the following four cases are considered:

Case I: IC process observations $\{\mathbf{X}_n, n \geq 1\}$ are i.i.d. with the IC distribution $N_p(\mathbf{0}, \boldsymbol{I}_p)$.

Case II: IC process observations $\{\mathbf{X}_n, n \geq 1\}$ are generated from Model (11). Their mean and correlation structures are specified in Model (8), where the means are defined to be

$$\boldsymbol{\mu}_j = \left[sin(2\pi t_j), cos(2\pi t_j), sin^2(2\pi t_j), cos^2(2\pi t_j), sin(2\pi t_j) + cos(2\pi t_j)\right],$$

$t_j = (j + m_0)/m_0$ for $j = -m_0 + 1, -m_0 + 2, \ldots, 0$, each component of the error term $\boldsymbol{\epsilon}_j$ has the standardized χ_3^2 distribution, and the covariance matrix of $\boldsymbol{\epsilon}_j$ is $\boldsymbol{I}_p$.

Case III: Same as Case II, except that the error terms $\{\boldsymbol{\epsilon}_j\}$ are assumed to follow the vector-AR(1) model $\boldsymbol{\epsilon}_j = 0.2\boldsymbol{\epsilon}_{j-1} + \boldsymbol{\eta}_j$, where $\boldsymbol{\epsilon}_0 = \mathbf{0}$, each component of $\boldsymbol{\eta}_j$ has the standardized χ_3^2 distribution, and the covariance matrix of $\boldsymbol{\eta}_j$ is $\boldsymbol{I}_p$.

Case IV: Same as Case III, except that the covariance matrix of $\boldsymbol{\eta}_j$ is $\boldsymbol{\Sigma} = (\sigma_{l_1 l_2})_{p\times p}$ with $\sigma_{l_1 l_2} = 0.5^{|l_1 - l_2|}$, for $l_1, l_2 = 1, 2, \ldots, p$.

For the four cases described above, Case I is the conventional case with i.i.d. mean **0** IC process observations and the normal IC process distribution. Cases II–IV consider three different dynamic processes. The dynamic process in Case II still has independent observations at different observation times, and the p quality variables are independent with each other as well. The dynamic process in Case III is the same as that in Case II, except that process observations are serially correlated. Dynamic process observations in Case IV are serially correlated and the p quality variables are mutually associated as well.

4.1 Evaluation of the IC Performance

We first evaluate the IC performance of the related control charts. In the simulation study, the IC sample size m_0 is fixed at 2,000. The weighting parameter λ in the chart AC and its two modified versions are chosen to be 0.2, as suggested in Hu and Runger [19], the moving window size w in the charts RTC and DSVM and their modified versions are chosen to be 10, as suggested in Deng et al. [12] and He et al. [18], and the number of nearest observations k in the chart KNN and its modified versions is chosen to be 30, as suggested in Sukchotrat et al. [33]. For each method, its actual ARL_0 value is computed as follows. First, an IC dataset of size m_0 is generated from the IC model, and the IC parameters are estimated from the IC data. Second, the conditional ARL_0 value of the chart given the IC dataset is calculated based on 1,000 replicated simulations of online process monitoring of 2,000 sequential process observations. Third, the previous two steps are repeated for 100 times, and the sample average of the 100 conditional ARL_0 values is used as the estimated actual ARL_0 value of the chart. The standard error of the estimated actual ARL_0 value can also be computed as the standard deviation of the 100 conditional ARL_0 values divided by $\sqrt{100}$. The estimated ARL_0 values in different cases considered are shown in Table 1.

