# Estimation and generation of training patterns for Control Chart Pattern Recognition

De La Torre Gutierrez, Hector; Pham, Duc

*DOI:*
10.1016/j.cie.2016.02.016

# Estimation and generation of training patterns for Control Chart Pattern Recognition

H. De-la-Torre[a,*], D. T. Pham[a]

[a]*School of Engineering, Edgbaston, Birmingham, West Midlands B15 2TT*

**Abstract**

Most applications of machine learning (ML) algorithms to control chart pattern recognition (CCPR) have focused on pattern detection and identification, rather than obtaining more detailed information about the pattern, which is important for effective assignable cause analysis. If real control chart data is not available for training purposes, synthetic patterns must be generated. Furthermore, pattern recognition accuracies achieved by different CCPR systems are usually not comparable since these were developed with different training data. How to create a diverse range of patterns for designing CCPR systems that can be compared and that are able to recognise a greater variety of patterns is an issue that needs studying.

This paper presents a scheme to generate training patterns that addresses this issue of diversity and comparability. The scheme also comprises change point detection and mean change categorisation methods that implement nonlinear models (NLMs) for estimating abnormal pattern parameters. The effect of this new pattern generation scheme on the accuracy of pattern recognition has been studied using two ML algorithms: Support Vector Machine (SVM) and Probabilistic Neural Network (PNN).

With the proposed pattern generation scheme, the mean pattern recognition accuracy was increased by 6.90% and 8.42% when SVM and PNN were used, respectively.

*Keywords:* `Pattern generation`, `Support Vector Machine`, `Probabilistic Neural Network`, `Pattern parameters estimation`, `Control chart pattern recognition`

## 1. INTRODUCTION

Root Cause Analysis is an important task in most quality assurance systems. A good quality assurance system is one that can quickly and precisely find and address quality failures. To achieve this, quality assurance is assisted by Statistical Quality Control in order to monitor the production system and its critical quality features. In recent production systems, the identification of causes of poor quality has been a comprehensive task that sometimes can be uncertain because of human intervention. Statistical Quality Control is used to sort out this uncertainty problem, but even so, human intervention is required in conventional Statistical Quality Control systems. An example of this is the identification of patterns in Statistical Process Control Charts and the one-one matching of these patterns with their assignable causes. According to the Statistical Quality Control Handbook created by [1], control charts can exhibit 15 types of patterns. These include seven simple patterns: Normal (NOR), Upward/Downward Trends (UT/DT), Cycles (CYC), Downward/Upward Shifts (DS/US) and Systematic (SYS). The remaining patterns are either combinations or particular cases of some of

*Corresponding author
*Email address:* `hxd394@bham.ac.uk` (H. De-la-Torre)

**Nomenclature**

| | | | |
|---|---|---|---|
| ANOVA | analysis of variance | sin | sine function |
| BESSEL | bessel kernel | SSE | sum of squared error |
| CCPR | control chart pattern recognition | SVM | Support Vector Machine |
| CI | confidence interval | SYS | systematic |
| CYC | cyclic | TANH | hyperbolic tangent kernel |
| DS | Downward Shift | US | Upward Shift |
| DT | Downward Trend | UT | Upward Trend |
| LAPLA | laplace kernel | $y_t$ | measurement of the quality characteristic under study at time t |
| ML | machine learning | $\alpha$ | significance level |
| NLM | nonlinear model | $\beta$ | abnormal pattern parameter |
| NOR | normal | $\epsilon_t$ | random error of the fitted regression model at time t |
| Nt | inherent noise at time t | $\sigma_N$ | standard deviation of the variable N |
| PNN | Probabilistic Neural Network | $\mu$ | mean value |
| PGS | pattern generation scheme | $\tau$ | time when a change point occurs |
| RBF | radial basis function kernel | | |

these. The seven simple patterns can be divided into two groups: steady and non-steady mean. Normal, Cyclic and Systematic patterns belong to the steady-mean group, and Downward/Upward Trends and Downward/ Upward Shifts belong to the non-steady-mean group. Figure 1 shows the seven simple control chart patterns.

For pattern identification and cause assignment, it is necessary to identify abnormalities in the current control chart and extract information such as frequency, magnitude, time when a certain abnormality happened, etc. This data is obtained to help the root cause analysis in the efficient identification of assignable causes of poor quality in the production system; such information can help to distinguish between patterns that are identified as the same but have different root causes, e.g., a Cyclic pattern with period of 12 might be produced due to wear of a tool, while another pattern with period equal to 24 might be caused by variations in the input voltage.

In recent years, Machine learning (ML) algorithms have been implemented for the efficient identification of control chart patterns. The first phase in the operation of a supervised learning ML algorithm consists in training the algorithm by presenting patterns similar to those to be classified afterwards. Ideally, observation samples (process data) should be collected from the real process environment and be used as inputs during the training. However, since a large amount of data is required for control chart pattern recognition (CCPR), synthetic samples need to be generated. This is commonly done using Monte-Carlo simulation [2].

A drawback of Monte-Carlo simulation is that this method is not sufficiently robust to noise which can greatly affect the generalisation ability of the algorithm if special care is not paid to the patterns used for training. Consequently, the statistical properties of the patterns generated for training the algorithm can be altered, e.g., a positive slope can appear shallower or even disappear altogether due to noise, and so the ML algorithm is trained with a-priori misclassified patterns, thus setting incorrect pattern classification boundaries.

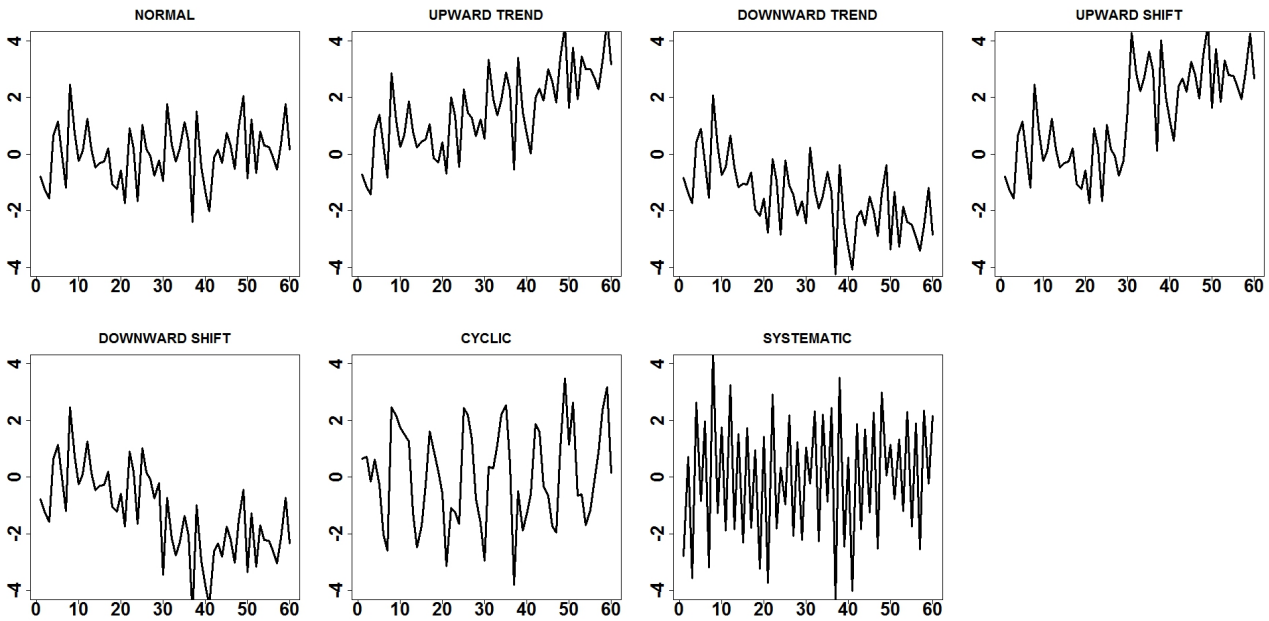


Figure 1: Seven simple control chart patterns

The aim of this research is to develop a pattern generation scheme (PGS) for CCPR that is robust to variations in the pattern parameters used for training. This PGS randomises all the parameters that categorise each abnormal pattern, such as cycle amplitude, periodicity, slope, shift magnitude, change point position and systematic departure [3, 2]. Furthermore, the proposed PGS can be used by other authors as a standard scheme for producing data for training CCPR systems that can be compared.

