



Employing machine learning techniques in monitoring autocorrelated profiles

Ali Yeganeh¹ · Arne Johannssen¹ · Nataliya Chukhrova² · Saddam Akber Abbasi^{3,4,5} · Farhad Pourpanah⁶

Received: 31 October 2022 / Accepted: 8 March 2023 / Published online: 29 April 2023
© The Author(s) 2023

Abstract

In profile monitoring, it is usually assumed that the observations between or within each profile are independent of each other. However, this assumption is often violated in manufacturing practice, and it is of utmost importance to carefully consider autocorrelation effects in the underlying models for profile monitoring. For this reason, various statistical control charts have been proposed to monitor profiles when between- or within-data is correlated in Phase II, in which the main aim is to develop control charts with quicker detection ability. As a novel approach, this study aims to employ machine learning techniques as control charts instead of statistical approaches in monitoring profiles with between-profile autocorrelations. Specifically, new input features based on conventional statistical control chart statistics and normalized estimated parameters are defined that are capable of adequately accounting for the between-autocorrelation effect of profiles. In addition, six machine learning techniques are extended and compared by means of Monte Carlo simulations. The simulation results indicate that machine learning techniques can obtain more accurate results compared with statistical control charts. Moreover, adaptive neuro-fuzzy inference systems outperform other machine learning techniques and the conventional statistical control charts.

Keywords Adaptive neuro-fuzzy inference system · Artificial neural network · Deep learning · Long short-term memory · Statistical process monitoring · Support vector regression

1 Introduction

Statistical process monitoring (SPM) is usually employed for industrial processes to omit assignable causes that deteriorate the product outcome. SPM is the major field for controlling process variations to eventuate lower costs in

waste, scrap, rework and claims, better quality, and more insights into the capability of the process. Seven main tools, entailing scatter diagrams, Pareto charts, control charts, histograms, cause-and-effect diagrams, check sheets and stratification, are utilized in SPM to implement inspection and monitoring procedures [1]. Among them,

✉ Arne Johannssen
arne.johannssen@uni-hamburg.de

Ali Yeganeh
yeganeh.ali1369@gmail.com

Nataliya Chukhrova
nataliya.chukhrova@hcu-hamburg.de

Saddam Akber Abbasi
sabbasi@qu.edu.qa

Farhad Pourpanah
farhad.086@gmail.com

² HafenCity University, 20457 Hamburg, Germany

³ Department of Mathematics, Statistics and Physics, Qatar University, Doha, Qatar

⁴ Department of Mathematics, Statistics and Physics, College of Arts and Sciences, Qatar University, 2713 Doha, Qatar

⁵ Statistical Consulting Unit, College of Arts and Sciences, Qatar University, 2713 Doha, Qatar

⁶ Department of Electrical and Computer Engineering, Center for Computer Vision and Deep Learning, University of Windsor, Windsor, Canada

¹ Faculty of Business Administration, University of Hamburg, 20146 Hamburg, Germany

control charts are the most successful and effective tools for quality control of manufacturing processes [2–4].

To employ a control chart for process monitoring, two phases entailing Phase I and II should be initially defined. In Phase I, it is tried to achieve proper estimations of the process parameters, whereas Phase II monitoring aims to find assignable causes in which the process situation changes from In-Control (IC) to Out-of-Control (OC) state [5–9]. Average Run Length (ARL) and Standard Deviation of Run Length (SDRL) are two common performance indicators in Phase II. The ARL is the average number of samples to be obtained by the predefined control chart before the chart triggers an OC signal. Thus, a control chart with larger (smaller) values of the ARL is to be preferred when the underlying process is in IC (OC) state [the more common notation is ARL_0 (ARL_1)] [10–12]. In addition, the SDRL is defined in a similar way as a secondary criterion in Phase II (i.e., $SDRL_0$ and $SDRL_1$).

There are two common approaches to monitor a manufacturing processes with the help of control charts namely *monitoring quality characteristics* and *profile monitoring* [13]. In this paper, we focus on profile monitoring. Here, the quality of a process or product is modelled via a functional relationship between a response (dependent) variable and one or more explanatory (independent) variable(s) [14]. The aim of profile monitoring is to check the stability of a predefined IC relationship (or profile) over time, and it is essential to reach a true OC signal as soon as possible when the IC model shifts to an unknown OC profile [15].

Different IC models can be employed due to the nature of the underlying problem, such as circular [16], linear [5, 17–22], logistic [23–25], nonlinear [26], nonparametric [27, 28], multichannel [29], polynomial [30] or quadratic [31]. Among them, linear profiles have received more attention in the literature [32, 33]. The majority of previous studies in linear profile monitoring is based on the independency assumption regarding within or between profiles in relation to the error terms. However, this assumption is often violated in manufacturing practice that is characterized by autocorrelated profiles and consequently, conventional approaches may lead to inaccurate outcomes for this type of profiles.

Autocorrelated profiles consist of within- and between-correlation models in the related literature [34–37]. In the first group, Soleimani, Noorossana and Amiri [38] developed four control charts including T^2 and three well-known Exponentially Weighted Moving Average (EWMA) charts

considering the first order Autoregressive (AR) model, i.e., AR(1). The results showed the superiority of EWMA-based approaches over T^2 . Autoregressive Moving Average (ARMA), Vector ARMA (VARMA) etc. are other more complex models that have been developed in this field [39–43]. Due to the higher potential for applications of the second group, this paper focuses on between-profile autocorrelation. To the best of the authors' knowledge, the pioneering work is Noorossana, Amiri and Soleimani [44], in which they developed T^2 , EWMA of residuals (EWMA/R) and transformed individual EWMA (EWMA-3) control charts for situations where autocorrelation effects exist between profiles. Similar to Soleimani, Noorossana and Amiri [38], they concluded that EWMA-based methods outperform T^2 . Wang and Lai [45] aggregated the individual EWMA statistics to a Multivariate EWMA (MEWMA) control chart for profiles with between-autocorrelation, and it has been shown that MEWMA outperforms T^2 . Khedmati and Niaki [46] considered both linear and polynomial profiles; they first utilized the U statistic for removing the effect of autocorrelation and then developed a T^2 -based control chart. The experimental results showed that this method performs better than conventional T^2 control charts, but comparisons with EWMA are missing. Koosha and Amiri [47] proposed a similar T^2 -based control chart for monitoring autocorrelated logistic profiles. Wang and Huang [48] modified the estimation procedure of the EWMA approach, and the simulation results demonstrated that this scheme has a faster detection ability than that of conventional EWMA.

From the literature, it can be inferred that the probability of occurring autocorrelations in practical applications is very high. Therefore, an early detection of OC conditions is more important than simulation results, as a delay in detection may result in the production of nonconformities and additional costs. However, the conventional control charts such as T^2 , EWMA/R and EWMA-3 are not able to perform well in line with this aim as their performance deteriorate in the occurrence of autocorrelation in comparison with simple situations; for example, it can be referred to the results of ARL_1 in Noorossana, Amiri and Soleimani [44] versus Kim et al. [20].

Hence, proposing a novel control chart with a tangible ability in reduction of the OC signaling time in autocorrelated profiles is crucial. To remedy this challenge, in recent years, several studies incorporated machine learning techniques in the SPM context in monitoring roundness [49], nonlinear [50–52], linear [53, 54] and logistic

[25, 55, 56] profiles. As a different approach, Chen et al. [57] employed a deep learning technique, called stacked denoising autoencoders, to monitor autocorrelated profiles. Specifically, this scheme extracts a number of features from the process using autoencoders, and then the extracted features are used to develop control charts based on T^2 and EWMA. In other words, the main task of their approach is to select proper features from the process, whereas the direct usage of machine learning techniques as a control chart would be more promising. As far as the authors know, there are no further articles where machine learning techniques are employed in monitoring autocorrelated profiles.

The aim of this paper is to develop a robust control chart based on machine learning techniques to alleviate the above-mentioned challenges, i.e., reducing the values of the ARL_1 and $SDRL_1$ for autocorrelated linear profiles that can result in early detection of OC situations in Phase II. To achieve this, three combinations of input features based on the effect of the mean of responses, the mean of errors, and T^2 statistic, each in the current and previous sample, are defined to feed into the machine learning techniques for monitoring profiles with between-autocorrelation of first order, i.e., AR(1). Since each machine learning technique performs differently in tackling various problems, six machine learning techniques ranging from shallow to deep structures including adaptive neuro-fuzzy inference system (ANFIS), artificial neural network (ANN) with Back-Propagation (BP) training, Convolutional Neural Network (CNN), long short-term memory (LSTM) network, Radial Basis Function (RBF) network and support vector regression (SVR), are employed to find the most appropriate one. To sum up, the main contributions of this paper are as follows:

- Improving the detection ability of Phase II control charts for monitoring linear autocorrelated profiles with the help of machine learning techniques,
- Defining different combinations of input features based on the effect of the mean of responses, the mean of errors, and T^2 statistic, each in the current and previous sample, for monitoring the between-autocorrelation effect of profiles,
- Evaluating the performance of the defined input features and finding the best combination using the proposed machine learning-based control chart, and
- Identification of the most appropriate machine learning technique under the most suitable input combination for this problem.

The rest of this article is organized as follows. In Sect. 2, definitions of autocorrelated linear profiles are discussed. Section 3 presents the framework of the proposed approach. Results of simulation studies regarding performance comparisons are given and discussed in Sect. 4. To show the effectiveness of our method, an illustrative example is given in Sect. 5. Finally, Sect. 6 gives some conclusions and suggests future research directions.

2 Preliminaries

In this section, first, the general relations of linear profiles with between-autocorrelation error terms are presented. Then, three common control charts, namely T^2 , EWMA/R and EWMA-3, are briefly introduced. Finally, details about the Ordinary Least Squares (OLS) estimation of the parameters are discussed. These basics are necessary against the backdrop that (1) our proposed method employs the T^2 statistic as input feature, and (2) these conventional charts are used for comparison purposes in our analyses.

2.1 The linear autocorrelated profile in Phase II

A common linear profile, which is the simplest but the most fundamental type of profiles [19], is defined as:

$$Y_{ij} = A_0 + A_1X_i + \varepsilon_{ij}, \varepsilon_{ij} \sim N(0, \sigma^2),$$

$$i = 1, 2, \dots, n; j = 1, 2, \dots, \quad (1)$$

where X_i represent the explanatory variable in a linear profile and the response variable Y_{ij} is the quality characteristic under study. The parameters of the above IC model (intercept A_0 , slope A_1 and error variance σ^2) are estimated from Phase I samples, and it is usually assumed that sample size n and independent variable X_i are fixed in each profile. When there is an AR(1) structure between the random error terms, (1) becomes:

$$Y_{ij} = A_0 + A_1X_i + \varepsilon_{ij}, \varepsilon_{ij} = \phi\varepsilon_{i(j-1)} + a_{ij}, a_{ij} \sim N(0, \sigma^2),$$

$$i = 1, 2, \dots, n; j = 1, 2, \dots, \quad (2)$$

where ϕ is a constant autocorrelation coefficient, which is assumed to be known in Phase II. To monitor the above IC profile, we briefly present three common approaches in the following subsections.