From Table 1, we can have the following conclusions. (i) The four original machine learning control charts all have a reasonable performance in Case I when the process observations are i.i.d. with a normal distribution, but they are unreliable to use in all other cases when some or all of these assumptions are violated because their estimated actual ARL_0 values are substantially different from the nominal ARL_0 level of 200 in these cases. (ii) The first modified version of the four machine learning control charts AC-D-WOC, RTC-D-WOC, DSVM-D-WOC, and KNN-D-WOC perform well in Cases I and II when the independence assumption is valid, but their performance is quite poor in Cases III and IV when this assumption is violated. (iii) As a comparison, the second modified version of the four machine learning control charts AC-D-C, RTC-D-C, DSVM-D-C, and KNN-D-C have a reasonably good performance in all cases considered, since its estimated actual ARL_0 values are always within 10% of the nominal ARL_0 level. Therefore, this example confirms that the IC performance of the machine learning control charts can be improved in a substantial way by using the suggested modification discussed in Sect. 3.2.

Table 1 Actual ARL_0 values and their standard errors (in parentheses) of four machine learning control charts and their modified versions when their nominal ARL_0 values are fixed at 200, $p = 5$, and $m_0 = 2,000$

Methods	Case I	Case II	Case III	Case IV
AC	207(4.22)	53.7(1.40)	44.7(1.44)	42.8(1.37)
AC-D-WOC	195(4.7)	187(3.77)	107(3.38)	110(3.44)
AC-D-C	203(3.96)	193(3.65)	189(3.42)	191(3.74)
RTC	191(3.88)	103(2.33)	89.6(1.74)	90.4(1.83)
RTC-D-WOC	187(3.76)	194(4.27)	142(2.95)	139(2.87)
RTC-D-C	189(3.69)	192(4.08)	206(4.93)	209(5.03)
DSVM	210(4.19)	125(2.67)	113(2.05)	110(2.56)
DSVM-D-WOC	193(4.30)	196(4.22)	147(3.03)	148(2.99)
DSVM-D-C	191(4.04)	202(3.99)	194(4.12)	195(4.15)
KNN	208(4.36)	141(2.45)	143(2.56)	139(2.41)
KNN-D-WOC	205(4.27)	190(3.15)	170(3.27)	167(2.95)
KNN-D-C	189(3.96)	193(4.20)	202(4.32)	204(4.41)

4.2 Evaluation of the OC Performance

Next, we evaluate the OC performance of the related charts in case when $m_0 = 2,000$. In order to make the comparison more meaningful, we intentionally adjust the control limits of different control charts so that their actual ARL_0 values equal the nominal ARL_0 value of 200 in all cases considered. In the next simulation example, it is assumed that all quality variables have a same shift at the beginning of online process monitoring with the shift size δ changing from 0 to 1 with a step of 0.25. Because different control charts have different procedure parameters (e.g., the moving window sizes of RTC and DSVM) and their performance may not be comparable if their parameters are set to be the same, here we compare their optimal OC performance to make the comparison fair. Namely, to detect a given shift by a chart, the related procedure parameter is chosen by minimizing the OC average run length, denoted as ARL_1, of the chart while maintaining its ARL_0 value at 200. The resulting ARL_1 value is called optimal ARL_1 value hereafter. The results of the optimal ARL_1 values of these machine learning control charts and their modified versions in Cases I–IV are presented in Fig. 1.

From the figure, we can have the following conclusions. First, all four machine learning control charts and their two modified versions perform reasonably well in Case I when the process observations are i.i.d. with a normal distribution, since their model assumptions are all satisfied. Second, The first modified version of the related control charts AC-D-WOC, RTC-D-WOC, DSVM-D-WOC, and KNN-D-WOC are the most effective one among the three version of all charts in Case II when the independence assumption is valid, but are less effective in Cases III and IV when this assumption is invalid. Third, the second modified version of the four machine learning control charts AC-D-C, RTC-D-C, DSVM-D-C, and KNN-D-C have the