The PGS presented in this paper comprises three steps: initial pattern generation, mean change classification and pattern categorisation. In the first step, the generality of the CCPR system is ensured due to the wide-range of training pattern parameters values used and the random assignment of these during pattern generation. In the second step, break points (if there are any) are detected. Due to noise, this step is not straightforward. First, the most likely position of a potential break point is determined. Then, the statistical significance of the magnitude of this most likely break point is computed to decide whether there really is a break point. If so, the amount of shift of the pattern is determined and the pattern is categorised as US or DS.

The last step of the proposed PGS concerns the final classification of the training patterns. Once it has been determined that the pattern under study has either a steady mean, continuous change in mean or step change in mean, the pattern is reclassified by fitting a NLM to the pattern data taking account of the statistical significance of the parameters.

The performance of the PGS as represented by the recognition accuracy achieved by two very different ML algorithms, Support Vector Machines (SVMs) and Probabilistic Neural Networks (PNNs) was measured. These algorithms are based on statistical learning theory. The main difference in their learning methods relates to their risk minimisation [4]. In the case of SVM, structural risk minimisation is used to minimise an upper bound based on the expected risk, whereas in PNN, traditional empirical risk minimisation is adopted to minimise the error during training [5]. Another advantage of SVMs and PNNs is the number or free parameters to be set, being only one in the case of PNNs, and at most four for SVMs. These two algorithms have achieved good recognition accuracy, with PNN requiring little training data.

When the CCPR system is in use and a NLM is fitted to the CC data of the already identified pattern, it

is expected that the most statistically significant parameter linked to one of the pattern classes will correlate with the pattern class determined by the CCPR system. This is because the CCPR system was trained with patterns whose membership to given classes was decided by the statistical significance level of their parameters. For example, if a pattern is classified as an UT by the CCPR system, the slope will be the most significant parameter when a NLM is fitted to the data.

The remainder of this paper is organised as follows. Section 2 reviews the issues studied here. The proposed PGS for supervised ML algorithms is introduced in section 3. Section 4 describes the results obtained. Finally, conclusions of this work and suggestions for further research are provided in section 5.

## 2. LITERATURE REVIEW

This section reviews previous work on pattern generation for CCPR, application of PNN and SVM in CCPR and the estimation of abnormal pattern parameters when ML algorithms are used for identifying patterns in control charts.

### *2.1. Pattern Generation for CCPR*

Little attention has been paid to the pattern parameters used during training. Most authors have adopted the pseudorandom number generator proposed by [6] and implemented as default in software such as *R and MATLAB* for the generation of the inherent noise.

It was found that the values of the abnormal pattern parameters used for training and testing the recognition systems varied greatly. Table 1 shows the range of the parameter values used for generating patterns in the most relevant recent works. However, in the table, it can be observed that there are two parameters that have not been randomised during pattern generation, namely, the break point position and the period of the Cyclic patterns. These two parameters must be randomly generated in a proper PGS in order to enable to the recognition system to identify a broader variety of pattern types and magnitudes.

Table 1: Maximums and minimums of the training pattern parameters

| | **Pattern type** | **UT, slope** | | **DT, slope** | | **US / DS break point** | | **US, Magnitude** | | **DS, Magnitude** | | **CYC, Amplitude** | | **CYC, Frequency** | | **SYS, systematic departure** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Range** | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max |
| **Author(s)** | [7] | 0.005 | 0.605 | -0.605 | -0.005 | One point | | 0.005 | 1.805 | -1.805 | -0.005 | 0.005 | 1.805 | Fixed | | 0.005 | 1.805 |
| | [8] | 0.05 | 0.10 | -0.10 | -0.05 | One point | | 1.5 | 2.5 | -2.5 | -1.5 | 1.5 | 2.5 | 8, 16 | | 1 | 3 |
| | [9] | 0.05 | 0.10 | -0.10 | -0.05 | One point | | 2.5 | 3.0 | -3.0 | -2.5 | 1.5 | 2.5 | 20 | | 1 | 3 |
| | [5] | 0.00 | 0.22 | -0.22 | 0.00 | One point | | 0.0 | 2.5 | -2.5 | 0.0 | 1 | 2.5 | 8 | | 1.5 | 3 |
| | [10] | 0.05 | 0.10 | -0.10 | -0.05 | Three points | | 1.5 | 2.5 | -2.5 | -1.5 | 1.5 | 2.5 | 8, 16 | | 1 | 3 |
| | [11] | 0.05 | 0.10 | -0.10 | -0.05 | Three points | | 1.5 | 2.5 | -2.5 | -1.5 | 1.5 | 2.5 | 8,16 | | 1 | 3 |
| | [12] | 0.015 | 0.025 | -0.025 | -0.015 | One point | | 0.7 | 2.5 | -2.5 | -0.7 | 0.5 | 2.5 | 10 | | – | – |
| | [13] | 0.20 | 0.50 | -0.50 | -0.20 | One point | | 1.5 | 4.0 | -4.0 | -1.5 | 0 | 3 | 4 | 12 | – | – |
| | [14] | 0.05 | 0.10 | -0.10 | -0.05 | Three points | | 1.5 | 2.5 | -2.5 | -1.5 | 1.5 | 2.5 | 8,16 | | 1 | 3 |
| | [15] | 0.10 | 0.26 | -0.26 | -0.10 | One point | | 1 | 3 | -3 | -1 | Not given | | 7,8,9 | | – | – |
| | [16] | 0.1 | 0.25 | -0.25 | -0.10 | One point | | 1.5 | 3.0 | -3.0 | -1.5 | 1.5 | 3.0 | 16 | | – | – |
| | [17] | 0.2 | 0.5 | -0.5 | -0.2 | One point | | 1.5 | 4.0 | -4.0 | -1.5 | 0 | 3 | 4 | 12 | – | – |

So far, [18] have been the only authors to have studied the training parameter problem. They used design of experiments to assess the effect of some of the parameter ranges used during pattern generation, finding that the values of maximum and minimum shifts and slopes greatly affect Type-1 and Type-2 errors. Other authors such as [19] and [20] studied other parameters like Neural Network configuration and inspection window length and how they influenced the performance of the learning algorithm.

### *2.2. SVM and PNN*

SVM is a relatively recent algorithm in the field of ML. Within less than two decades of being created, many of its advantages with respect to the best existing methods have become evident: generalisation capacity, ease of use and solution uniqueness [21]. SVMs can deal with nonlinear formulations, provide a trade-off between dimensionality (space complexity) and accuracy and have shown good results in pattern recognition applications. Further details on SVMs can be found in [22], [23] and [24].

SVMs have been applied to diverse problems, from text classification [25], object recognition [26], image classification [27], and bioinformatics [28, 29].

As a classification system, SVMs have also been used for CCPR and abnormal patterns parameter identification. Authors such as [7] and [30] achieved good pattern recognition accuracies with SVMs. Other authors such as [11, 5] and [8] have utilised signal processing techniques such as Independent Component Analysis and Wavelet transforms to pre-process the control chart data, and also achieved good pattern recognition accuracies.

The PNN is a feed-forward neural network based on the Bayesian Criterion and Parzen Window for Probability Distribution Function estimation, also showing good pattern recognition accuracies. The most important advantage of the PNN is that training is easy and instantaneous. Other advantages offered by the PNN are that only one parameter (so-called smoothing parameter) has to be set by the user, good accuracy can be achieved even with small samples, and the network is tolerant to erroneous data and operates completely in parallel without requiring feedback from the individual neurons to the inputs [31]. For further details, see [31, 32] and [33].

PNNs have been applied to diverse tasks such as pattern recognition [34, 33, 35, 36, 37], image processing [38, 39] and many others [40, 41, 42, 43]. [44] is the only author to have applied PNN to CCPR, combining it with Wavelet transforms to identify some abnormal pattern parameters.

### *2.3. Estimation of parameters of abnormal patterns*

In the literature, models that deal with the recognition and classication of patterns in addition to estimating their corresponding parameters are very rare [45].