2.2 The T^2 control chart for monitoring autocorrelated profiles in Phase II

By some calculations, it can be easily shown that the estimated responses are obtained in the j^{th} generated sample over time as follows [44]:

$$\hat{y}_{ij} = \phi y_{i(j-1)} + (1 - \phi)(A_0 + A_1 X_i). \tag{3}$$

Thus, the empirical residuals can be written as:

$$e_{ij} = \hat{y}_{ij} - y_{ij}. \tag{4}$$

Noorossana, Amiri and Soleimani [44] used a modified form of the T^2 statistics proposed by Kang and Albin [14] in a simple linear profile:

$$t_j^2 = e_j' \sum_e^{-1} e_j, e_j = (e_{1j}, e_{2j}, \dots, e_{nj}), \tag{5}$$

where \sum_e is the symmetric $n \times n$ matrix $\sigma^2 I$. Since the chart statistic (t_j^2) is ensured to be larger than zero, it is compared with a predefined Upper Control Limit (UCL_T) to reach an OC signal. Note that the Lower Control Limit (LCL_T) is equal to 0.

2.3 The EWMA/R control chart for monitoring autocorrelated profiles in Phase II

In the EWMA/R control chart, two simultaneous statistics monitor the generated profiles. The first statistic is related to the mean of the residuals and is defined as follows [44, 53]:

$$z_j = \theta \bar{e}_j + (1 - \theta)z_{(j-1)}, \bar{e}_j = \frac{1}{n} \sum_{i=1}^n e_{ij}, \tag{6}$$

In (6), it holds $z_0 = 0$, and θ is the EWMA constant that usually has a value between 0.1 and 0.9 [20]. Following previous works [21, 53, 54, 58], θ is set to 0.2 in this paper.

The second statistic of the EWMA/R chart is the range of the empirical residuals defined by [37, 48]:

$$r_j = \max(e_{ij}) - \min(e_{ij}). \tag{7}$$

The EWMA/R declares the process as IC if both of the following conditions are met [14]:

$$-L\sigma \sqrt{\frac{\theta}{(2 - \theta)n}} < z_j < L\sigma \sqrt{\frac{\theta}{(2 - \theta)n}}, \tag{8}$$

$$\sigma(d_2 - Ld_3) < r_j < \sigma(d_2 + Ld_3)$$

In (8), the value of L is assigned to reach a predefined ARL_0 , while d_2 and d_3 are two constants that depend on the sample size (see Montgomery [1] for more details).

2.4 The EWMA-3 control chart for monitoring autocorrelated profiles in Phase II

To solve the problem of dependency between the estimators in linear profiles, Kim, Mahmoud and Woodall [20] suggested to have zero mean explanatory variables, where the least squares estimators of slope and intercept are independent random variables. By this coding (transformation), the EWMA-3 approach has the IC model

$$Y_{ij} = B_0 + B_1 X_{di} + \varepsilon_{ij}, \varepsilon_{ij} = \phi \varepsilon_{i(j-1)} + a_{ij}, \varepsilon_{ij} \sim N(0, \sigma^2), \tag{9}$$

$$i = 1, 2, \dots, n; \quad j = 1, 2, \dots,$$

where the coded explanatory variables ($X_{di} = X_i - \bar{X}$) lead to the transformed IC intercept $B_0 = A_0 + A_1 \bar{X}$, A_0 and A_1 are defined as in (1). Note that the transformed IC slope is $B_1 = A_1$ in this approach. The OLS estimation of the parameters ($\hat{B}_{0j}, \hat{B}_{1j}, \hat{\sigma}_{ij}^2$) generates three separate EWMA-based errors for intercept (e_{ij}), slope (e_{sj}) and standard deviation (e_{ij}), as follows:

$$e_{ij} = \hat{B}_{0j} + \phi \hat{B}_{0(j-1)} - (1 - \phi)B_0,$$

$$e_{sj} = \hat{B}_{1j} + \phi \hat{B}_{1(j-1)} - (1 - \phi)B_1,$$

$$e_{ij} = y_{ij} - \phi y_{i(j-1)} - (1 - \phi)B_0 + B_1 X_{di}, \tag{10}$$

$$MSE_j = \frac{1}{n} \sum_{i=1}^n e_{ij}^2.$$

Based on the OLS estimation, the Mean Square Error (MSE) of the j th profile is considered as the estimator of the error variance (for details see Kim et al. [20], Huwang et al. [59] and Yeganeh and Shadman [54]), thus, three chart statistics can be calculated as follows:

$$EWMA_{Ij} = \theta e_{Ij} + (1 - \theta)EWMA_{I(j-1)},$$

$$EWMA_{Sj} = \theta e_{Sj} + (1 - \theta)EWMA_{S(j-1)},$$

$$EWMA_{Ej} = \max[\theta(MSE_j - 1) + (1 - \theta)EWMA_{E(j-1)}, 0]. \tag{11}$$

The control limits of the three separate control charts are designed as:

$$\begin{aligned}
 UCL_I &= -LCL_I = L_I \sigma \sqrt{\frac{\theta}{n(2-\theta)}} \\
 UCL_S &= -LCL_S = L_S \sigma \sqrt{\frac{\theta}{n(2-\theta)}} \\
 UCL_E &= L_E \sqrt{\frac{\theta \text{Var}(\text{MSE}_j)}{(2-\theta)}}
 \end{aligned}
 \tag{12}$$

It is worth noting that $LCL_E = 0$, and some suggestions regarding $\text{Var}(\text{MSE}_j)$ can be found in Kim, Mahmoud and Woodall [20], Noorossana et al. [44] and Hosseinifard et al. [53]. EWMA-3 triggers an OC signal when at least one of the statistics exceed the control limits. The proposed constants entailing L_I , L_S and L_E are usually adjusted to reach a desired value of ARL_0 in such a way that each of the separate charts achieves an identical individual ARL_0 .

2.5 OLS estimation of the parameters

Since the parameters estimated via OLS are employed as inputs of the machine learning techniques, some details on the OLS estimators are given below [20, 21, 53]. While the intercept of the original and the transformed model is estimated by $\hat{A}_{0j} = \bar{Y}_j - \hat{A}_{1j}\bar{X}$ and $\hat{B}_{0j} = \bar{Y}_j$, respectively, the slope parameter in both models is estimated via $\hat{A}_{1j} = \hat{B}_{1j} = \frac{S_{XYj}}{S_{XX}}$, where $S_{XYj} = \sum_{i=1}^n Y_{ij}(X_i - \bar{X})$ and $S_{XY} = \sum_{i=1}^n (X_i - \bar{X})^2$ (with $\bar{Y}_j = \frac{1}{n} \sum_{i=1}^n Y_{ij}$ and $\bar{X}_j = \frac{1}{n} \sum_{i=1}^n X_{ij}$). Note that the definition of MSE_j based on (10) is equivalent to the estimator of the error variance.

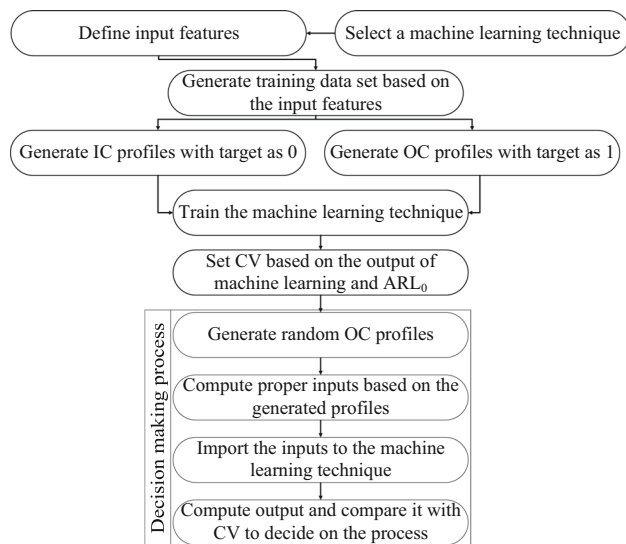


Fig. 1 The general step-by-step flowchart of the proposed method

3 The proposed control chart for monitoring autocorrelated profiles

The basic idea in this paper is to use machine learning techniques instead of statistical control charts for monitoring profiles with between-autocorrelation. For this purpose, several features are extracted from the process to embed into the machine learning techniques. Using training patterns and the obtained control limits help to improve decision-making about the process.

To employ a machine learning technique as a control chart, four main steps are defined in the following. In the *first step*, the structure of the input features and outputs of a machine learning technique are determined, and then, a training data set based on the input features and outputs is generated by simulating IC and OC profiles in the *second step*. The *third step* uses the generated data set to train a machine learning technique, and finally, by the definition of a control limit, the machine learning technique provides information about the process condition in the *fourth step*. Figure 1 shows a step-by-step flowchart regarding the proposed method in monitoring autocorrelated profiles. Moreover, details of these steps are presented in the following subsections.

3.1 Defining the structure of input features and outputs

Extracting proper input features is a key step in the implementation of machine learning techniques [60]. In the literature, various strategies have been employed for extracting features. For example, Chen et al. [57] and Sergin and Yan [51] used autoencoders to obtain Phase I information, which is not compatible for this study. Hosseinifard et al. [53] and Yeganeh and Shadman [54] suggested to take the OLS estimations of the linear profile parameters as input features. This input structure performs well in profiles without autocorrelation, but it is not able to provide reliable results when between-autocorrelation of the profiles is present since the effect of autocorrelation is not captured by the distribution of the OLS estimators. So, in addition to these inputs $(\hat{A}_{0j}, \hat{A}_{1j}, \hat{\sigma}_j)$, further appropriate input features are proposed in the following, which are suitable to account for between-autocorrelation effects of first order (i.e., AR(1) autocorrelation). In particular, the proposed input structure addresses the effect of the mean of responses, the mean of errors, and T^2 statistic each in the current sample (j) and the previous sample ($j-1$). It should be mentioned that one of the main benefits of machine learning-based algorithms is their independency to the basic assumptions about the process. So, when replacing proper estimations related to other autocorrelation models

such as MA, ARMA and ARIMA, the proposed method can be easily applied for other process conditions.

In machine learning, there are several approaches for normalizing and scaling of the inputs, one of which involves the parameter distribution to be used. For instance, Yeganeh and Shadman [54] scaled the parameters of simple linear profiles, i.e., intercept, slope and standard deviation, with normal and chi square distribution (the relations are not reported for brevity but the interested reader is referred to Eqs. (4) to (9) in Yeganeh and Shadman [54]). Since the autocorrelation was not considered in Yeganeh and Shadman [54] for OLS estimators, we suggest to utilise the between-autocorrelation effect of the first order by means of deviations between current (previous) OLS estimators and their respective IC values, i.e., $\widehat{A}_{0j} - A_0, \widehat{A}_{1j} - A_1, \widehat{\sigma}_j - \sigma_0 (\widehat{A}_{0(j-1)} - A_0, \widehat{A}_{1(j-1)} - A_1, \widehat{\sigma}_{(j-1)} - \sigma_0)$. Using this approach, the means of current and previous responses $(\bar{y}_j, \bar{y}_{(j-1)})$ are automatically incorporated in the input structure as they are functions of the OLS estimates $(\widehat{A}_{0j} = \bar{y}_j - \widehat{A}_{1j}\bar{x}, \widehat{A}_{0(j-1)} = \bar{y}_{(j-1)} - \widehat{A}_{1(j-1)}\bar{x})$.