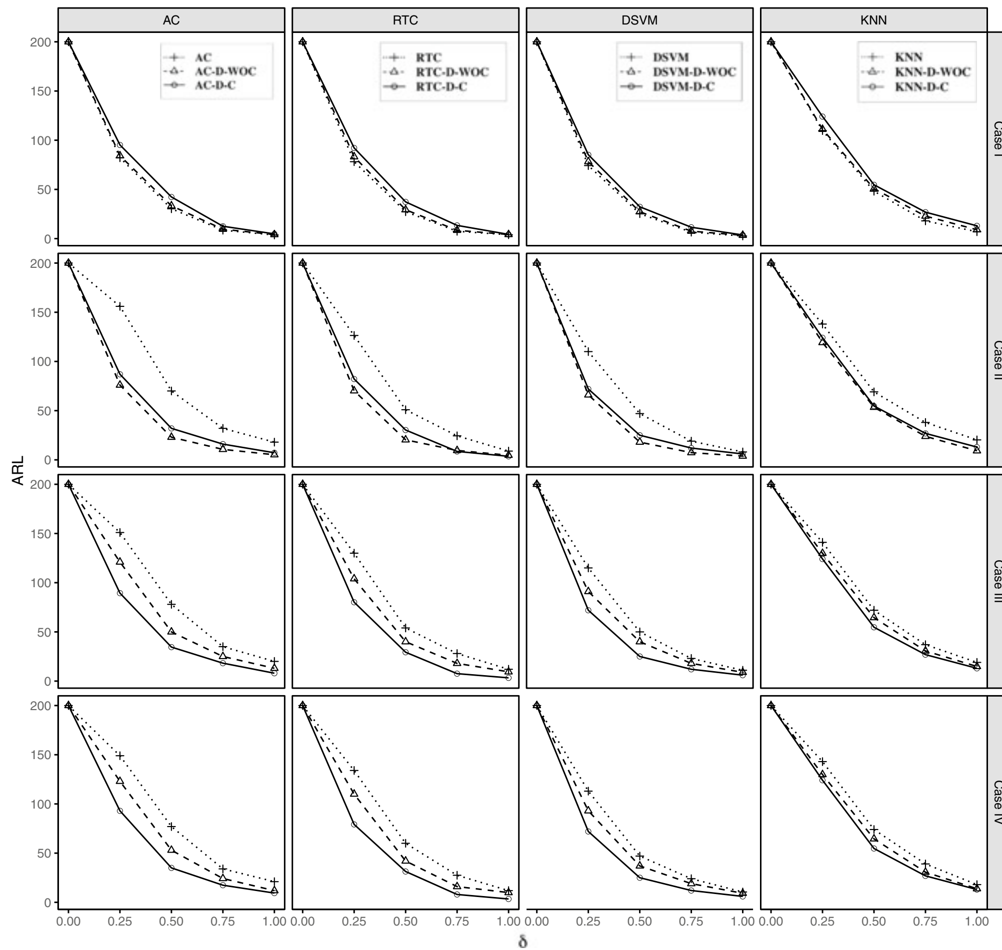

Fig. 1 Optimal ARL_1 values of the four control charts and their modified versions when their nominal ARL_0 values are fixed at 200, $p = 5$, $m_0 = 2{,}000$, and all quality characteristics have the same shift with the shift size δ changing among 0.25, 0.5, 0.75, and 1

best performance among the three versions of all charts in Cases III and VI when the process under monitoring is dynamic with serial data correlation.

5 An Application

In this section, we demonstrate the application of the modified machine learning control charts discussed in Sects. 3 and 4 using a real dataset, which contains electricity generation and weather data in Spain. This dataset can be downloaded from the web page of Kaggle with the link https://www.kaggle.com/datasets/nicholasjhana/energy-consumption-generation-prices-and-weather. Electricity is generated using a variety of resources, including coal, natural gas, nuclear energy, and solar energy, and its usage usually depends on the meteorological conditions [32]. For examples,

the colder months often bring more electricity usage as more electricity is spent on the heating system. However, when excess electricity is generated, much time and resources would be wasted because electricity cannot be stored in large quantities efficiently [25]. Therefore, it is important to online monitor the electricity generation and demand in the electric industry. If something unusual happens (e.g., unseasonable cold weather), the electric utility companies can take actions quickly to adjust the amount of electricity generated to meet demand. In this analysis, the amount of electricity generated by three most common energy sources, including gas, coal and oil, and two important environmental variables, i.e., temperature and humidity are considered. The dataset used here contains observations of the five variables during a time period from January 1, 2015 to December 31, 2016. The original data of these five variables are shown in Fig. 2. From the figure, it can be seen that there is a quite