[46] developed an intelligent system capable of recognising common abnormal patterns and identifying their characteristics. However, their method was not statistically robust as they used small ranges of abnormal patterns parameters during the training of the recognition system, and did not estimate the parameters at the end of the system, considering them only as another characteristic to identify. Other authors such as [47, 48, 15, 49] created a sequence of intelligent sub-systems where in a first step, the pattern was identified by a sub-system and the magnitude of the pattern subsequently determined by another sub-system. Training and testing were carried out with patterns generated from a narrow range of pattern parameters.

Possible misclassification due to the effect of noise during pattern generation was ignored in all the reviewed papers; thus, the potential increase in the probability of Type 1 and Type 2 errors was neglected.

## 3. PROPOSED SCHEME

### *3.1. Initial pattern generation*

Patterns are generated with only one possible change point in the time window examined; e.g. only one shift pattern can occur in the time window examined.

Firstly, a random vector, $N_t$, normally distributed with zero mean and variance $\sigma_N^2$ is generated by applying the pseudorandom number generator proposed by [6].

The pattern is synthesised using the random vector $N_t$ the pattern expressions A1-A5 given in the Appendix. The values of the parameters in A1-A5 are chosen randomly between the maxima and minima shown in Table 2

Table 2: Parameters used during the pattern generation, in $\sigma_N$ terms

| **Pattern type** | **Name** | **Parameter** | **Minimum** | **Maximum** |
|---|---|---|---|---|
| UT | Slope | $\beta_1$ | 0.01 | 0.3 |
| DT | Slope | $\beta_1$ | -0.3 | -0.01 |
| US \ DS | Break point position | $d$ | 0 | 1 |
| US | Magnitude of the mean shift | $\beta_2$ | 0.1 | 3 |
| DS | Magnitude of the mean shift | $\beta_2$ | -3 | -0.1 |
| CYC | Cycle amplitude | $\beta_3$ | 0.1 | 3 |
| CYC | Wave frequency | $\beta_4$ | 3 | 16 |
| SYS | Systematic departure | $\beta_5$ | 0.1 | 3 |

Each of the parameters shown in Table 2 was designed to follow a uniform distribution in the specified range. The maximum and minimum values of the slope were determined to remain within the $6\sigma$ limits after 10 observations. The values for the period of the Cyclic patterns were set to have at least four recurrences in the observation window. The minimum for the other patterns was set near to zero, and the maximum to stay inside the $6\sigma$ limits in the observation window.

### *3.2. Mean change classification*

Every pattern created in the previous step is employed in the proposed mean change classification.

As mentioned above, the proposed methodology for determining the type of mean change occurring in the control chart is based on the identification of potential change points and nested NLMs. A change point estimator focuses on finding the point in time where the process parameters have changed because of some assignable cause(s), i.e. it estimates the time when a change in the mean occurred. For example, consider a normal process where $y_i \sim N(\mu_0, \sigma^2)$,$i = 1, 2, 3, \ldots, \tau$ and $y_i \sim N(\mu_1, \sigma^2)$,$i = \tau + 1, \tau + 2, \ldots, n$. That is, the process follows a normal distribution with mean $\mu_0$ and variance $\sigma_2$, until the change point, $\tau$. Following

the change point $\tau$, one parameter of the process has changed (from $\tau_0$ to $\tau_1$). The aim is to estimate $\tau$ and the difference between $\mu_0$ and $\mu_1$, $\tau$ being the time when the mean changed Upward/Downward Shifts are considered the only patterns where a change point is detected.

The proposed mean change point classification methodology consists of two stages: identifying the most likely change point and evaluating that change point by means of an F-test for nested models.

*3.2.1. Most likely change point fitting all the possible NLMs*

Considering the number of parameters to be estimated and the degrees of freedom for the significance tests, a minimum sample of 15 is desirable to have a good estimation of regression parameters. Therefore, all possible piecewise regression models are fitted assuming change points at: $\tau = 16, 17, \ldots, (n-15)$. The Bayesian Information Criterion (BIC) [50] is extracted from each possible fitted model in order to determine which of the fitted models is the most likely to have a change point, with $\tau$ chosen to correspond to the least Bayesian Information Criterion value. The following is the NLM assuming the existence of a change point:

$$y_t = \beta_0 + \beta_1 t + \beta_2 d + \beta_3 sin\left(\frac{2\pi t}{\beta_4}\right) + \beta_5(-1)^t + \epsilon_t \tag{1}$$

where $d$ and the parameters $\beta_1$ to $\beta_5$ are as defined in the Appendix section. $\beta_0$ and $\epsilon_t$ represent the intercept with the y-axis of the regression model and the random error at time t, respectively.

*3.2.2. Nested models and model selection*

The model represented by equation 2 below corresponds to a NLM not assuming the existence of change points, i.e., only continuous change in mean is considered. It is observed that the model that does not take account of the existence of change points is fully contained in equation 1 which relates to a model with the most likely change point. Thus, the mean change categorisation problem becomes a selection between two nested NLMs, considering the model fitted under the supposition of no change point as the reduced model and the model fitted with the most likely change point as the full model.

$$y_t = \beta_0 + \beta_1 t + \beta_3 sin\left(\frac{2\pi t}{\beta_4}\right) + \beta_5(-1)^t + \epsilon_t \tag{2}$$

As it is unknown if there is a break point and if its magnitude is statistically significant, the selection of which model best fits the patterns is a model selection problem with nested models, raising the following two hypotheses:

$H_0$: There are no break points (the reduced model fits better) $H_1$: A break point is detected (the full model fits better)

An F-test for nested models is used [51, 52] in order to determine which hypothesis to reject, i.e.

$$F_{\nu_2}^{\nu_1} = (SSE_{full} - SSE_{reduced})/(SSE_{full}/(n-k-1)) \tag{3}$$

where k is the number of parameters of the full model excluding the intercept, and F belongs to an F-distribution with one degree of freedom in the numerator ($\nu_1 = 1$) and *n-k-1* degrees of freedom in the denominator ($\nu_2 = n-6$). Three significance levels of $\alpha = 0.01, 0.02$ and $0.03$ are established in this research to determine whether the control chart under study has a break point or not.

*3.3. Pattern classification*

If it is decided that the studied pattern has a change point, a model of type 1 (represented by equation 1) is fitted to the data; otherwise, a model of type 2 (represented by equation 2) is adopted.

The statistical significance of the parameters $\beta$s related of the model fitted in the previous step determines the class to which the pattern belongs. In this work, three significance levels were chosen, $\alpha = 0.01, 0.02$ and 0.03. The pattern parameter whose p-value is less than the significance level will determine the class of the pattern. For further information about statistical testing significance in NLMs, refer to [53]

Figure 2 shows a flowchart that summarises the scheme proposed for the generation of patterns.

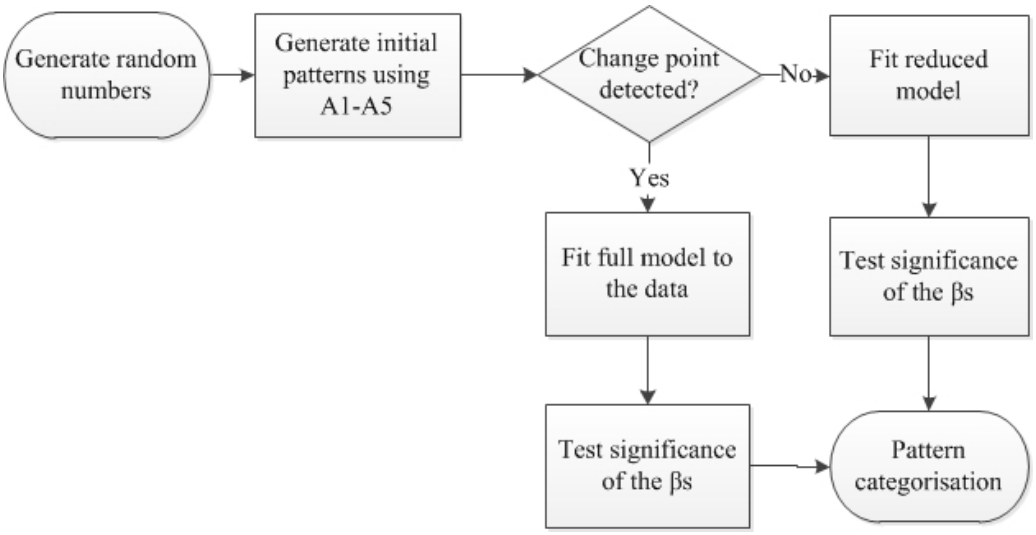


Figure 2: Flowchart of the proposed pattern generation scheme

Once the CCPR system has been trained, it will be able to recognise a broad range of patterns. If the system identifies a pattern as abnormal, the parameter(s) related to this pattern can be statistically estimated by applying equation 1 or 2.