By considering the above inputs, numerous input combinations can be defined in the proposed method. Having investigated several input features, the following three input combinations I, II and III with 8, 4 and 10 inputs, respectively, are employed for each machine learning technique based on their ability to: (1) adequately address the specific effects of between-autocorrelation of type AR(1), and (2) incorporate the deviations between OLS estimates and their respective IC values for efficient Phase II monitoring:

- (I) $\widehat{A}_{0(j-1)} - A_0, \widehat{A}_{1(j-1)} - A_1, \widehat{\sigma}_{(j-1)} - \sigma_0, \widehat{A}_{0j} - A_0, \widehat{A}_{1j} - A_1, \widehat{\sigma}_j - \sigma_0, \bar{e}_j, \bar{e}_{(j-1)}$.
- (II) $\widehat{\sigma}_{(j-1)} - \sigma_0, \widehat{\sigma}_j - \sigma_0, t^2_{(j-1)}, t^2_j$.
- (III) $\widehat{A}_{0(j-1)} - A_0, \widehat{A}_{1(j-1)} - A_1, \widehat{\sigma}_{(j-1)} - \sigma_0, \widehat{A}_{0j} - A_0, \widehat{A}_{1j} - A_1, \widehat{\sigma}_j - \sigma_0, \bar{e}_j, \bar{e}_{(j-1)}, t^2_{(j-1)}, t^2_j$.

These input combinations are motivated by their ability to consider effects regarding autocorrelation in various ways. While input combination I addresses the raw AR(1) structure of the underlying model, input combination II aims to isolate the effects of the current and previous T^2 statistics extended by an additional consideration of the current and previous error variances. Finally, input combination III is the union of input combinations I and II, and

therefore, combines both main effects. In these three input combinations, the following notations are utilized:

- Estimated parameters via OLS in the previous sample $(\widehat{A}_{0(j-1)}, \widehat{A}_{1(j-1)}, \widehat{\sigma}_{(j-1)})$.
- Estimated parameters via OLS in the current sample $(\widehat{A}_{0j}, \widehat{A}_{1j}, \widehat{\sigma}_j)$.
- Mean of error terms in the previous sample $(\bar{e}_{(j-1)})$.
- Mean of error terms in the current sample (\bar{e}_j) .
- T^2 statistic in the previous sample $(t^2_{(j-1)})$.
- T^2 statistic in the current sample (t^2_j) .

Note that the T^2 statistic is added as an input feature and no further statistics from other conventional control charts in order to: (1) increase the worse performance of the T^2 control chart, and (2) avoid overparameterization and complexity with regard to further common competitors.

3.2 Generation of the training data set

To construct the training data set, the IC and OC profiles are generated by means of simulations. From the simulated profiles, the inputs are constructed based on the predefined three input structures I, II and III in Sect. 3.1. For example, the inputs of the j th generated profile consist of 10 features in input combination III as $\widehat{A}_{0(j-1)} - A_0, \widehat{A}_{1(j-1)} - A_1, \widehat{\sigma}_{(j-1)} - \sigma_0, \widehat{A}_{0j} - A_0, \widehat{A}_{1j} - A_1, \widehat{\sigma}_j - \sigma_0, \bar{e}_{(j-1)}, \bar{e}_j, t^2_{(j-1)}$ and t^2_j . Considering the suggestions by Hosseini-fard, Abdollahian and Zeephongsekul [53], equal numbers of IC and OC profiles are generated in a way that the target values of IC and OC profiles are set to 0 and 1, respectively.

Hence, we consider the size of the training data set (number of rows) as $6G$. First, $3G$ IC profiles are generated and their input features are recorded with a target value equal to 0. Then, $3G$ OC profiles (G profiles with shift in intercept, G profiles with shift in slope, and G profiles with shift in standard deviation) are obtained in the same way with a target value equal to 1. Finally, the training data set has $6G$ rows and 9 ($8 + 1$), 5 ($4 + 1$) and 11 ($10 + 1$) columns for input combinations I, II and III, respectively (note that the last column represents the target values). For better understanding, pseudo code 1 illustrates the process of data set generation for input combination III (an analogous procedure also applies to input combinations I and II).

Pseudo code 1. Generation of the training data set for input combination III

```

Define IC model and assign  $G$ ;
Data set Train = Create a temporary vector with size 11;
% IC data generation
Target = 0;  $j = 1$ ;  $\bar{e}_0 = 0$ ;  $\bar{y}_0 = \frac{\sum_{i=1}^n A_0 + A_1 x_i}{n}$ ;
  while ( $j < 3G$ )
    Generate random IC profile;
    Compute [ $\hat{A}_{0(j-1)} - A_0, \hat{A}_{1(j-1)} - A_1, \hat{\sigma}_{(j-1)} - \sigma_0, \hat{A}_{0j} - A_0, \hat{A}_{1j} - A_1, \hat{\sigma}_j - \sigma_0, \bar{e}_{(j-1)}, \bar{e}_j, t^2_{(j-1)}, t^2_j$ ];
    Data set Train = [Data set Train; [ $[\hat{A}_{0(j-1)} - A_0, \hat{A}_{1(j-1)} - A_1, \hat{\sigma}_{(j-1)} - \sigma_0, \hat{A}_{0j} - A_0, \hat{A}_{1j} - A_1, \hat{\sigma}_j - \sigma_0, \bar{e}_{(j-1)}, \bar{e}_j, t^2_{(j-1)},$ 
 $t^2_j]$ ;Target]];  $j = j + 1$ ;
  end while;
% OC data generation with shift in intercept
Target = 1;  $j = 1$ ;  $\bar{e}_0 = 0$ ;  $\bar{y}_0 = \frac{\sum_{i=1}^n A_0 + A_1 x_i}{n}$ ;
  while ( $j < G$ )
    Generate random OC profile with a shift in intercept;
    Compute [ $\hat{A}_{0(j-1)} - A_0, \hat{A}_{1(j-1)} - A_1, \hat{\sigma}_{(j-1)} - \sigma_0, \hat{A}_{0j} - A_0, \hat{A}_{1j} - A_1, \hat{\sigma}_j - \sigma_0, \bar{e}_{(j-1)}, \bar{e}_j, t^2_{(j-1)}, t^2_j$ ];
    Data set Train = [Data set Train; [ $[\hat{A}_{0(j-1)} - A_0, \hat{A}_{1(j-1)} - A_1, \hat{\sigma}_{(j-1)} - \sigma_0, \hat{A}_{0j} - A_0, \hat{A}_{1j} - A_1, \hat{\sigma}_j - \sigma_0, \bar{e}_{(j-1)}, \bar{e}_j, t^2_{(j-1)},$ 
 $t^2_j]$ ;Target]];  $j = j + 1$ ;
  end while;
% OC data generation with shift in slope
Target = 1;  $j = 1$ ;  $\bar{e}_0 = 0$ ;  $\bar{y}_0 = \frac{\sum_{i=1}^n A_0 + A_1 x_i}{n}$ ;
  while ( $j < G$ )
    Generate random OC profile with a shift in slope;
    Compute [ $\hat{A}_{0(j-1)} - A_0, \hat{A}_{1(j-1)} - A_1, \hat{\sigma}_{(j-1)} - \sigma_0, \hat{A}_{0j} - A_0, \hat{A}_{1j} - A_1, \hat{\sigma}_j - \sigma_0, \bar{e}_{(j-1)}, \bar{e}_j, t^2_{(j-1)}, t^2_j$ ];
    Data set Train = [Data set Train; [ $[\hat{A}_{0(j-1)} - A_0, \hat{A}_{1(j-1)} - A_1, \hat{\sigma}_{(j-1)} - \sigma_0, \hat{A}_{0j} - A_0, \hat{A}_{1j} - A_1, \hat{\sigma}_j - \sigma_0, \bar{e}_{(j-1)}, \bar{e}_j, t^2_{(j-1)},$ 
 $t^2_j]$ ;Target]];  $j = j + 1$ ;
  end while;
% OC data generation with shift in standard deviation
Target = 1;  $j = 1$ ;  $\bar{e}_0 = 0$ ;  $\bar{y}_0 = \frac{\sum_{i=1}^n A_0 + A_1 x_i}{n}$ ;
  while ( $j < G$ )
    Generate random OC profile with a shift in standard deviation;
    Compute [ $\hat{A}_{0(j-1)} - A_0, \hat{A}_{1(j-1)} - A_1, \hat{\sigma}_{(j-1)} - \sigma_0, \hat{A}_{0j} - A_0, \hat{A}_{1j} - A_1, \hat{\sigma}_j - \sigma_0, \bar{e}_{(j-1)}, \bar{e}_j, t^2_{(j-1)}, t^2_j$ ];
    Data set Train = [Data set Train; [ $[\hat{A}_{0(j-1)} - A_0, \hat{A}_{1(j-1)} - A_1, \hat{\sigma}_{(j-1)} - \sigma_0, \hat{A}_{0j} - A_0, \hat{A}_{1j} - A_1, \hat{\sigma}_j - \sigma_0, \bar{e}_{(j-1)}, \bar{e}_j, t^2_{(j-1)},$ 
 $t^2_j]$ ;Target]];  $j = j + 1$ ;
  end while;

```

3.3 Training a machine learning technique

After obtaining a training data set, a machine learning technique can be trained on its basis. In this paper, six common machine learning techniques, i.e., ANFIS, ANN,

RBF, SVR, CNN and LSTM, with the ability of generating continuous outputs are investigated. For better understanding, we provide a brief description about the parameters and adjustments of each method in MATLAB software.