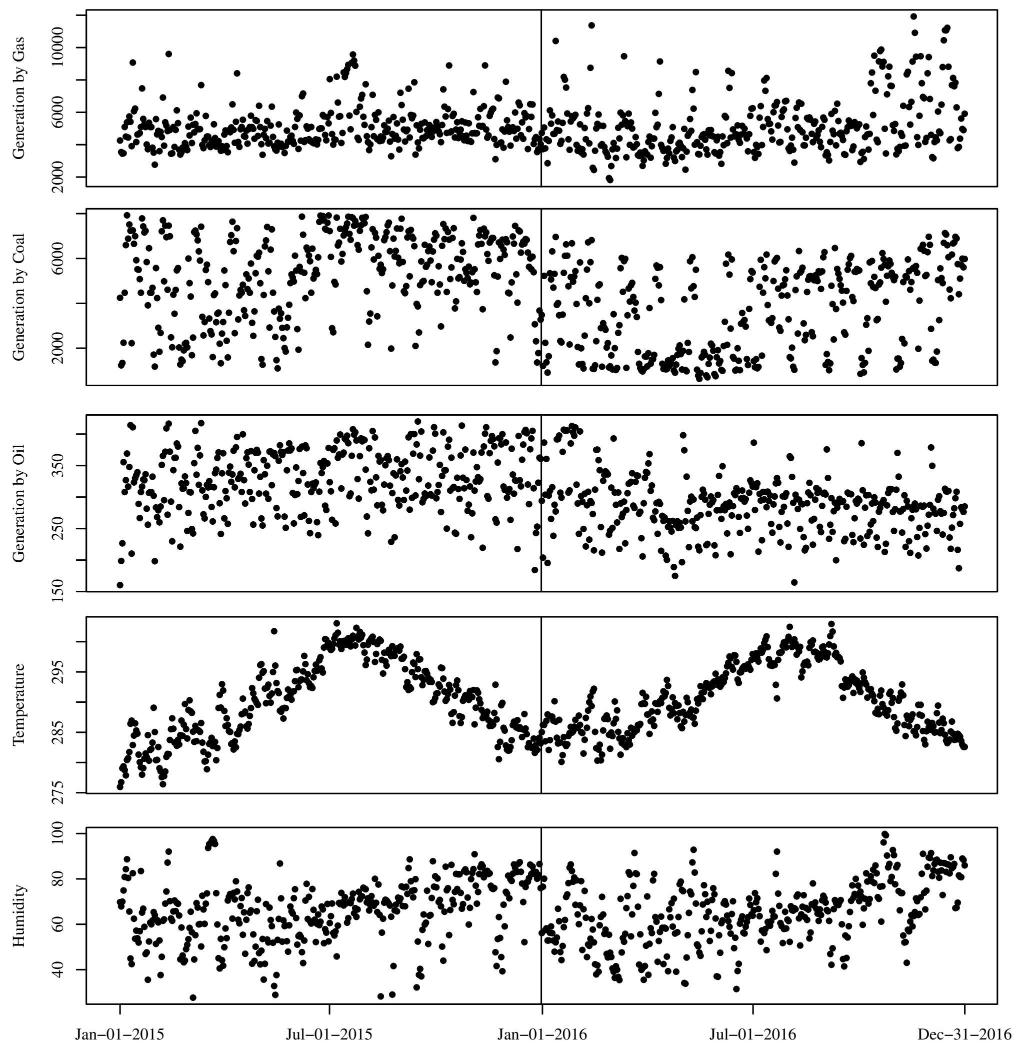

Fig. 2 Original observations of five variables considered in the electricity example. The solid vertical line in each plot separates the initial IC data from the data for online process monitoring