*3.4. Illustrative example*

Seven patterns were generated using equations A1-A5 sampled at times $t = 1, 2, , n = 60$, Figure 3 depicts these patterns.

The third column of Table 3 gives the parameter values used for the initial pattern generation. The fourth column shows which model fits each pattern better and the p-values obtained from the F-test. To obtain the fourth column, it was necessary to fit two models to each pattern. Then, using the SSEs from these, the F-value and its respective p-value were obtained to determine which model fits better. The fifth column lists the p-values of the significant term of the model that was determined to fit better in the previous step. Finally, the last column gives the final class of the pattern according to the proposed PGS. The significance level chosen for this example is $\alpha = 0.01$.

Regarding the Trend patterns, it was observed that DT was reclassified as NOR since none of the pattern parameters of the model that fitted it better was statistically significant, this being opposite to the case of the UT pattern where the estimated slope of 0.0219 was statistically significant. For the US pattern, it was observed that the Full model fitted it better and the most likely break point of magnitude 1.4098 was observed at $\tau$=29.

For the case of DS, it is worth showing the operation of the PGS step by step. Firstly, a full model and a reduced model were fitted, giving the models shown in equations 4 and 5.

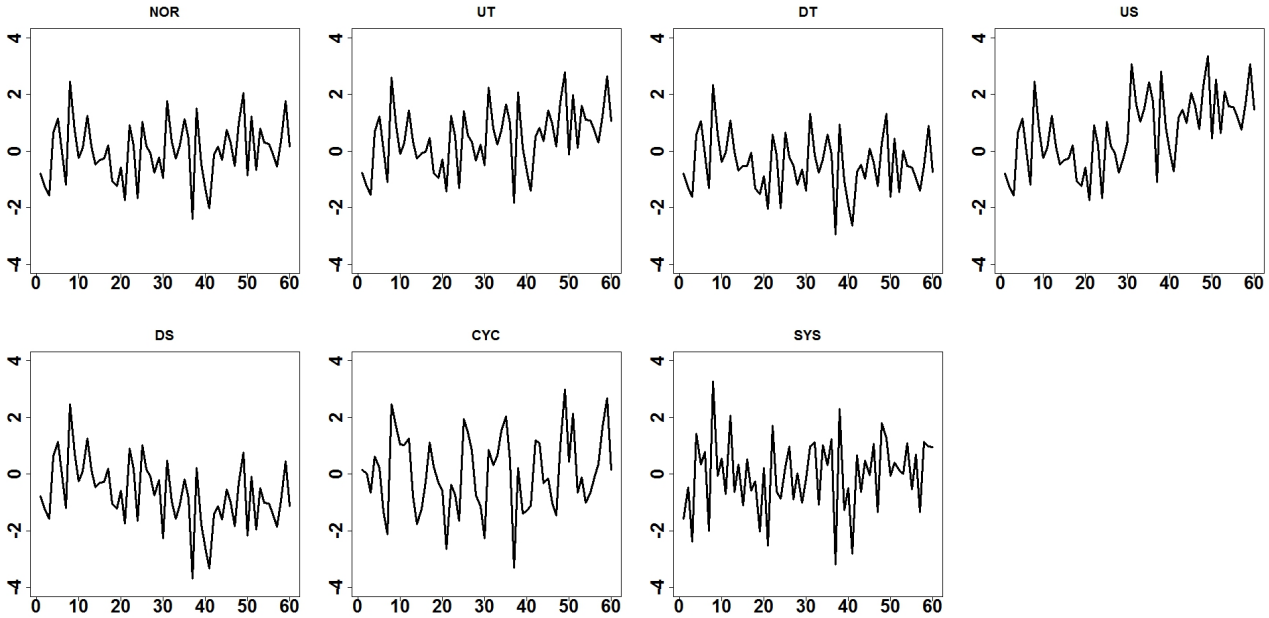


Figure 3: Seven illustrative patterns

$$y_t = -0.23 + 0.005t - 1.23d - 0.41sin\left(\frac{2\pi t}{13}\right) - 0.02(-1)^t + \epsilon_t \tag{4}$$

$$y_t = 0.11 - 0.03t - 0.36sin\left(\frac{2\pi t}{13}\right) - 0.03(-1)^t + \epsilon_t \tag{5}$$

For the case of the DS, it is worth to show the PGS step by step. Firstly, a full and a reduced model were fitted, obtaining the models shown in equation 4 and 5. Nesting these two models, the following F-test is obtained: $F_{54}^{1} = (61.183 - 55.652)\ \ (55.652/54) = 5.3661$

As the p-value associated with the F-value was less than the established significance level ($\alpha$=0.01), it was determined that the reduced model fitted the pattern better; therefore, equation 5 was used for categorising the pattern. In this equation, the only statistically significant parameter was the one related to the Trend pattern. Therefore, this pattern was categorised as DS.

## 4. RESULTS

### *4.1. Pattern generation*

To measure the performance of the proposed PGS at different significance levels, 10,000 patterns of each type were initially generated, being sampled at 60 equal time intervals $t_1, t_2, \ldots, t_{60}$. As mentioned previously, three different significance levels were used, $\alpha$= 0.01, 0.02, 0.03. Table 4 shows the allocation of these 70,000 patterns. They fall into three categories: retained in initial class, reclassified or discarded. It is was found that when the significance level used was $\alpha$= 0.01, 65.38% of the patterns remained in the classes that were initially generated, 14.71% were reclassified and 19.61% were discarded due to two or more patterns produced by the PGS not being related to any of the classes in a statistically significant way. It is worth noting that 19.23% of the patterns initially generated as Normal were classified as Cyclic. This could be due to a possible periodic behaviour of the RNG. It can also be observed that around 15% of the patterns initially generated as Shift patterns were reclassified as Normal. This could be due to the small amounts of shift in these patterns.

Table 3: Analysis of the seven patterns as illustrative examples

| Initial pattern class | Parameter | Initial magnitude | Best model fitted (p-value of F-test) | Estimated parameter (p-value) | Final patter class |
|---|---|---|---|---|---|
| NOR | - | - | Reduced (0.0477) | - | NOR |
| UT | Slope | $0.015\sigma$ | Reduced (0.0467) | 0.0219 (0.0056) | UT |
| DT | Slope | $-0.015\sigma$ | Reduced (0.0407) | -0.0080 (0.0296) | NOR |
| US | Shift magnitude ($\tau$) | $1.3\sigma$ ($\tau$=30) | Full (0.0098, $\tau = 29$) | 1.4098 (0.0098) | US |
| DS | Shift magnitude ($\tau$) | $1.3\sigma$ ($\tau$=30) | Reduced (0.0243) | -0.0253 (0.0024) | DT |
| CYC | Amplitude (frequency) | $1.3\sigma$ (8) | Reduced (0.0975) | 1.2421 (0.0000) | CYC |
| SYS | Systematic departure | $0.8\sigma$ | Reduced (0.0940) | 0.7830 (0.0000) | SYS |

For a significance level of $\alpha$=0.02, 54.26% of the patterns retained their initial classes, 13.41% were reclassified, and the remaining 32.33% were discarded. When the significance level was chosen to be $\alpha$=0.03, 42.17% of the patterns were discarded, 12.40% were reclassified, and the other 45.43% remained in the same pattern class as initially generated. Using this significance level, it was found that 28.69% of the patterns initially created as Normal were reclassified. Reclassification rates for Normal patterns were higher at the three significance levels. It was also noted that the pattern with the lowest discarding percentage was the Normal pattern. As for the Shift patterns, it was found that the reclassification percentage decreased as the significance level increased.