- The ANFIS approach utilizes fuzzy *IF-THEN* rules to train the parameters with some basic algorithms such as subtractive clustering and grid partitioning, in which the idea of rule generation is different. The ‘*genfis*’ function, which is a well-known single-output Sugeno fuzzy inference system, is used to obtain a grid partition for the training procedure.
- The ANN structure, especially a Multi-Layer Perceptron (MLP), with gradient-based optimization is employed here. An important issue in ANNs is related to the adjustment of the number of hidden layers and the neurons. The function ‘*feedforwardnet*’ utilizes a fully connected network architecture using the BP Levenberg–Marquardt training algorithm (‘*trainlm*’ option). In this study, a single hidden layer with 10 neurons is suggested for training.
- RBF considers different training approaches based on the idea of clustering. It has only one hidden layer in a way that the neurons with a distinct spread (radius) are added to its structure until the pre-specified error or maximum number of neurons is obtained. Because the training procedure is completed by the aim of training error reduction, the probability of overfitting would be generally high in this approach. The function ‘*newrb*’ with spread (radius) 1, error rate 0.05 and maximum neuron size 100 is selected for training purposes.
- As additional machine learning techniques, SVM and SVR obtain the parameters based on generating a hyperplane in the problem space in order to minimize the gaps between the predicted and obtained values considering a kernel function for mapping the inputs to the problem space. As we aim to reach a continuous output (regression problem) in this study, the SVR function ‘*fitrsvm*’ with the gaussian kernel function is used for training (more details about the classification and regression nature of machine learning-based control charts can be found in Yeganeh and Shadman [54]). The parameter epsilon, which determines the distance between the real and the estimated planes in the space, is an important parameter of the SVR technique. As the common range of epsilon is [0.3–0.5] [61], the value 0.3 is selected in this paper.
- The deep learning technique CNN is utilized to investigate its detection ability in the SPM field. Generally, a

CNN layer moves some filters along the input vertically and horizontally and computes the dot product of the weights and the input, and then adds a bias term to reach some novel features from the process. CNNs have several parameters such as padding and filter size. As the inputs of this paper are in vector form, the layers are created with ‘*convolution1dLayer*’ function with filter size 5.

- As a further deep learning technique, LSTM is trained to evaluate its performance. Due to the consideration of time dependencies with the aim of time units, LSTM can identify the time series related patterns in an effective way. The most important parameter of LSTM is the number of hidden neurons in each unit that is nearly like the number of hidden layers in common ANNs. Two LSTM layers are defined by the ‘*lstmLayer*’ function with 40 hidden neurons in a way that the Adam optimizer is utilised to obtain the best weights.

3.4 Decision on the process condition

Considering the definition of target values, Hosseinifard et al. [53] set the LCL of their proposed method to 0 and denoted the UCL as Cutting Value (CV). The CV is adjusted by simulations to reach the desired value of ARL_0 . After adjustment of the CV, the output of the considered machine learning technique, e.g., ANN, is compared with the CV to make a decision on the process [43]. If the output of the ANN in the j th sample (O_j) is larger than the CV, this indicates an OC condition (see Fig. 1 in Hosseinifard et al. [53] for more details).

By employing this approach in our proposed framework, we can identify the process condition when comparing the output O_j of a machine learning technique with the respective CV. For better understanding, pseudo code 2 illustrates the procedure of reaching an OC signal in one iteration of simulations when input combination III is used. To compute the ARL and SDRL by means of Monte Carlo simulations, this procedure is iterated 10,000 times.

Pseudo code 2. Signaling procedure in one iteration for the proposed method with input combination III

Consider a specific shift in the profile parameters;
 Assign CV based on ARL_0 ;
 $O_0 = 0; j = 0; \bar{e}_0 = 0$;
while ($O_j < CV$)
 $j = j + 1$;
 Generate random profile with pre-specific shift;
 Compute $[\hat{A}_{0(j-1)} - A_0, \hat{A}_{1(j-1)} - A_1, \hat{\sigma}_{(j-1)} - \sigma_0, \hat{A}_{0j} - A_0, \hat{A}_{1j} - A_1, \hat{\sigma}_j - \sigma_0, \bar{e}_{(j-1)}, \bar{e}_j, t_{(j-1)}^2, t_j^2]$;
 Import the input to the machine learning technique and compute O_j ;
end while;
 Trigger an OC signal;

4 Simulation study

To show the effectiveness of our proposed method, a comprehensive simulation study is conducted in this section. To compare the six machine learning techniques ANFIS, ANNBP, CNN, LSTM, RBF and SVR, the CV for each method is set to reach ARL_0 equal to 200, as it is the most common value in profile monitoring. The next aim is to find the best input among the three combinations I, II and III. As such, three input combinations under different parameter settings are used as inputs to the machine learning techniques. Due to the page limit, we only present the results of three input combinations for the shift in intercept in Table 1 under $\phi = 0.1$ and other results can be given to the interested readers upon request. As can be seen, nearly all machine learning techniques produced the best (i.e., lowest) values in terms of ARL_1 for input combination III. This is due to the fact that input combination III is able to combine both main effects of input combinations I and II, namely:

- (1) appropriately addressing the raw AR(1) structure of the underlying model, and.
- (2) the effects of the current and previous T^2 statistics as well as of current and previous error variances.

In other words, only considering (2), i.e., using input combination II, is not enough to reach proper results. On the other hand, concentrating on (1), i.e., using input combination I, enables to obtain better performance on average than with (2). The combination of (1) and (2) via input combination III clearly strengthens the effect of (1) and leads to superior results.

Similar results are obtained for the rest of parameter settings; thus, we only present the results of input combi-

nation III for the rest of the experiments. In addition, due to the first-priority importance of detecting small shifts in the underlying process, the focus is mainly on smaller shifts in the simulation studies. As for larger shifts, there are generally the same patterns as for smaller shifts.

For comparisons of single shifts in intercept, slope and standard deviation, the IC model is taken from Noorossana et al. [44] and Wang and Huang [48], where $A_0 = 3$, $A_1 = 2$ and $\sigma^2 = 1$. In addition, the explanatory variables have the values 2, 4, 6, 8 ($n = 4$), and 0.1, 0.5, 0.9 are considered as fixed values of ϕ . For comparisons of simultaneous shifts, the IC model is extracted from Wang and Lai [45].

In Sect. 4.1, the performance of various machine learning techniques is compared and the method with the best performance is selected. In Sect. 4.2, the selected machine learning technique and conventional statistical control charts in Phase II profile monitoring are compared. Finally, Sect. 4.3 reports the performance of the best approach for simultaneous shifts in profile parameters.

4.1 Comparing different machine learning techniques based on input combination III

In this subsection, three individual shifts are considered for each parameter to compare the performance of each technique in a way that the shifted parameters are $A_0 + \lambda\sigma$, $A_1 + \eta\sigma$ and $\gamma\sigma$. The values of ARL_1 associated with different machine learning techniques are given in Table 2. To have a fair comparison, the values of the $SDRL_1$ are additionally provided in Table 3 for each of the considered shifts. The bold values represent the approach with the best performance.

Table 1 ARL₁ comparisons considering all shifts in the intercept (λ) for the input combinations I, II and III under $\phi = 0.1$

Method	λ (input combination I)									
	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
ANFIS	48.681	7.936	2.068	2.000	2.002	2.002	2.000	2.000	2.000	2.000
ANNBP	260.462	114.170	21.057	4.174	4.020	4.000	3.998	4.000	3.998	3.998
CNN	222.460	257.410	339.520	450.670	589.050	-	-	-	-	-
LSTM	128.089	87.461	55.138	38.476	30.272	19.685	16.281	10.951	8.421	7.682
RBF	180.161	138.318	102.489	72.724	54.446	37.491	25.035	15.881	10.098	6.368
SVR	96.598	24.097	2.521	2.032	2.001	1.999	1.987	1.977	1.950	1.913
Method	λ (input combination II)									
	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
ANFIS	138.288	68.732	31.028	14.752	7.481	4.449	2.805	2.088	1.727	1.383
ANNBP	178.636	83.813	36.501	17.717	9.196	5.688	4.242	3.141	2.844	2.718
CNN	233.525	177.409	119.579	78.835	49.223	29.841	18.586	8.628	4.573	2.509
LSTM	146.448	93.650	49.129	30.746	17.140	11.107	6.209	4.501	2.339	1.757
RBF	215.453	199.574	207.502	237.539	317.386	451.197	778.421	1878.443	4860.056	15,991.005
SVR	161.838	97.662	53.292	34.853	23.866	21.463	22.473	30.146	46.205	96.342
Method	λ (input combination III)									
	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
ANFIS	43.486	3.096	2.013	2.000	2.000	2.000	2.000	2.000	2.000	2.000
ANNBP	91.760	12.226	4.066	3.993	3.982	3.969	3.942	3.881	3.824	3.695
CNN	196.730	196.318	141.750	74.888	36.169	17.267	9.039	5.033	3.192	2.093
LSTM	88.250	43.784	23.097	11.525	6.487	4.156	2.717	1.998	1.536	1.286
RBF	177.846	98.682	46.009	23.104	11.437	6.128	3.835	2.579	1.904	1.537
SVR	155.589	70.235	21.689	2.999	2.051	2.010	2.011	2.030	2.075	2.290

Table 2 ARL₁ comparisons considering small shifts in intercept (λ), slope (η) and standard deviation (γ) for the input combination III and various values of ϕ