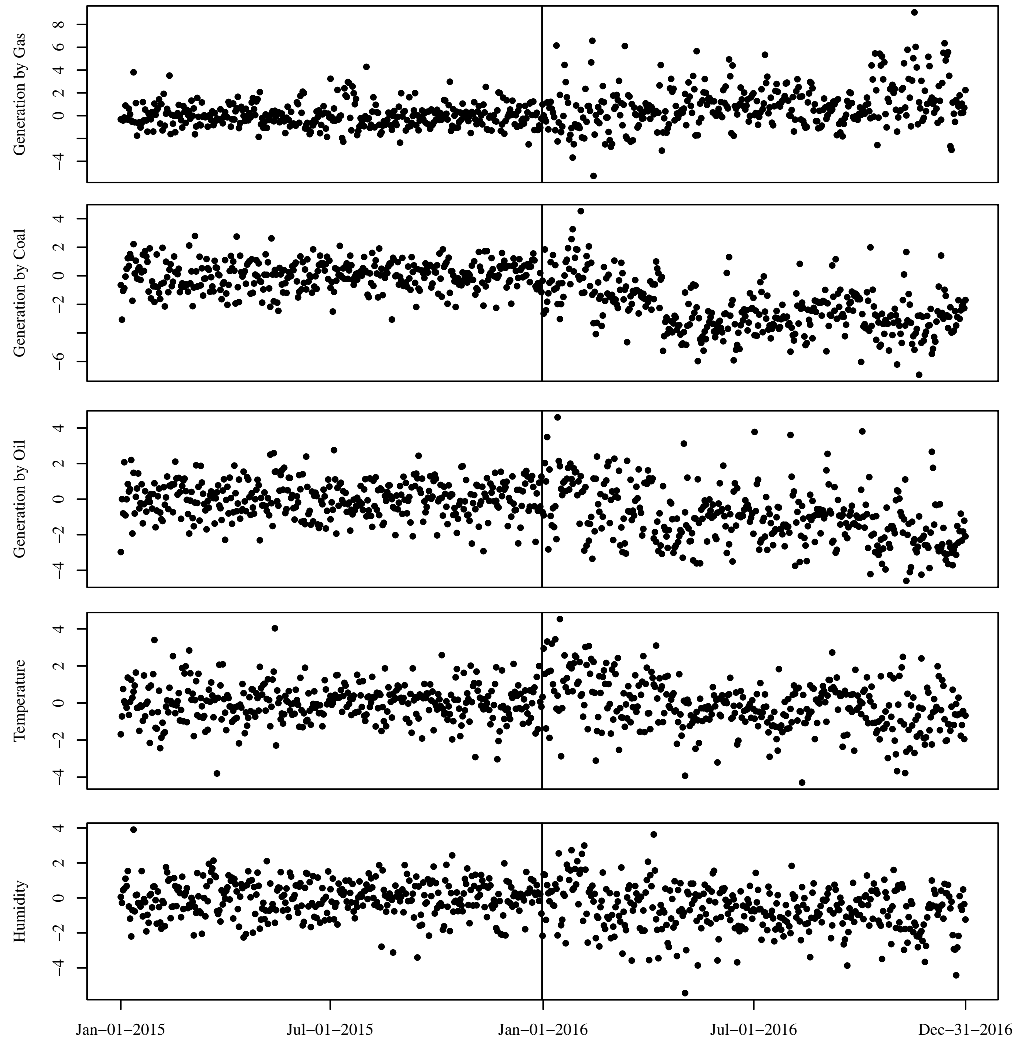

Fig. 3 Standardized and decorrelated observations of five variables considered in Fig. 2. The solid vertical line in each plot separates the initial IC data from the data for online process monitoring

obvious yearly seasonality in the observed data, the temperature is higher during summer times, and the amount of electricity generated by coal seems higher in the last six months of each year. In our analysis, the data in the first year are used as the IC data for estimating the regular longitudinal pattern of the five variables, and the data in the second year are used for online process monitoring.