Table 4: Pattern classification using the proposed PGS

| | **Initial class** | | | | | | |
|---|---|---|---|---|---|---|---|
| $\alpha$ **level \ Final class** | **NOR (%)** | **UT (%)** | **DT (%)** | **US (%)** | **DS (%)** | **CYC (%)** | **SYS (%)** |
| $\alpha$**=0.01** | | | | | | | |
| NOR | 65.30 | 1.47 | 1.49 | 15.19 | 16.06 | 2.11 | 5.49 |
| UT | 0.79 | 72.32 | 0.00 | 8.73 | 0.03 | 0.04 | 0.01 |
| DT | 0.68 | 0.00 | 71.99 | 0.02 | 9.02 | 0.04 | 0.07 |
| US | 2.44 | 0.70 | 0.00 | 49.28 | 0.14 | 0.06 | 0.22 |
| DS | 2.15 | 0.00 | 0.77 | 0.12 | 49.39 | 0.04 | 0.20 |
| CYC | 19.23 | 1.06 | 0.99 | 5.45 | 5.44 | 80.29 | 1.65 |
| SYS | 0.72 | 0.02 | 0.00 | 0.17 | 0.18 | 0.01 | 71.22 |
| Discarded | 8.69 | 24.43 | 24.76 | 21.04 | 19.74 | 17.41 | 21.14 |
| $\alpha$ **= 0.02** | | | | | | | |
| NOR | 48.55 | 0.88 | 0.93 | 9.29 | 9.88 | 1.25 | 3.43 |

*Continued on next page*

Table 4 – *Continued from previous page*

| | **Initial class** | | | | | | |
|---|---|---|---|---|---|---|---|
| **$\alpha$ level \ Final class** | **NOR (%)** | **UT (%)** | **DT (%)** | **US (%)** | **DS (%)** | **CYC (%)** | **SYS (%)** |
| UT | 0.84 | 57.65 | 0.00 | 6.11 | 0.04 | 0.03 | 0.07 |
| DT | 0.83 | 0.00 | 56.51 | 0.01 | 6.49 | 0.05 | 0.09 |
| US | 3.04 | 0.80 | 0.00 | 44.43 | 0.12 | 0.08 | 0.20 |
| DS | 2.94 | 0.00 | 0.96 | 0.17 | 44.71 | 0.05 | 0.33 |
| CYC | 25.90 | 1.26 | 1.08 | 6.81 | 6.41 | 69.66 | 1.85 |
| SYS | 1.14 | 0.02 | 0.00 | 0.23 | 0.22 | 0.00 | 58.34 |
| Discarded | 16.76 | 39.39 | 40.52 | 32.95 | 32.13 | 28.88 | 35.69 |
| **$\alpha$ = 0.03** | | | | | | | |
| NOR | 37.31 | 0.57 | 0.76 | 6.63 | 6.78 | 0.71 | 2.46 |
| UT | 0.88 | 46.69 | 0.00 | 4.71 | 0.02 | 0.03 | 0.06 |
| DT | 0.86 | 0.00 | 45.20 | 0.03 | 4.95 | 0.06 | 0.09 |
| US | 3.40 | 0.90 | 0.00 | 39.61 | 0.13 | 0.08 | 0.13 |
| DS | 3.31 | 0.00 | 0.97 | 0.18 | 40.19 | 0.05 | 0.25 |
| CYC | 28.69 | 1.28 | 1.13 | 6.59 | 6.38 | 60.67 | 1.91 |
| SYS | 1.39 | 0.02 | 0.00 | 0.18 | 0.22 | 0.00 | 48.37 |
| Discarded | 24.16 | 50.54 | 51.94 | 42.07 | 41.33 | 38.40 | 46.73 |

### *4.2. Training*

For the SVM training, four sets of 2800 patterns were generated, 400 for each pattern class; one of these sets was created not using the proposed PGS and the other three generated using the proposed PGS, setting the significance level to $\alpha$=0.01, 0.02 and 0.03. In order to deal with nonlinear decision boundaries, five different kernels implemented in the kernlab [54] library of R-software [55] were tested. These were the Bessel, Laplace, Polynomial, Radial basis function and Hyperbolic tangent kernels.

Five different sample sizes were considered for training PNNs. These were n = 60, 80, 100, 120 and 140 patterns of each type. Using these five sample sizes, four data sets were generated, one with conventionally produced patterns, and the other three sets using the PGS with the three aforementioned significance levels ($\alpha$=0.01, 0.02 and 0.03).

To reduce the dimension of the input vectors for training the SVMs and PNNs, the shape features the shape features initially proposed by [56] and then improved by [14] were adopted since they are independent of the scale and length of the data, reduce the training time significantly and increase the pattern recognition accuracy. The number of window segments was set to four.

The function nls implemented in R-software [55], was used to fit NLMs according to the proposed methodology.

A five-fold cross validation and hold-one-out validation for the SVM and the PNN, respectively, were employed for model validation, and the misclassification rate under these schemes was used as the cost function

to be minimised during the training. The Bees Algorithm (BA), proposed by [57] was implemented to find the best sets of free parameters of the SVM and the PNN, the aforementioned cost function being the objective function value to be minimised. This algorithm was selected for its proven ability to determine globally optimal solutions to complex optimisation problems [58]. Table 5 shows the values adopted for the Bees Algorithm parameters. For a definition of those parameters, see [59].

Table 5: Parameter values used in the Bees Algorithm

| **Parameter** | **Symbol** | **Value in PNN** | **Value in SVM** |
|---|---|---|---|
| Initial population | $n$ | 30 | 30 |
| Number of best sites | $m$ | 5 | 4 |
| Number of elite sites | $e$ | 3 | 2 |
| Patch size (Smoothing parameter $\sigma$ in PNN) | *ngh* | 0.01 | - |
| Patch size (Cost parameter C in SVM) | *ngh-c* | - | 0.5 |
| Patch size (Kernel parameters in SVM) | *ngh-k* | - | 0.02 |
| Number of elite bees for the elite sites | *ne* | 4 | 4 |
| Number of bees for the remaining best points | *nb* | 2 | 2 |

### *4.3. Testing*

The four SVMs trained in Section 4.2 were put through 100 test runs. A test run consisted of applying 100 patterns of each type to the trained SVM. The pattern recognition accuracies obtained in the one hundred runs were compared against those achieved with a SVM trained using conventionally generated patterns. Figure 4 shows the 95% CI for the mean accuracy of each kernel tested.

Table 6 shows the accuracies achieved by the four designs (No-PGS and the three levels of the proposed PGS) disaggregated by pattern type. ANOVA of three factors with single, double and triple interactions was used to analyse the results. The three factors were: kernel type, four pattern generators and pattern type. It was observed from this ANOVA that using the proposed PGS significantly increased the mean accuracy by 6.90%. Also, the kernel that achieved the best accuracy was the Laplace kernel.

Table 6: Results for different types using five different kernels in the SVM for three different values of $\alpha$

| **Kernel \ Pattern** | **Without the PGS (%)** | **With PGS, $\alpha = 0.01$ (%)** | **With PGS, $\alpha = 0.02$ (%)** | **With PGS, $\alpha = 0.03$ (%)** |
|---|---|---|---|---|
| **Total testing** | **85.66** | **92.79** | **92.40** | **92.49** |