Method	ϕ																													
	0.1					0.5					0.9																			
	λ										η										γ									
	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.25	0.05	0.075	0.1	0.125	0.025	0.05	0.075	0.1	0.125	0.025	0.05	0.075	0.1	0.125
ANFIS	43.486	3.096	2.013	2.000	2.000	86.916	22.219	3.145	2.010	2.002	144.677	89.822	47.460	21.007	5.794	91.760	12.226	4.066	3.993	3.982	203.930	52.482	7.588	4.111	4.000	194.451	92.170	30.952	5.293	3.995
ANNBP	196.730	43.784	23.097	11.525	6.487	120.918	74.857	48.816	34.408	23.429	216.380	217.183	212.464	222.914	211.987	196.730	196.318	141.750	74.888	36.169	225.980	273.747	218.808	270.869	262.101	216.380	217.183	212.464	222.914	211.987
CNN	88.250	43.784	23.097	11.525	6.487	120.918	74.857	48.816	34.408	23.429	193.426	103.727	79.522	53.711	39.301	88.250	43.784	23.097	11.525	6.487	120.918	74.857	48.816	34.408	23.429	193.426	103.727	79.522	53.711	39.301
LSTM	177.846	98.682	46.009	23.104	11.437	162.878	139.245	115.857	53.551	37.980	188.115	119.235	56.713	8.788	2.395	177.846	98.682	46.009	23.104	11.437	162.878	139.245	115.857	53.551	37.980	188.115	119.235	56.713	8.788	2.395
RBF	155.589	70.235	21.689	2.999	2.051	115.096	40.703	14.052	2.365	2.040	164.683	108.124	68.474	33.823	8.631	155.589	70.235	21.689	2.999	2.051	115.096	40.703	14.052	2.365	2.040	164.683	108.124	68.474	33.823	8.631
SVR																														
Method	η										γ																			
	0.025	0.05	0.075	0.1	0.125	0.025	0.05	0.075	0.1	0.125	0.025	0.05	0.075	0.1	0.125	0.025	0.05	0.075	0.1	0.125	0.025	0.05	0.075	0.1	0.125	0.025	0.05	0.075	0.1	0.125
ANFIS	88.629	19.977	3.890	2.170	2.006	111.182	55.031	23.630	7.894	2.554	175.783	143.763	109.363	74.653	50.791	163.597	42.886	12.072	4.445	4.014	291.387	182.663	83.709	22.442	8.482	248.431	158.491	87.173	50.070	26.125
ANNBP	195.581	200.452	199.732	161.668	113.750	189.323	224.919	231.323	246.545	267.192	214.247	219.537	205.613	225.729	227.702	195.581	200.452	199.732	161.668	113.750	189.323	224.919	231.323	246.545	267.192	214.247	219.537	205.613	225.729	227.702
CNN	99.174	53.083	30.527	17.920	11.012	126.571	86.143	60.878	48.327	29.694	135.823	120.072	89.606	82.092	62.414	99.174	53.083	30.527	17.920	11.012	126.571	86.143	60.878	48.327	29.694	135.823	120.072	89.606	82.092	62.414
LSTM	202.962	146.796	98.082	62.851	36.595	167.429	203.143	200.265	144.918	143.245	193.292	154.166	121.484	96.986	48.968	202.962	146.796	98.082	62.851	36.595	167.429	203.143	200.265	144.918	143.245	193.292	154.166	121.484	96.986	48.968
RBF	162.401	93.672	52.546	24.767	11.594	170.755	102.016	52.892	29.386	12.044	160.631	137.422	97.349	80.197	57.297	162.401	93.672	52.546	24.767	11.594	170.755	102.016	52.892	29.386	12.044	160.631	137.422	97.349	80.197	57.297
SVR																														
Method	γ										γ																			
	1.2	1.4	1.6	1.8	2	1.2	1.4	1.6	1.8	2	1.2	1.4	1.6	1.8	2	1.2	1.4	1.6	1.8	2	1.2	1.4	1.6	1.8	2	1.2	1.4	1.6	1.8	2
ANFIS	49.657	24.790	16.089	12.685	10.510	35.421	12.779	6.909	4.544	3.510	64.695	25.511	12.803	7.884	5.532	49.657	24.790	16.089	12.685	10.510	35.421	12.779	6.909	4.544	3.510	64.695	25.511	12.803	7.884	5.532
ANNBP	82.403	30.891	16.389	10.916	8.077	78.915	27.910	16.744	10.462	8.281	140.020	66.717	35.559	22.451	16.802	82.403	30.891	16.389	10.916	8.077	78.915	27.910	16.744	10.462	8.281	140.020	66.717	35.559	22.451	16.802
CNN	95.838	22.484	8.502	4.866	3.162	164.192	31.384	11.677	5.283	3.010	51.960	20.969	10.511	6.432	4.525	95.838	22.484	8.502	4.866	3.162	164.192	31.384	11.677	5.283	3.010	51.960	20.969	10.511	6.432	4.525
LSTM	56.844	28.600	18.280	13.776	11.124	75.571	30.347	29.939	19.571	21.122	81.518	62.422	62.602	58.711	63.530	56.844	28.600	18.280	13.776	11.124	75.571	30.347	29.939	19.571	21.122	81.518	62.422	62.602	58.711	63.530
RBF	27.041	8.772	4.580	2.966	2.248	22.020	10.816	3.735	2.714	2.184	40.287	15.934	9.748	8.203	8.842	27.041	8.772	4.580	2.966	2.248	22.020	10.816	3.735	2.714	2.184	40.287	15.934	9.748	8.203	8.842
SVR	44.255	18.177	11.632	10.448	10.952	47.245	23.683	18.357	16.566	19.410	55.655	39.398	40.116	42.161	62.349	44.255	18.177	11.632	10.448	10.952	47.245	23.683	18.357	16.566	19.410	55.655	39.398	40.116	42.161	62.349

Table 3 SDR_{L1} comparisons considering small shifts in intercept (λ), slope (η) and standard deviation (γ) for the input combination III and various values of ϕ

Method	ϕ														
	0.1					0.9									
	λ														
	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1					
ANFIS	41.511	1.593	0.116	0.000	0.000	85.159	20.693	1.519	0.099	0.047	140.243	85.831	44.917	18.773	4.235
ANNBP	87.715	9.829	0.433	0.134	0.189	191.145	45.771	4.809	0.539	0.000	167.322	84.394	26.709	4.140	0.174
CNN	223.457	216.683	158.727	81.585	38.033	301.267	283.730	249.934	294.253	339.858	214.801	208.212	211.106	215.770	205.786
LSTM	90.034	42.669	23.332	10.744	5.789	105.531	71.531	42.589	40.874	26.227	143.447	89.651	78.859	52.650	42.535
RBF	172.089	96.945	45.931	23.110	10.818	172.131	134.313	103.372	45.601	42.635	173.271	107.553	48.343	7.315	0.787
SVR	158.883	72.569	20.742	1.716	0.296	114.596	39.834	12.570	0.937	0.308	162.555	105.062	63.735	35.477	9.428

Method	η														
	0.025	0.05	0.075	0.1	0.125	0.025	0.05	0.075	0.1	0.125	0.025	0.05	0.075	0.1	0.125
ANFIS	85.578	18.215	2.431	0.509	0.074	109.446	53.362	22.096	6.311	0.948	171.757	140.779	104.133	70.735	46.725
ANNBP	164.013	40.871	9.803	1.155	0.290	270.335	206.368	77.230	20.346	5.885	221.148	153.464	79.820	43.473	23.121
CNN	228.867	233.531	220.039	182.966	126.248	218.986	217.980	251.446	231.830	245.173	208.397	213.699	190.458	215.281	228.720
LSTM	100.931	53.275	29.572	17.382	10.540	130.737	71.420	59.156	47.145	28.092	133.759	113.919	99.049	84.907	60.108
RBF	196.794	147.626	96.612	62.440	34.897	182.283	233.070	236.216	129.882	125.021	182.574	143.612	117.028	94.293	41.753
SVR	164.184	93.626	51.153	23.162	10.523	173.153	93.497	53.139	28.743	11.554	138.379	140.612	97.978	79.200	66.815

Method	γ														
	1.2	1.4	1.6	1.8	2	1.2	1.4	1.6	1.8	2	1.2	1.4	1.6	1.8	2
ANFIS	49.146	22.640	15.057	11.837	10.032	33.972	11.499	5.688	3.474	2.471	61.485	23.262	10.958	6.424	4.150
ANNBP	83.257	27.136	13.885	7.851	5.554	73.825	28.245	13.832	7.111	6.017	132.698	62.626	35.204	19.247	15.173
CNN	103.764	24.601	8.314	4.394	2.663	170.356	31.692	10.560	4.734	2.787	47.777	17.547	8.244	4.529	2.950
LSTM	54.309	26.961	17.488	13.406	10.901	65.382	36.286	31.122	19.931	23.268	81.664	60.809	64.828	63.546	59.199
RBF	25.456	7.800	3.737	2.163	1.491	18.639	11.243	2.921	1.826	1.219	34.356	12.626	8.112	6.717	8.054
SVR	42.762	17.152	10.568	9.066	10.470	44.726	25.081	16.282	15.034	18.415	59.005	38.480	40.647	43.803	71.503

Table 4 ARL₁ comparisons considering a wider range of shifts in intercept (λ), slope (η) and standard deviation (γ) for the input combination III and $\phi = 0.9$

Method	λ									
	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
ANFIS	144.677	89.822	47.460	21.007	5.794	2.173	2.090	2.086	2.100	2.124
ANNBP	194.451	92.170	30.952	5.293	3.995	3.940	3.910	3.799	3.649	3.393
CNN	216.380	217.183	212.464	222.914	211.987	211.104	207.297	189.189	170.760	160.546
LSTM	139.426	103.727	79.522	53.711	39.301	29.880	18.956	12.976	6.747	3.904
RBF	188.115	119.235	56.713	8.788	2.395	2.152	2.264	2.315	2.433	2.519
SVR	164.683	108.124	68.474	33.823	8.631	2.285	2.173	2.124	2.281	2.313
Method	η									
	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2	0.225	0.25
ANFIS	175.783	143.763	109.363	74.653	50.791	31.012	15.633	5.825	2.567	2.176
ANNBP	248.431	158.491	87.173	50.070	26.125	7.654	4.190	3.955	3.870	3.815
CNN	214.247	219.537	205.613	225.729	227.702	222.676	212.587	210.799	216.418	202.293
LSTM	135.823	120.072	89.606	82.092	62.414	49.494	44.859	33.867	29.811	15.373
RBF	193.292	154.166	121.484	96.986	48.968	17.264	4.476	2.476	2.281	2.241
SVR	160.631	137.422	97.349	80.197	57.297	35.293	23.586	6.671	2.659	2.020
Method	γ									
	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3
ANFIS	64.695	25.511	12.803	7.884	5.532	4.240	3.496	3.044	2.744	2.516
ANNBP	140.020	66.717	35.559	22.451	16.802	11.323	9.338	7.308	6.486	5.544
CNN	51.960	20.969	10.511	6.432	4.525	3.669	3.218	2.713	2.416	2.223
LSTM	81.518	62.422	62.602	58.711	63.530	69.185	89.924	94.494	100.394	98.791
RBF	40.287	15.934	9.748	8.203	8.842	11.009	15.450	21.748	33.891	51.241
SVR	55.655	39.398	40.116	42.161	62.349	97.647	157.309	250.867	465.365	736.960

As can be seen in Table 2, ANFIS obtains the best performance in terms of ARL₁ for small shifts in intercept and slope, given $\phi = 0.1$ and $\phi = 0.5$. As for $\phi = 0.9$, mostly ANNBP ($\lambda = 0.6, 0.8, \eta = 0.075, 0.1, 0.125$) and LSTM ($\lambda = 0.2, \eta = 0.025, 0.05$) produce better results. These results are also reflected in the values of SDRL₁ with only a few exceptions (see Table 3). As for small shifts in the standard deviation, RBF outperforms other machine learning techniques, with only two exceptions for $\phi = 0.9, \gamma = 1.8$ and 2 , where CNN performs marginally better. However, RBF is not a preferable approach in comparison with ANFIS regarding shifts in intercept and slope.

Note that the deep learning techniques CNN and LSTM are generally not able to reach comparable results for the most shifts, especially for smaller shifts. These findings can be justified before the backdrop that deep neural networks have superiority over shallow networks in problems with a large number of features such as image processing and Natural Language Processing (NLP) [62, 63].

Based on the above findings and numerical evidence regarding a wider range of shifts, ANFIS is the best of the considered machine learning approaches regarding shifts in intercept and slope, and RBF is the superior method regarding shifts in the standard deviation. However, for $\phi = 0.9$, ANFIS and RBF are not throughout the best methods. To illustrate this issue, we additionally provide the results regarding ARL₁ and SDRL₁ of all techniques considering $\phi = 0.9$ with a wider range of shifts in Tables 4 and 5, respectively.

To reach a better judgment about the results of Tables 4 and 5, the *Relative Mean Index* (RMI) is implemented to select the best machine learning technique. This measure is frequently utilized in SPM (see, for example, Han and Tsung [64], Perry [65] and Yeganeh et al. [24]) and considers the average difference from the superior approach in each treatment. The smaller the RMI, the quicker the detection ability. Table 6 reports the values of RMI based on the simulation results given in Tables 4 and 5.