For the IC data, we first compute the initial LLK estimates $\widehat{\boldsymbol{\mu}}_j$ by (10), and then obtain the residuals $\mathbf{X}_j - \widehat{\boldsymbol{\mu}}_j$, for all j. Next, we use the Ljung-Box test for checking serial data correlation in the residuals of each variable. The p-values of this test are all $< 2.3 \times 10^{-9}$ for the five variable. Thus, there is a significant autocorrelation in the IC data. The Augmented Dickey-Fuller (ADF) test for stationarity of the autocorrelation gives p-values of <0.01 for all five variables, which implies that the stationarity assumption is valid in this case. To check the normality assumption for

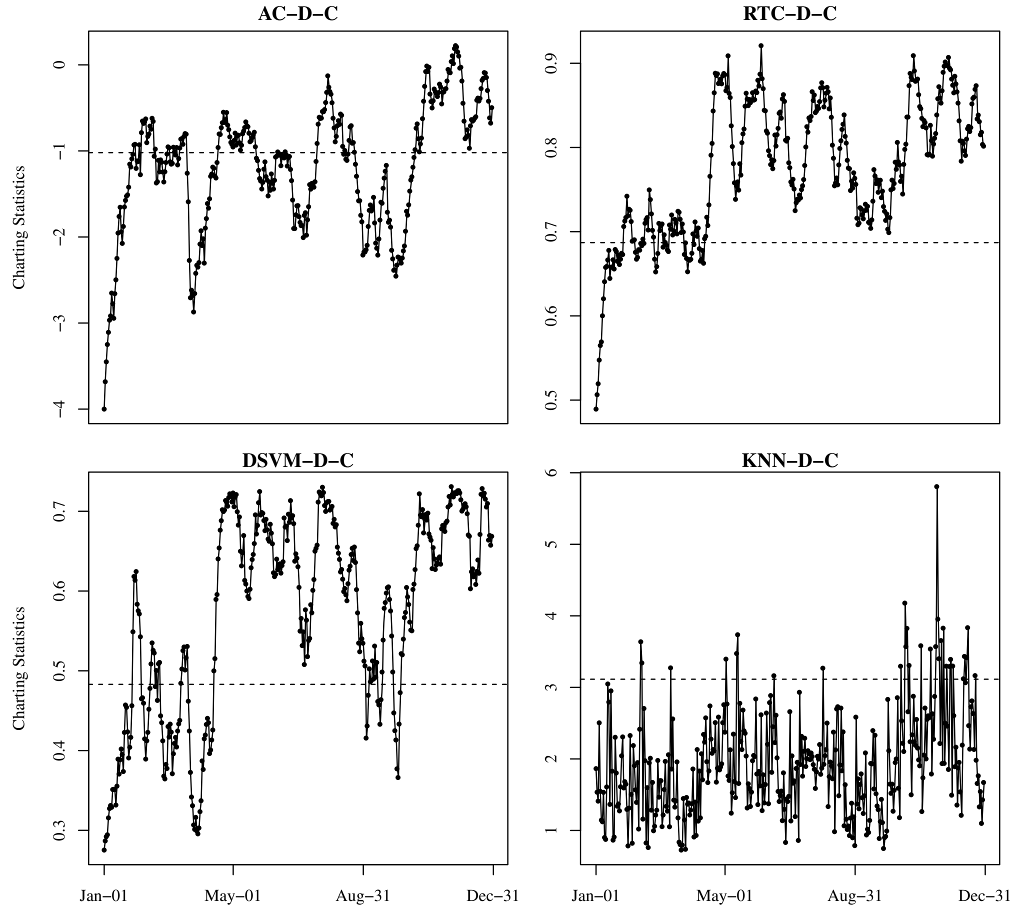

Fig. 4 Control charts for online monitoring of the data during January 1 and December 31, 2016. In each plot, the horizontal dashed line denotes the control limit of the related control chart