*Continued on next page*

Table 6 – *Continued from previous page*

| Kernel \ Pattern | Without the PGS (%) | With PGS, $\alpha = 0.01$ (%) | With PGS, $\alpha = 0.02$ (%) | With PGS, $\alpha = 0.03$ (%) |
|---|---|---|---|---|
| **Bessel** | | | | |
| **Total** | **85.65** | **92.51** | **92.23** | **92.33** |
| NOR | 85.42 | 86.54 | 86.49 | 87.02 |
| UT | 82.51 | 93.53 | 93.32 | 93.81 |
| DT | 89.80 | 94.46 | 94.10 | 94.33 |
| US | 83.39 | 92.57 | 92.82 | 92.35 |
| DS | 85.57 | 92.28 | 90.93 | 91.05 |
| CYC | 82.99 | 93.43 | 92.80 | 92.61 |
| SYS | 89.85 | 94.76 | 95.15 | 95.12 |
| **Laplace** | | | | |
| **Total** | **85.38** | **92.92** | **92.55** | **92.82** |
| NOR | 84.72 | 86.43 | 86.81 | 86.06 |
| UT | 82.31 | 94.12 | 93.94 | 94.65 |
| DT | 88.97 | 95.26 | 95.23 | 94.92 |
| US | 85.68 | 92.54 | 92.67 | 93.77 |
| DS | 85.39 | 92.33 | 91.19 | 90.91 |
| CYS | 81.35 | 94.56 | 93.22 | 93.77 |
| SYS | 89.26 | 95.22 | 94.81 | 95.64 |
| **Polynomial** | | | | |
| **Total** | **85.77** | **92.79** | **92.37** | **92.48** |
| NOR | 85.29 | 86.43 | 86.50 | 86.19 |
| UT | 82.57 | 93.91 | 93.07 | 93.78 |
| DT | 89.76 | 95.27 | 94.28 | 94.81 |
| US | 85.74 | 92.35 | 93.63 | 93.27 |
| DS | 85.23 | 91.75 | 90.71 | 90.48 |
| CYC | 81.85 | 94.17 | 93.20 | 93.78 |
| SYS | 89.94 | 95.67 | 95.23 | 95.04 |
| **Radial Basis Function** | | | | |
| **Total** | **85.80** | **92.87** | **92.38** | **92.39** |
| NOR | 85.46 | 87.43 | 87.09 | 85.90 |
| UT | 82.53 | 93.31 | 92.71 | 93.92 |
| DT | 89.74 | 95.44 | 95.01 | 95.08 |
| US | 85.50 | 91.50 | 92.64 | 93.03 |
| DS | 85.56 | 92.74 | 90.87 | 90.14 |
| CYC | 82.08 | 94.01 | 92.65 | 92.60 |

*Continued on next page*

Table 6 – *Continued from previous page*

| **Kernel \ Pattern** | **Without the PGS (%)** | **With PGS, $\alpha = 0.01$ (%)** | **With PGS, $\alpha = 0.02$ (%)** | **With PGS, $\alpha = 0.03$ (%)** |
|---|---|---|---|---|
| SYS | 89.70 | 95.67 | 95.66 | 96.05 |
| **Hyperbolic Tangent** | | | | |
| **Total** | **85.72** | **92.86** | **92.46** | **92.45** |
| NOR | 85.42 | 86.78 | 86.44 | 85.92 |
| UT | 82.35 | 94.18 | 93.35 | 93.92 |
| DT | 89.96 | 94.74 | 94.79 | 94.75 |
| US | 85.42 | 92.62 | 93.47 | 93.64 |
| DS | 85.14 | 92.54 | 90.73 | 90.46 |
| CYC | 81.80 | 94.04 | 93.00 | 93.30 |
| SYS | 89.98 | 95.14 | 95.41 | 95.16 |

For the PNN, one hundred test runs were carried out, each involving 50 patterns of each type. Figure 5 shows the 95% CI for the mean accuracy obtained using the proposed PGS as well as the results for a PNN trained conventionally.

Table 7 shows the mean accuracies achieved during the testing of the PNNs disaggregated by each of the four designs, the seven simple patterns and the five sample sizes. ANOVA of three factors with up to triple interactions was also employed to analyse the accuracies achieved by the PNNs. The three factors considered were: sample size, pattern generation design and pattern type. It was found that the mean accuracy was significantly increased when the PGS scheme was employed.

Table 7: Results for different pattern types using five different sample sizes in the PNN

| **Sample size \ Pattern** | **Without the PGS (%)** | **With PGS, $\alpha = 0.01$ (%)** | **With PGS, $\alpha = 0.02$ (%)** | **With PGS, $\alpha = 0.03$ (%)** |
|---|---|---|---|---|
| **Total testing** | **78.42** | **87.51** | **86.77** | **86.25** |
| **n=60** | | | | |
| **Total** | **77.66** | **86.81** | **84.92** | **85.53** |
| NOR | 62.48 | 74.34 | 68.34 | 72.92 |
| UT | 79.18 | 85.06 | 86.40 | 84.90 |
| DT | 82.90 | 88.64 | 89.88 | 87.44 |
| US | 75.74 | 91.42 | 83.94 | 93.50 |
| DS | 82.38 | 90.44 | 91.50 | 91.34 |
| CYC | 76.18 | 87.14 | 87.30 | 86.24 |
| SYS | 84.78 | 90.64 | 87.08 | 82.40 |

*Continued on next page*

Table 7 – *Continued from previous page*

| **Sample size \ Pattern** | **Without the PGS (%)** | **With PGS, $\alpha = 0.01$ (%)** | **With PGS, $\alpha = 0.02$ (%)** | **With PGS, $\alpha = 0.03$ (%)** |
|---|---|---|---|---|
| **n=80** | | | | |
| **Total** | **77.95** | **87.15** | **86.27** | **85.68** |
| NOR | 64.84 | 71.76 | 73.82 | 75.14 |
| UT | 73.18 | 87.30 | 91.22 | 88.72 |
| DT | 86.90 | 87.88 | 84.94 | 88.48 |
| US | 71.74 | 94.24 | 90.84 | 86.30 |
| DS | 83.66 | 88.38 | 89.24 | 89.66 |
| CYC | 77.38 | 90.04 | 88.86 | 85.26 |
| SYS | 87.94 | 90.46 | 84.94 | 86.22 |
| **n=100** | | | | |
| **Total** | **78.68** | **87.51** | **87.24** | **85.66** |
| NOR | 68.76 | 77.44 | 73.54 | 73.82 |
| UT | 78.40 | 88.54 | 85.92 | 84.30 |
| DT | 83.68 | 86.44 | 91.36 | 88.18 |
| US | 71.94 | 91.24 | 89.08 | 83.82 |
| DS | 83.54 | 88.72 | 90.68 | 90.94 |
| CYC | 79.08 | 90.76 | 90.92 | 87.94 |
| SYS | 85.38 | 89.46 | 89.20 | 90.64 |
| **n=120** | | | | |
| **Total** | **78.91** | **87.95** | **87.88** | **86.80** |
| NOR | 71.72 | 78.24 | 77.46 | 77.14 |
| UT | 79.54 | 87.40 | 89.38 | 86.42 |
| DT | 80.96 | 89.80 | 89.72 | 89.76 |
| US | 73.72 | 90.00 | 89.56 | 88.28 |
| DS | 83.56 | 88.22 | 90.68 | 92.80 |
| CYC | 77.86 | 92.42 | 88.36 | 84.48 |
| SYS | 85.00 | 89.58 | 90.02 | 88.70 |
| **n=140** | | | | |
| **Total** | **78.91** | **88.11** | **87.53** | **87.60** |
| NOR | 73.76 | 77.56 | 77.30 | 77.74 |
| UT | 76.74 | 88.16 | 87.56 | 86.88 |
| DT | 85.90 | 89.42 | 89.12 | 88.40 |
| US | 69.66 | 91.76 | 89.04 | 90.34 |
| DS | 84.10 | 90.04 | 90.42 | 90.82 |
| CYC | 75.52 | 92.06 | 87.46 | 88.08 |

*Continued on next page*

Table 7 – *Continued from previous page*

| **Sample size \ Pattern** | **Without the PGS (%)** | **With PGS, $\alpha = 0.01$ (%)** | **With PGS, $\alpha = 0.02$ (%)** | **With PGS, $\alpha = 0.03$ (%)** |
|---|---|---|---|---|
| SYS | 86.72 | 87.78 | 91.78 | 90.92 |

As mentioned in the Introduction, it was also of interest in this study to assess the proposed scheme with two very different ML algorithms, namely, SVM and PNN. The performance of the proposed PGS was measured for the three aforementioned $\alpha$ levels. Figure 6 shows the mean and the 95% CI of the accuracies achieved. It can be seen that with both ML algorithms the accuracy was increased when the proposed PGS was used. These accuracies marginally increased when the $\alpha$ level was changed to 0.01 for both algorithms.