Table 5 SDRL₁ comparisons considering a wider range of shifts in intercept (λ), slope (η) and standard deviation (γ) for the input combination III and $\phi = 0.9$

Method	λ									
	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
ANFIS	140.243	85.831	44.917	18.773	4.235	0.437	0.293	0.283	0.301	0.334
ANNBP	167.322	84.394	26.709	4.140	0.174	0.342	0.416	0.601	0.762	0.920
CNN	214.801	208.212	211.106	215.770	205.786	198.972	201.125	190.211	203.272	206.984
LSTM	143.447	89.651	78.859	52.650	42.535	37.556	28.578	24.562	15.248	9.900
RBF	173.271	107.553	48.343	7.315	0.787	0.367	0.460	0.465	0.496	0.512
SVR	162.555	105.062	63.735	35.477	9.428	1.189	0.777	0.565	0.576	0.659
Method	η									
	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2	0.225	0.25
ANFIS	171.757	140.779	104.133	70.735	46.725	27.908	13.429	4.196	0.896	0.398
ANNBP	221.148	153.464	79.820	43.473	23.121	7.017	1.580	0.359	0.494	0.581
CNN	208.397	213.699	190.458	215.281	228.720	207.836	205.684	203.201	199.125	212.372
LSTM	133.759	113.919	99.049	84.907	60.108	45.684	52.771	39.508	34.940	22.459
RBF	182.574	143.612	117.028	94.293	41.753	15.299	3.223	0.873	0.671	0.448
SVR	138.379	140.612	97.978	79.200	66.815	38.406	26.332	7.422	2.333	0.453
Method	γ									
	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3
ANFIS	61.485	23.262	10.958	6.424	4.150	2.937	2.230	1.846	1.525	1.339
ANNBP	132.698	62.626	35.204	19.247	15.173	10.206	8.682	6.912	6.142	4.722
CNN	47.777	17.547	8.244	4.529	2.950	2.119	1.719	1.160	0.902	0.754
LSTM	81.664	60.809	64.828	63.546	59.199	76.749	103.126	138.067	108.859	113.859
RBF	34.356	12.626	8.112	6.717	8.054	10.517	20.154	24.031	38.982	54.605
SVR	59.005	38.480	40.647	43.803	71.503	115.210	162.199	274.770	525.780	742.972

Table 6 Comparing different machine learning techniques based on the RMI criterion with respect to simulation results given in Tables 4 and 5 ($\phi = 0.9$)

Shift type	ANFIS	ANNBP	CNN	LSTM	RBF	SVR
ARL ₁						
Intercept	0.497	0.495	57.634	5.550	0.271	0.989
Slope	0.952	0.333	36.899	4.859	0.434	1.234
Standard deviation	0.260	2.237	0.068	19.315	4.913	71.041
Average (ARL ₁)	0.570	1.022	31.534	9.908	1.873	24.421
SDRL ₁						
Intercept	2.791	0.502	447.485	62.728	6.000	6.969
Slope	2.446	0.146	167.293	27.493	0.720	4.690
Standard deviation	0.555	4.146	0.080	52.626	15.026	198.764
Average (SDRL ₁)	1.931	1.598	204.953	47.615	7.249	70.141

The order of machine learning techniques in terms of ARL₁ (SDRL₁) for shifts in the intercept is RBF, ANNBP, ANFIS, SVR, LSTM, CNN (ANNBP, ANFIS, RBF, SVR, LSTM, CNN). As for shifts in the slope, the order in terms of both ARL₁ and SDRL₁ is ANNBP, RBF, ANFIS, SVR,

LSTM, CNN, while the order for shifts in the standard deviation is CNN, ANFIS, ANNBP, RBF, LSTM, SVR for both ARL₁ and SDRL₁. To select the machine learning technique with the best average performance regarding all the shifts, we consider the average of the RMI values in

Table 7 Comparing the performance of ANFIS and RBF with conventional statistical competitors considering small shifts in intercept (λ), slope (η) and standard deviation (γ) in terms of ARL_1 ($\phi = 0.1$)

Method	λ				
	0.2	0.4	0.6	0.8	1
T^2	162.800	99.700	52.500	27.400	14.600
EWMA/R	78.600	21.900	10.000	6.200	4.500
EWMA-3	68.900	19.700	9.300	5.900	4.300
ANFIS	43.486	3.096	2.013	2.000	2.000
RBF	177.846	98.682	46.009	23.104	11.437
Method	η				
	0.025	0.05	0.075	0.1	0.125
T^2	181.300	140.500	96.100	62.700	39.700
EWMA/R	129.900	53.900	24.500	13.800	9.200
EWMA-3	111.400	43.400	20.500	12.000	8.300
ANFIS	88.629	19.977	3.890	2.170	2.006
RBF	202.962	146.796	98.082	62.851	36.595
Method	γ				
	1.2	1.4	1.6	1.8	2
T^2	28.200	9.100	4.600	3.000	2.200
EWMA/R	34.000	12.100	6.100	3.900	2.800
EWMA-3	23.900	8.200	4.700	3.300	2.600
ANFIS	49.657	24.790	16.089	12.685	10.510
RBF	27.041	8.772	4.580	2.966	2.248

Table 8 Comparing the performance of ANFIS with conventional statistical competitors considering simultaneous small shifts in intercept (λ and slope (η) in terms of ARL_1 ($\phi = 0.9$)

η	λ					Method
	0.2	0.4	0.6	0.8	1	
0.025	196.150	189.860	178.720	162.830	140.660	T^2
	180.420	155.300	127.930	102.660	81.120	MEWMA
	135.733	60.576	31.535	10.410	3.455	ANFIS
0.05	192.530	183.330	169.010	149.150	123.280	T^2
	163.660	137.270	111.010	88.340	69.260	MEWMA
	94.710	41.753	20.034	4.766	2.871	ANFIS
0.075	186.870	174.180	156.570	132.940	104.470	T^2
	145.870	119.460	95.480	75.230	58.900	MEWMA
	63.953	28.939	10.529	4.012	2.777	ANFIS
0.1	178.750	162.610	141.230	114.160	85.130	T^2
	127.140	102.620	81.470	63.990	49.620	MEWMA
	39.036	15.850	3.564	2.968	2.537	ANFIS
0.125	167.780	148.600	123.370	94.780	67.030	T^2
	109.180	87.680	69.090	54.080	41.400	MEWMA
	19.410	7.530	2.470	1.970	1.910	ANFIS

terms of ARL_1 and $SDRL_1$, respectively. Here, ANFIS (ANNBP) has the best overall average performance because the average of RMI with respect to ARL_1 ($SDRL_1$) is 0.570 (1.598), but the difference in $SDRL_1$ between ANNBP and ANFIS (1.598 vs. 1.931) is negligible. According to these results, ANFIS is selected as the best machine learning technique, and thus we use it as benchmark technique in the subsequent simulations.

The superiority of ANFIS could be due to the establishment of two main training parts including the antecedent part and the conclusion in its structure by the aim of fuzzy *IF-THEN* rules. The effects of the current and previous T^2 statistics, current and previous errors as well as of the current and previous coefficients' estimations, the chance of getting unnatural trends is increased when employing the ANFIS technique. A similar conclusion has been achieved by Aziz Kalteh and Babouei [66] in a way that they indicated the suitable performance of ANFIS in control chart pattern recognition problems.

4.2 Comparing the best machine learning technique with conventional statistical approaches

In this subsection, the performance of ANFIS as the best machine learning technique is compared with conventional statistical control charts based on individual and simultaneous shifts in intercept, slope and standard deviation for $\phi = 0.1$. Because the performance of RBF regarding shifts in the standard deviation is better as compared to ANFIS, RBF is also embedded in the comparisons. Three

competitors entailing T^2 , EWMA/R and EWMA-3 are selected following Noorossana et al. [44]. Table 7 shows the results of ARL_1 for ANFIS and the competitors. Note that the setups are the same as in the previous subsection, and the results of ANFIS and RBF are extracted from Table 2.

As can be seen, ANFIS performs considerably better than the other methods for shifts regarding intercept and slope. There is a large difference especially for smaller shifts; for example, ARL_1 is 3.096 for ANFIS given $\lambda = 0.4$, while ARL_1 is 99.7, 21.9 and 19.7 for the conventional competitors T^2 , EWMA/R and EWMA-3, respectively. However, EWMA-3 and RBF obtain the best results for shifts in the standard deviation. In addition, the statistical methods outperform ANFIS regarding shifts in the standard deviation.

4.3 Comparing the best machine learning technique with conventional statistical approaches considering simultaneous shifts

In industrial processes, simultaneous shifts may occur, so a control chart should also be able to detect such type of shifts. Wang and Lai [45] conducted several simulations about simultaneous shifts with the IC model proposed by Noorossana, Amiri and Soleimani [44]. In the following, we compare ANFIS with two reported schemes in Wang and Lai [45], i.e., T^2 and MEWMA. MEWMA is an advanced version of EWMA control charts that integrates the effect of previous samples in one statistic, and some researchers reported that the performance of this approach in profile monitoring is very well [17, 21, 59]. Table 8 shows the results in terms of ARL_1 for simultaneous shifts in intercept and slope. Note that we restrict the comparison to location parameters and do not consider further simultaneous shifts, which include shifts in the standard deviation, due to the superior performance of ANFIS regarding

shifts in intercept and slope. This is also in line with the approach proposed in Wang and Lai [45].

According to Table 8, ANFIS outperforms both other methods. The deviations in terms of ARL_1 are tangible; for example, the values of ARL_1 are 135.73, 180.42 and 196.15 (1.91, 34.22 and 49.64) for ANFIS, MEWMA and T^2 , respectively, for the smallest (largest) shift $\lambda = 0.2$ and $\eta = 0.025$ ($\lambda = 1$ and $\eta = 0.125$). While there is no distinct trend for the absolute deviations between the values of ARL_1 of ANFIS and each of both competitors for increasing shift sizes, there is generally an increasing behaviour for the corresponding relative deviations, i.e.,

$$\frac{ARL_1^{MEWMA} - ARL_1^{ANFIS}}{ARL_1^{ANFIS}} \text{ and } \frac{ARL_1^{T^2} - ARL_1^{ANFIS}}{ARL_1^{ANFIS}}$$

regarding small shifts (see Table 8). That is, the larger the shifts in slope and/or in intercept, the larger the relative deviations.

As for larger shifts in slope and/or intercept ($\lambda > 1$ and $\eta > 0.125$, not tabulated due to lower relevance), the values of ARL_1 regarding ANFIS decrease to a small extent, while the values of ARL_1 regarding MEWMA and T^2 become closer to the respective ARL_1 values of ANFIS, i.e., we observe a decreasing behavior for the corresponding relative deviations regarding larger shifts. To sum up, ANFIS clearly outperforms both methods in detecting simultaneous shifts and its detection ability is especially better for lower shift sizes.