the data, the Shapiro test is performed, and it gives a p-values of 2.9×10^{-7}, which implies that the distribution of the standardized IC data is significantly different from a normal distribution. Therefore, the four modified control charts AC-D-C, RTC-D-C, DSVM-D-C and KNN-D-C should be appropriate to use in this example, because the IC data have a dynamic pattern, significant stationary serial data correlation, and a non-normal distribution. The standardized and decorrelated data of the five variables by the procedure discussed in Sect. 3.2 are shown in Fig. 3, from which it can be seen that the standardized and decorrelated IC data are indeed quite stable, and the the standardized and decorrelated data in the second year seem to be quite different from the IC data starting from the very beginning of the second year.

Next, we apply the four charts AC-D-C, RTC-D-C, DSVM-D-C and KNN-D-C to this dataset for online process monitoring starting from January 1, 2016. In all control charts, the nominal ARL_0 values are fixed at 200, and their control limits are computed in the same way as that in the simulation study of Sect. 4. All four control charts are shown in Fig. 4. From the plots in the figure, the charts AC-D-C, RTC-D-C,

DSVM-D-C and KNN-D-C give their first signals on the Jan 30th, Jan 28th, Jan 27th, and Feb 13th, respectively. By checking the standardized and decorrelated process observations shown in Fig. 3, it seems that all fours chart can detect a systematic change in the process well and the chart DSVM-D-C gives the earliest signal among them.

6 Concluding Remarks

Some control charts based on machine learning approaches have been developed recently in the SPC literature. However, most existing machine learning control charts are based on the assumptions that the process observations at different time points are independent and identically distributed. So, they would be unreliable to use in case when the IC process distribution changes over time. In this chapter, we have suggested a modification procedure for some representative existing machine learning control charts using the nonparametric longitudinal modeling and sequential decorrelation algorithms. Numerical studies show that the performance of these modified control charts is substantially better than their original versions in cases when the IC process distribution is time-varying.

There are still some issues about the modified machine learning control charts that need to be addressed in our future research. For example, these machine learning methods require a relatively large IC dataset. But, in some applications, a relatively large IC dataset may not be available. In such cases, self-starting control charts might be helpful (cf., [15]). In addition, the current proposed methods assume that the serial correlation in process observations is short-ranged and stationary. Even through these assumptions should be reasonable in many applications, the serial data correlation could be long-range and non-stationary in some other applications (cf., [3, 6]).

References

1. Aggarwal CC (2018) Neural networks and deep learning. Springer, New Yorker
2. Altman NS (1990) Kernel smoothing of data with correlated errors. J Am Stat Assoc 85:749–758
3. Altmann EG, Cristadoro G, Esposti MD (2012) On the origin of long-range correlations in texts. Proc Natl Acad Sci USA 109:11582–11587
4. Apley DW, Lee HC (2008) Robustness comparison of exponentially weighted moving-average charts on autocorrelated data and on residuals. J Qual Technol 40:428–447
5. Apley DW, Tsung F (2002) The autoregressive T^2 chart for monitoring univariate autocorrelated processes. J Qual Technol 34:80–96
6. Beran J (1992) Statistical methods for data with long-range dependence. Stat Sci 4:404–416
7. Brabanter KD, Brabanter JD, Suykens JAK, De Moor B (2001) Kernel regression in the presence of correlated errors. J Mach Learn Res 12:1955–1976
8. Breiman L (2001) Random forests. Mach Learn 45:5–32
9. Capizzi G, Masarotto G (2016) Efficient control chart calibration by simulated stochastic approximation. IIE Trans 48:57–65