## 5. CONCLUSION AND FUTURE WORK

The literature review identified that there were no standard methods for generating control chart patterns. In the proposed PGS, all the pattern parameters are randomly assigned to the patterns, i. e., even parameters such as break point position and cycle period were randomised. Also in the literature review, it was found that these two parameters had been ignored by all the authors, despite that fact they are of interest in root cause analysis.

Another issue studied in this paper was the determination of the minima and maxima of pattern parameters during pattern generation. Finding an objective method to set the range of parameter values was an aim of this research. This issue was addressed by nesting two NLMs, with the p-value of the related parameter determining the pattern class. Without the objective method proposed here, by using different parameter ranges during pattern generation, different decision boundaries are estimated. This makes the recognition accuracies achieved not comparable and is a common mistake found in the CCPR literature.

To design the proposed pattern generation scheme, it was necessary to develop a robust procedure for identifying and categorising break points in the mean value in control charts. Such a method not only detects the potential existence of sudden changes in the mean but also statistically estimates the magnitude of these changes. The proposed scheme is also able to handle noise as the estimation of the p-values employed during pattern categorisation is based on the ratio of the estimated parameters and their standard errors.

The performance of the proposed PGS during the initial generation of patterns was measured. It was found that, as the significance level was increased, the percentage of discarded patterns also increased. The percentage of reclassified patterns remained approximately constant for the three values tested. Thus, the significance level mainly affects the number of patterns to be discarded and it is necessary to generate more patterns initially as the significance level is increased.

During the analysis of the performance of the proposed PGS at three different significance levels, two alternative pattern recognition systems were presented: SVM and PNN. In the case of SVM with the proposed PGS, the pattern recognition accuracy was significantly increased by 7.13%, 6.74% and 6.83% when the significance level was $\alpha = 0.01$, 0.02 and 0.03, respectively. In the case of PNN, the mean accuracy was increased by 9.09%, 8.35% and 7.83% when the significance level was $\alpha = 0.01$, 0.02 and 0.03, respectively. Furthermore, it was observed in both recognition systems that increases of 0.01 in the significance level did not significantly affect the

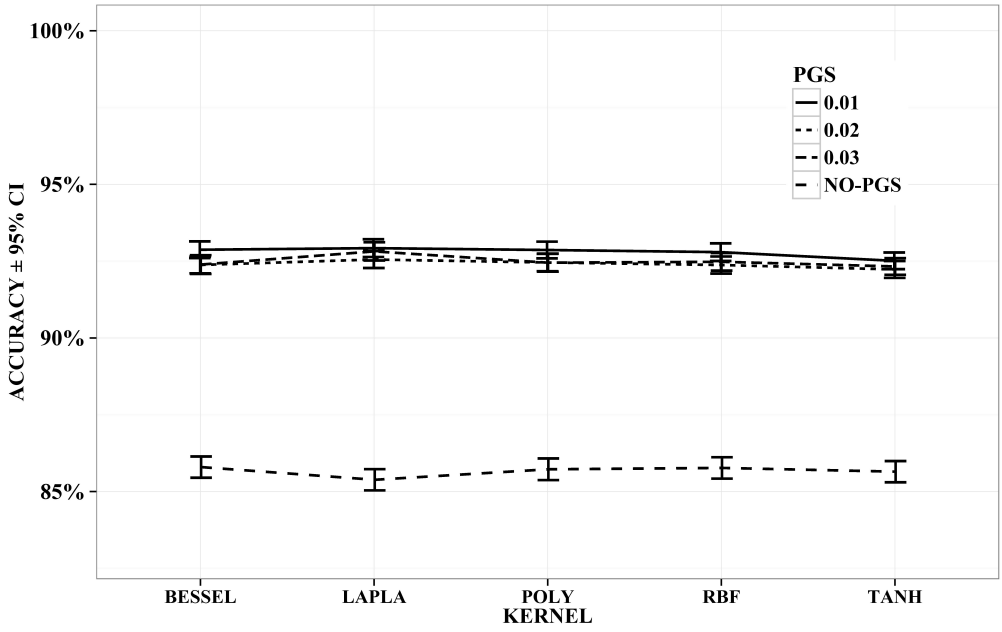


Figure 4: Accuracies achieved with five different kernels and three $\alpha$ levels

pattern recognition accuracy. However, when the increment was from $\alpha$=0.01 to $\alpha$=0.03, a significant decrease in the mean accuracy was observed.

As previously noted, the significance level set during pattern generation mainly affected the number of patterns discarded, and had a small effect on the pattern recognition accuracy of the two tested ML algorithms. It is recommended to use low significance levels such as 0.01 in order to reduce computational efforts and generate fewer patterns.

As mere identification of patterns is sometimes not enough for efficient root cause analysis, further information related to the identified pattern needs to be extracted. Since the CCPR system was trained using patterns that ensured the estimation of correct decision boundaries, generality of the model and statistical significance of the model parameters, details such as cycle amplitude, periodicity, slope, shift magnitude, change point position and systematic departure can be obtained by fitting a NLM to the control chart data as implemented in the proposed PGS.

Future work will be focused on two main paths: developing pattern generation schemes for processes where the inherent noise is modelled by time series; and developing a pattern recognition/prediction scheme for patterns that are slow or weak in a specific inspection window, thus enabling the appearance of a pattern to be anticipated and pre-emptive actions, such as predictive maintenance or repairs, to be taken.

## 6. ACKNOWLEDGEMENTS

The first author acknowledges the support of the Mexican National Council for Science and Technology, CONACYT, enabling him to pursue graduate studies at University of Birmingham − United Kingdom under Grant No. 548421.

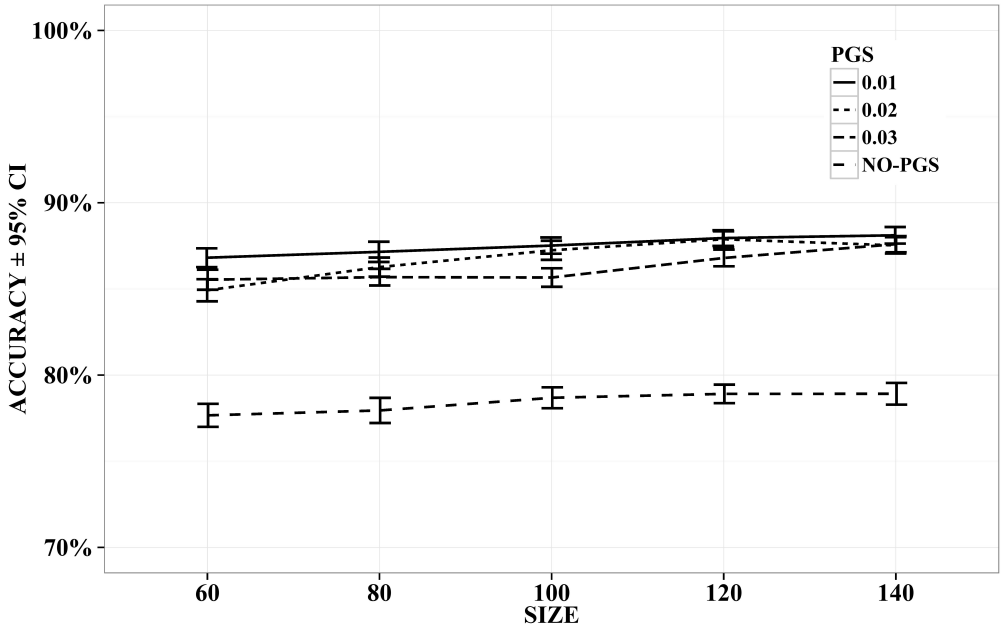


Figure 5: PNN trained with and without patterns created using the proposed pattern generation scheme

## 7. APPENDIX

The following mathematical expressions were used initially to generate the patterns to be detected:

- Normal Pattern (NOR):

$$y_t = \mu + N_t \tag{A1}$$

where $y_i$ represents a quality characteristic sampled at time $i$, $\mu$ is the mean value of the process, fixed to zero, and $N_t$ is a normally distributed variable with mean equal to zero and variance $\sigma_N$ equal to one, and represents the inherent noise in the process.

- Upward/Downward Trend (UT/DT):

$$y_t = \mu + N_t + \beta_1 t \tag{A2}$$

where $\beta_1$ is the Trend slope.