Statistical control charts usually require the fulfilment of some principal assumptions to reach the best performance, while the occurrence of complicated patterns in the manufacturing process may lead to the invalidity of some of the presumed assumptions and thus to deteriorations in their performance. In contrast, machine learning techniques encounter less challenges provided that input combinations and training procedure are defined properly. It could be concluded from the above results that the machine learning-based techniques, and especially ANFIS, perform better than conventional statistical methods when monitoring autocorrelated profiles; however, some computational effort may be required when implementing these approaches. Due to the existence of online data collection systems in real applications, big data storage and development of high technology computers, this challenge is becoming easier in a way that machine learning-based systems can automatically analyze process data to identify OC situations. To this end, the definition of proper input features, dataset development, relevant training adjustment and acceptable false alarm rates are essential tasks. These steps are usually performed as off-line modelling phase while the operation (online) phase refers to the implementation of the trained model on the online data to detect the process

Table 9 The IC (black) and OC (red) generated responses in the illustrative example

j	y_{ij}			
1	72.253	73.264	72.967	75.188
2	72.345	73.845	73.610	74.569
3	72.333	73.363	72.709	74.548
4	72.193	73.147	73.649	73.942
5	73.157	74.618	73.373	74.962
6	73.239	74.219	73.498	75.270
7	72.352	74.249	73.949	74.897

Table 10 Input and output values for the first seven generated profiles applying ANFIS with input combination III ($\phi = 0.565$)

j	Inputs										O_j
1	0	0	0	0	-0.2	0.001	0.433	4.406	0	-0.08	0.477
2	-0.2	0.0012	0.4332	4.406	0.219	-0.001	-0.26	0.963	-0.08	0.14	0.358
3	0.22	-0.0013	-0.2557	0.963	0.124	-0.004	0.213	3.209	0.14	-0.31	0.230
4	0.12	-0.004	0.213	3.209	0.076	-0.003	-0.37	1.404	-0.31	-0.12	0.318
5	0.08	-0.003	-0.3709	1.404	1.194	-0.007	0.304	9.352	-0.12	0.68	-0.357
6	1.19	-0.007	0.3045	9.352	1.08	-0.005	0.196	1.575	0.68	0.26	0.157
7	1.08	-0.005	0.196	1.575	0.21	0.002	-0.22	0.971	0.26	0.05	0.849

condition [57, 60]. By this procedure, the proposed machine learning-based approach in this paper can improve the monitoring of industrial processes in terms of OC detection ability.

5 Illustrative example

In this section, an illustrative example of a chemical process is conducted to demonstrate a real application. In fact, this example could be considered as a calibration system in the chemical industry. Sometimes, it is necessary to control a chemical process far from the laboratory with remote schemes in which some gas sensors are used as the controller. These sensors are used to monitor such a chemical process over time. Although it is a beneficial approach, it needs new calibration by changing the sensors' adjustments as the variability of gas sensors may affect the performance of the underlying calibration model [67]. These changes may be caused by different chemical materials, process conditions, and equipment movements so their calibration should be checked over time. The approach of profile monitoring can be applied to address calibration issues and for online monitoring of the process.

For these reasons, some studies such as Mahmood et al. [68] and Nadi et al. [36] suggested to apply profile monitoring. Metal oxide (MOX) as a conductometric type of gas sensors is one of the best options due to its sensitivity, operational ease, cost efficiency, rapid response, and the capability of spotting a high number of volatiles. The authors supposed MOX as a sensor and monitored a functional relationship between the resistance (R) of the sensor (i.e., MOX) as the dependent variable and the concentrations of carbon monoxide in the sensor as the independent variable.

To monitor this functional formula, they recorded the results of sensor resistance and different concentration levels over time. Based on the recorded data, the explanatory variables are fixed at 25, 100, 125, and

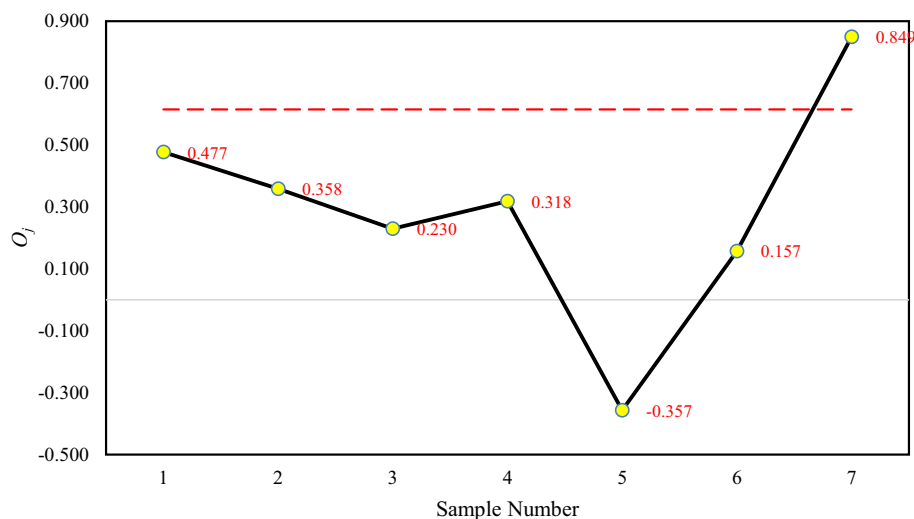
150 ppm. To reach a better performance, it is suggested to change the process situation with some additives. These substances are blended to a special process to accelerate the processing ability of the polymers, improve the characteristics such as durability, stiffness, and enhance the service life. A wide range of additives such as gas, feed, anti-wear, food, fuel, antioxidant, plastic additives have been extended yet. Indeed, gas additives are usually added to the gas sensor processes to adjust the flow of gas during the experiment [67]. However, previous works showed that the relation between resistance and carbon concentration might change in the case of additive materials. To address these issues, Nadi et al. [36] investigated situations related to the before and after of adding the additive material in a way that one additive material was added to the process after time 3278; so, the IC model was extracted from the first 3278 profiles. Considering these profiles, Nadi et al. [36] considered a simple linear IC model with the autocorrelation effect as follows:

$$\begin{aligned}
 Y_{ij} &= 71.741 + 0.0176X_i + \varepsilon_{ij}, \\
 \varepsilon_{ij} &= 0.565\varepsilon_{i(j-1)} + a_{ij}, a_{ij} \sim N(0, 0.142), \\
 i &= 1, 2, 3, 4; j = 1, 2, \dots
 \end{aligned}
 \tag{13}$$

To show the applicability of the proposed method in monitoring the above IC model, Nadi et al. [36] utilized simulations for OC data generation (instead of using the data after the 3278th profile). Following them, we first generated five IC profiles and then continued with the OC profile generation considering a shift in the intercept until reaching an OC signal. The magnitude of the OC shift was considered as 0.15 (or 0.5σ). Table 9 shows the response variables of the generated profiles (the black and red values are IC and OC profiles, respectively).

To specify the detection ability of ANFIS for this data set, it is trained based on the IC model in Eq. (13) and input combination III. Considering $ARL_0 = 200$, the CV is set to 0.615. After adjustment of the CV, the generated data in Table 9 is imported to ANFIS and the output of each input is computed. Table 10 reports the input and output values

Fig. 2 The final chart statistics of the first seven random generated profiles in the illustrative example ($\phi = 0.565$)



for the first seven generated profiles. Hence, ANFIS only needs two OC samples to trigger a signal. The signal in the 7th sample appears because the final statistic exceeds the CV (red horizontal line in Fig. 2), so ANFIS can trigger an OC signal ($O_7 = 0.849 > 0.615 = CV$).

6 Conclusions

In profile monitoring, the error term often does not follow a simple structure and is affected by autocorrelations. For this reason, a novel monitoring scheme for linear autocorrelated profiles with between-autocorrelation of first order in Phase II of process monitoring has been proposed in this paper. Unlike most of the existing methods that use common statistical control charts, this paper employed various machine learning techniques, such as ANFIS, ANNBP, CNN, LSTM, RBF and SVR as a control chart. To this aim, four main steps were defined. In the first step, the structure of the input features and outputs of a machine learning technique were determined, and then, a training data set based on the input features and outputs was generated by simulating IC and OC profiles in the second step. The third step utilized the generated data set to train a machine learning technique, and finally, by the definition of a control limit, the machine learning technique provided information about the process condition in the fourth step.

The study conducted pursued three main objectives. Due to the high importance of input features in machine learning, some input features, which are appropriate to account for between-autocorrelation effects of first order, were defined and compared to achieve the most appropriate input combination. The results indicated that input combination III, which is defined as the union of input combinations I and II and combines both main effects of these input combinations, is the most appropriate one. For the

second aim, different machine learning techniques were compared to identify the most adequate one. Experimental studies showed that ANNBP, CNN, LSTM and SVR were mostly not able to reach a satisfactory detection ability in comparison with ANFIS and RBF. Among ANFIS and RBF, ANFIS was preferable with respect to shifts in intercept and slope, while RBF had the best performance regarding shifts in the standard deviation. This superiority was obvious for low and moderate autocorrelation coefficients (i.e., $\phi = 0.1$ and 0.5), while it was not possible to identify a consistently best method for a larger value ($\phi = 0.9$). To address this issue, we additionally implemented an overall performance measure, called RMI. Following RMI, we found that ANFIS turns out to be the method with the best overall average performance for $\phi = 0.9$. The third aim of this study was to compare machine learning-based techniques with statistical control charts. This comparison led to the result that the detection ability of ANFIS outperformed all the competitors regarding shifts in intercept and slope. However, the detection ability of ANFIS regarding shifts in the standard deviation was inferior compared to the selected statistical control charts. In this regard, the EWMA-3 control chart performed better, and the best machine learning technique for this purpose was RBF (with a performance that is hardly worse than that of EWMA-3). Hence, machine learning-based control charts, and ANFIS in the first place, are suggested to be utilized in profiles that are characterized by between-sample AR(1) autocorrelation to considerably improve the detection ability of the control chart.

Employing the proposed novel input features with other machine learning techniques and other profile types such as nonlinear or Generalized Linear Models (GLMs) in the presence of autocorrelations could be a promising avenue for potential future research. Also, implementing the proposed method in profiles that are characterized by within

sample autocorrelation or in profiles with other autocorrelation patterns, such as ARMA or VARMA are further suggestions for potential future directions.