10. Chakraborti S, Graham MA (2019) Nonparametric (distribution-free) control charts: an updated overview and some results. Qual Eng 31:523–544
11. Cortes C, Vapnik VN (1995) Support-vector networks. Mach Learn 20:273–297
12. Deng H, Runger G, Tuv E (2012) System monitoring with real-time contrasts. J Qual Technol 44:9–27
13. Epanechnikov VA (1969) Non-parametric estimation of a multivariate probability density. Theory Probab Appl 14:153–158
14. Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning—Data mining, inference, and prediction. Springer, Berlin
15. Hawkins DM (1987) Self-starting CUSUMS for location and scale. Statistician 36:299–315
16. Hawkins DM, Olwell DH (1998) Cumulative sum charts and charting for quality improvement. Springer, New York
17. Hawkins DM, Qiu P, Kang CW (2003) The changepoint model for statistical process control. J Qual Technol 35:355–366
18. He S, Jiang W, Deng H (2018) A distance-based control chart for monitoring multivariate processes using support vector machines. Ann Oper Res 263:191–207
19. Hu J, Runger G (2010) Time-based detection of changes to multivariate patterns. Ann Oper Res 174:67–81
20. Li J (2021) Nonparametric adaptive CUSUM chart for detecting arbitrary distributional changes. J Qual Technol 53:154–172
21. Li W, Zhang C, Tsung F, Mei Y (2021) Nonparametric monitoring of multivariate data via KNN learning. Int J Prod Res 59:6311–6326
22. Montgomery DC (2012) Introduction to statistical quality control. Wiley, New York
23. Opsomer J, Wang Y, Yang Y (2001) Nonparametric regression with correlated errors. Stat Sci 16:134–153
24. Page ES (1954) Continuous inspection scheme. Biometrika 41:100–115
25. Poonpun P, Jewell W (2014) Analysis of the cost per kilowatt hour to store electricity. IEEE Trans Energy Convers 23:529–534
26. Qiu P (2014) Introduction to statistical process control. Chapman Hall/CRC, Boca Raton, FL
27. Qiu P (2018) Some perspectives on nonparametric statistical process control. J Qual Technol 50:49–65
28. Qiu P, Xiang D (2014) Univariate dynamic screening system: an approach for identifying individuals with irregular longitudinal behavior. Technometrics 56:248–260
29. Qiu P, Xie X (2022) Transparent sequential learning for statistical process control of serially correlated data. Technometrics 64:487–501
30. Roberts SV (1959) Control chart tests based on geometric moving averages. Technometrics 1:239–250
31. Shewhart WA (1931) Economic control of quality of manufactured product. D. Van Nostrand Company, New York
32. Staffell I, Pfenninger S (2018) The increasing impact of weather on electricity supply and demand. Energy 145:65–78
33. Sukchotrat T, Kim SB, Tsung F (2009) One-class classification-based control charts for multivariate process monitoring. IIE Trans 42:107–120
34. Tuv E, Runger G (2003) Learning patterns through artificial contrasts with application to process control. Trans Inf Commun Technol 29:63–72
35. Xiang D, Qiu P, Pu X (2013) Nonparametric regression analysis of multivariate longitudinal data. Stat Sinica 23:769–789
36. Xie X, Qiu P (2023) Control chart for dynamic process monitoring with an application to air pollution surveillance. Ann Appl Stat 17:47–66
37. Xie X, Qiu P (2022) Machine learning control charts for monitoring serially correlated data. In: Tran KP (ed) Control charts and machine learning for anomaly detection in manufacturing. Springer, pp 131–147
38. Xue L, Qiu P (2021) A nonparametric CUSUM chart for monitoring multivariate serially correlated processes. J Qual Technol 53:396–409
39. Zhang C, Tsung F, Zou C (2015) A general framework for monitoring complex processes with both in-control and out-of-control information. Comput Ind Eng 85:157–168