Acknowledgements The authors would like to thank four anonymous reviewers for their valuable feedback and suggestions, which were important and helpful to significantly improve the paper.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability statement Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Declarations

Conflicts of interest (i) this manuscript is the authors' original work, which has not been published nor submitted simultaneously elsewhere; (ii) all authors have checked the manuscript and agreed to the submission, and (iii) there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Montgomery DC (2019) Introduction to statistical quality control, 8th edn. Wiley, New York
- Weese M, Martinez W, Megahed FM, Jones-Farmer LA (2016) Statistical learning methods applied to process monitoring: an overview and perspective. *J Qual Technol* 48:4–24
- Tran PH, Ahmadi Nadi A, Nguyen TH, Tran KD, Tran KP (2022) Application of machine learning in statistical process control charts: a survey and perspective. In: Tran KP (ed) Control charts and machine learning for anomaly detection in manufacturing. Springer, Berlin, pp 7–42
- Fathizadan S, Niaki STA, Noorossana R (2017) Using independent component analysis to monitor 2-D geometric specifications. *Qual Reliab Eng Int* 33:2075–2087
- Abbas T, Rafique F, Mahmood T, Riaz M (2019) Efficient phase II monitoring methods for linear profiles under the random effect model. *IEEE Access* 7:148278–148296
- Yeganeh A, Pourpanah F, Shadman A (2021) An ANN-based ensemble model for change point estimation in control charts. *Appl Soft Comput* 110:107604
- Johannssen A, Chukhrova N, Castagliola P (2021) The performance of the hypergeometric np chart with estimated parameter. *Eur J Oper Res*. <https://doi.org/10.1016/j.ejor.2021.06.056>
- Ji C, Sun W (2022) A review on data-driven process monitoring methods: characterization and mining of industrial data. *Processes*. <https://doi.org/10.3390/pr10020335>
- Vanli OA, Castillo ED (2019) Statistical process control in manufacturing. In: Baillieul J, Samad T (eds) Encyclopedia of systems and control. Springer, London, pp 1–8
- Knott S (2021) Steady-state average run length(s): methodology, formulas, and numerics. *Seq Anal*. <https://doi.org/10.1080/07474946.2021.1940501>
- Chukhrova N, Johannssen A (2019) Hypergeometric p-chart with dynamic probability control limits for monitoring processes with variable sample and population sizes. *Comput Ind Eng* 136:681–701
- Chukhrova N, Johannssen A (2019) Improved control charts for fraction non-conforming based on hypergeometric distribution. *Comput Ind Eng* 128:795–806
- Yeganeh A, Chukhrova N, Johannssen A, Fotuhi H (2023) A network surveillance approach using machine learning based control charts. *Expert Syst Appl* 219:119660
- Kang L, Albin SL (2000) On-line monitoring when the process yields a linear profile. *J Qual Technol* 32:418–426
- Jones CL, Abdel-Salam A-SG, Mays DA (2021) Practitioners guide on parametric, nonparametric, and semiparametric profile monitoring. *Qual Reliab Eng Int* 37:857–881
- Zhao C, Du S, Deng Y, Li G, Huang D (2020) Circular and cylindrical profile monitoring considering spatial correlations. *J Manuf Syst* 54:35–49
- Yeganeh A, Shadman A, Amiri A (2021) A novel run rules based MEWMA scheme for monitoring general linear profiles. *Comput Ind Eng* 152:107031
- Yeganeh A, Shadman AR, Triantafyllou IS, Shongwe SC, Abbasi SA (2021) Run rules-based EWMA charts for efficient monitoring of profile parameters. *IEEE Access* 9:38503–38521
- Zhu J, Lin DKJ (2009) Monitoring the slopes of linear profiles. *Qual Eng* 22:1–12
- Kim K, Mahmoud MA, Woodall WH (2003) On the monitoring of linear profiles. *J Qual Technol* 35:317–328
- Zou C, Tsung F, Wang Z (2007) Monitoring general linear profiles using multivariate exponentially weighted moving average schemes. *Technometrics* 49:395–408
- Yeganeh A, Shadman A, Abbasi SA (2022) Enhancing the detection ability of control charts in profile monitoring by adding RBF ensemble model. *Neural Comput Appl* 34:9733–9757
- Ding D, Tsung F, Li J (2017) Ordinal profile monitoring with random explanatory variables. *Int J Prod Res* 55:736–749
- Yeganeh A, Abbasi SA, Shongwe SC (2021) A novel simulation-based adaptive MEWMA approach for monitoring linear and logistic profiles. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2021.3107482>
- Mohammadzadeh M, Yeganeh A, Shadman A (2021) Monitoring logistic profiles using variable sample interval approach. *Comput Ind Eng* 158:107438
- Chou S-H, Chang SI, Tsai T-R (2014) On monitoring of multiple non-linear profiles. *Int J Prod Res* 52:3209–3224
- Zou C, Tsung F, Wang Z (2008) Monitoring profiles based on nonparametric regression methods. *Technometrics* 50:512–526
- Zeng L, Neogi S, Zhou Q (2014) Robust phase I monitoring of profile data with application in low-E glass manufacturing processes. *J Manuf Syst* 33:508–521
- Zhou P, Liu P, Wang S, Zhang C, Zhang J, Li S (2022) Functional state-space model for multi-channel autoregressive profiles with application in advanced manufacturing. *J Manuf Syst* 64:356–371
- Yao C, Li Z, He C, Zhang J (2020) A Phase II control chart based on the weighted likelihood ratio test for monitoring polynomial profiles. *J Stat Comput Simul* 90:676–698
- Zhang Y, He Z, Zhang M, Wang Q (2016) A score-test-based EWMA control chart for detecting prespecified quadratic changes in linear profiles. *Qual Reliab Eng Int* 32:921–931

32. Maleki MR, Amiri A, Castagliola P (2018) An overview on recent profile monitoring papers (2008–2018) based on conceptual classification scheme. *Comput Ind Eng* 126:705–728
33. Woodall WH (2007) Current research on profile monitoring. *Production* 17:420–425
34. Fan S-KS, Jen C-H, Lee J-X (2019) Profile monitoring for autocorrelated reflow processes with small samples. *Processes*. <https://doi.org/10.3390/pr7020104>
35. Khalili S, Noorossana R (2022) Online monitoring of autocorrelated multivariate linear profiles via multivariate mixed models. *Qual Technol Quant Manag* 19:319–340
36. Nadi AA, Yeganeh A, Shadman A (2023) Monitoring simple linear profiles in the presence of within- and between-profile autocorrelation. *Qual Reliab Eng Int*. <https://doi.org/10.1002/qre.3254>
37. Yao J, Xian X, Wang C (2023) Adaptive sampling for monitoring multi-profile data with within-and-between profile correlation. *Technometrics*. <https://doi.org/10.1080/00401706.2023.2166125>
38. Soleimani P, Noorossana R, Amiri A (2009) Simple linear profiles monitoring in the presence of within profile autocorrelation. *Comput Ind Eng* 57:1015–1021
39. Cheng T-C, Yang S-F (2018) Monitoring profile based on a linear regression model with correlated errors. *Qual Technol Quant Manag* 15:393–412
40. Rahimi SB, Amiri A, Ghashghaei R (2021) Simultaneous monitoring of mean vector and covariance matrix of multivariate simple linear profiles in the presence of within profile autocorrelation. *Commun Stat Simul Comput* 50:1791–1808
41. Jensen WA, Birch JB, Woodall WH (2008) Monitoring correlation within linear profiles using mixed models. *J Qual Technol* 40:167–183
42. Jensen WA, Birch JB (2009) Profile monitoring via nonlinear mixed models. *J Qual Technol* 41:18–34
43. Narvand A, Soleimani P, Raissi S (2013) Phase II monitoring of auto-correlated linear profiles using linear mixed model. *J Ind Eng Int* 9:12
44. Noorossana R, Amiri A, Soleimani P (2008) On the monitoring of autocorrelated linear profiles. *Commun Stat Theory Methods* 37:425–442
45. Wang Y-HT, Lai Y (2019) Monitoring of autocorrelated general linear profiles. *J Stat Comput Simul* 89:519–535
46. Khedmati M, Niaki STA (2016) Phase II monitoring of general linear profiles in the presence of between-profile autocorrelation. *Qual Reliab Eng Int* 32:443–452
47. Koosha M, Amiri A (2013) Generalized linear mixed model for monitoring autocorrelated logistic regression profiles. *Int J Adv Manufact Technol* 64:487–495
48. Wang Y-HT, Huang W-H (2017) Phase II monitoring and diagnosis of autocorrelated simple linear profiles. *Comput Ind Eng* 112:57–70
49. Pacella M, Semeraro Q (2011) Monitoring roundness profiles based on an unsupervised neural network algorithm. *Comput Ind Eng* 60:677–689
50. Li C-I, Pan J-N, Liao C-H (2019) Monitoring nonlinear profile data using support vector regression method. *Qual Reliab Eng Int* 35:127–135
51. Sergin ND, Yan H (2021) Toward a better monitoring statistic for profile monitoring via variational autoencoders. *J Qual Technol*. <https://doi.org/10.1080/00224065.2021.1903821>
52. Yeganeh A, Abbasi SA, Pourpanah F, Shadman A, Johannssen A, Chukhrova N (2022) An ensemble neural network framework for improving the detection ability of a base control chart in non-parametric profile monitoring. *Expert Syst Appl* 204:117572
53. Hosseini-fard SZ, Abdollahian M, Zeephongsekul P (2011) Application of artificial neural networks in linear profile monitoring. *Expert Syst Appl* 38:4920–4928
54. Yeganeh A, Shadman A (2021) Monitoring linear profiles using Artificial Neural Networks with run rules. *Expert Syst Appl* 168:114237
55. Noorossana R, Niaki STA, Izadbakhsh H (2015) Statistical monitoring of nominal logistic profiles in phase II. *Commun Stat Theory Methods* 44:2689–2704
56. Yeganeh A, Shadman A (2021) Using evolutionary artificial neural networks in monitoring binary and polytomous logistic profiles. *J Manuf Syst* 61:546–561
57. Chen S, Yu J, Wang S (2020) Monitoring of complex profiles based on deep stacked denoising autoencoders. *Comput Ind Eng* 143:106402
58. Riaz M, Mahmood T, Abbasi SA, Abbas N, Ahmad S (2017) Linear profile monitoring using EWMA structure under ranked set schemes. *Int J Adv Manufact Technol* 91:2751–2775
59. Huwang L, Wang Y-HT, Xue S, Zou C (2014) Monitoring general linear profiles using simultaneous confidence sets schemes. *Comput Ind Eng* 68:1–12
60. Yu J, Zheng X, Wang S (2019) A deep autoencoder feature learning method for process pattern recognition. *J Process Control* 79:1–15
61. Cherkassky V, Ma Y (2004) Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Netw* 17:113–126
62. Das HS, Roy P (2019) Chapter 5—a deep dive into deep learning techniques for solving spoken language identification problems. In: Dey N (ed) *Intelligent speech signal processing*. Academic Press, Cambridge, pp 81–100
63. Borg A, Boldt M, Rosander O, Ahlstrand J (2021) E-mail classification with machine learning and word embeddings for improved customer support. *Neural Comput Appl* 33:1881–1902
64. Han D, Tsung F (2006) A reference-free cuscore chart for dynamic mean change detection and a unified framework for charting performance comparison. *J Am Stat Assoc* 101:368–386
65. Perry MB (2020) An EWMA control chart for categorical processes with applications to social network monitoring. *J Qual Technol* 52:182–197
66. Aziz Kalteh A, Babouei S (2020) Control chart patterns recognition using ANFIS with new training algorithm and intelligent utilization of shape and statistical features. *ISA Trans* 102:12–22
67. Fonollosa J, Fernández L, Gutiérrez-Gálvez A, Huerta R, Marco S (2016) Calibration transfer and drift counteraction in chemical sensor arrays using Direct Standardization. *Sens Actuators B Chem* 236:1044–1053
68. Mahmood T, Riaz M, Hafidz Omar M, Xie M (2018) Alternative methods for the simultaneous monitoring of simple linear profile parameters. *Int J Adv Manufact Technol* 97:2851–2871

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