- Upward/Downward Shift (US/DS):

$$y_t = \mu + N_t + \beta_2 d \tag{A3}$$

where $\beta_2$ is the shift magnitude in the mean; $d = 1$ after the shift, and $d = 0$ before the shift, being the break point when the shift occurred randomly chosen between $\tau = 16$ and $\tau = n - 15$.

- Cyclic (CYC):

$$y_t = \mu + N_t + \beta_3 \left( \frac{2\pi t}{\beta_4} \right) \tag{A4}$$

where $\beta_3$ and $\beta_4$ are the amplitude and frequency of the cyclic pattern respectively.

- Systematic (SYS):

$$y_t = \mu + N_t + \beta_5 (-1)^t \tag{A5}$$

where $\beta_5$ represent the systematic departure.

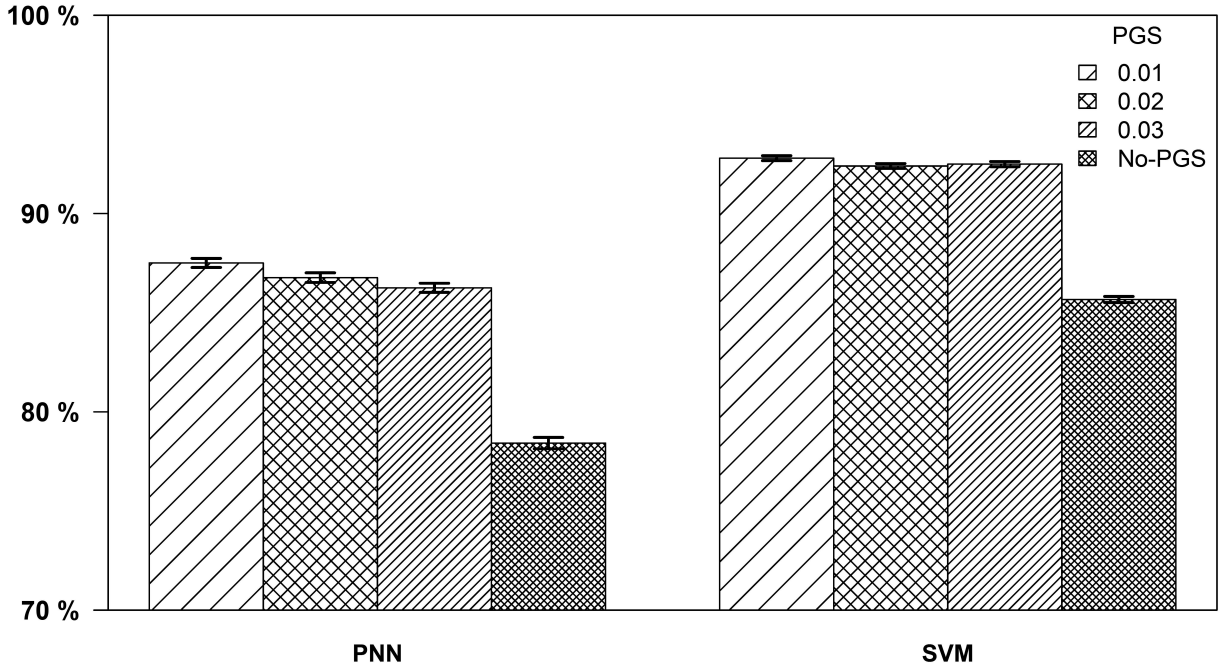


Figure 6: Accuracies achieved by PNN and SVM with different PGSs

## REFERENCES

[1] C. Western Electric, Statistical quality control handbook, Western Electric Co., Indianapolis, 1956.

[2] W. Hachicha, A. Ghorbel, A survey of control-chart pattern-recognition literature (19912010) based on a new conceptual classification scheme, Computers & Industrial Engineering 63 (1) (2012) 204–222.

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

[4] S. R. Gunn, Support vector machines for classification and regression, ISIS technical report 14.

[5] S. Du, D. Huang, J. Lv, Recognition of concurrent control chart patterns using wavelet transform decomposition and multiclass support vector machines, Computers & Industrial Engineering 66 (4) (2013) 683–695.

[6] M. Matsumoto, T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator, ACM Transactions on Modeling and Computer Simulation (TOMACS) 8 (1) (1998) 3–30.

[7] P. Xanthopoulos, T. Razzaghi, A weighted support vector machine method for control chart pattern recognition, Computers & Industrial Engineering 70 (2014) 134–149.

[8] L. Xie, N. Gu, D. Li, Z. Cao, M. Tan, S. Nahavandi, Concurrent control chart patterns recognition with singular spectrum analysis and support vector machine, Computers & Industrial Engineering 64 (1) (2013) 280–289.

[9] N. Gu, Z. Cao, L. Xie, D. Creighton, M. Tan, S. Nahavandi, Identification of concurrent control chart patterns with singular spectrum analysis and learning vector quantization, Journal of intelligent manufacturing 24 (6) (2013) 1241–1252.

[10] S. K. Gauri, Improved feature-based test statistic for assessing suitability of the preliminary samples for constructing control limits of x-bar chart, The International Journal of Advanced Manufacturing Technology 58 (9-12) (2012) 1171–1187.

[11] C.-J. Lu, Y. E. Shao, P.-H. Li, Mixture control chart patterns recognition using independent component analysis and support vector machine, Neurocomputing 74 (11) (2011) 1908–1914.

[12] A. Hassan, An improved scheme for online recognition of control chart patterns, International Journal of Computer Aided Engineering and Technology 3 (3) (2011) 309–321.

[13] V. Ranaee, A. Ebrahimzadeh, R. Ghaderi, Application of the psosvm model for recognition of control chart patterns, ISA transactions 49 (4) (2010) 577–586.

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

[15] P. Jiang, D. Liu, Z. Zeng, Recognizing control chart patterns with neural network and numerical fitting, Journal of Intelligent Manufacturing 20 (6) (2009) 625–635.

[16] C.-H. Wang, W. Kuo, Identification of control chart patterns using wavelet filtering and robust fuzzy clustering, Journal of Intelligent Manufacturing 18 (3) (2007) 343–350.

[17] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, Application of the bees algorithm to the training of radial basis function networks for control chart pattern recognition, in: Proceedings of 5th CIRP international seminar on intelligent computation in manufacturing engineering (CIRP ICME06), Ischia, Italy, 2006, pp. 711–716.

[18] M. A. Barghash, N. S. Santarisi, Pattern recognition of control charts using artificial neural networksanalyzing the effect of the training parameters, Journal of Intelligent Manufacturing 15 (5) (2004) 635–644.

[19] M. Pacella, Q. Semeraro, A. Anglani, Manufacturing quality control by means of a fuzzy art network trained on natural process data, Engineering Applications of Artificial Intelligence 17 (1) (2004) 83–96.

[20] R.-S. Guh, Optimizing feedforward neural networks for control chart pattern recognition through genetic algorithms, International Journal of Pattern Recognition and Artificial Intelligence 18 (02) (2004) 75–99.

[21] J.-G.-S. De Tejada, J. S. Martnez-Echevarra, Support Vector Machines, Springer US, 2007, book section 7, pp. 147–191. doi:10.1007/0-387-37452-3_7.

[22] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[23] C. J. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 121–167.

[24] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, Neural Networks, IEEE Transactions on 13 (2) (2002) 415–425.

[25] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, The Journal of Machine Learning Research 2 (2002) 45–66.

[26] M. Pontil, A. Verri, Support vector machines for 3d object recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 20 (6) (1998) 637–646.

[27] O. Chapelle, P. Haffner, V. N. Vapnik, Support vector machines for histogram-based image classification, Neural Networks, IEEE Transactions on 10 (5) (1999) 1055–1064.

[28] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, D. Haussler, Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16 (10) (2000) 906–914.

[29] S. Hua, Z. Sun, Support vector machine approach for protein subcellular localization prediction, Bioinformatics 17 (8) (2001) 721–728.

[30] R. B. Chinnam, Support vector machines for recognizing shifts in correlated and other manufacturing processes, International Journal of Production Research 40 (17) (2002) 4449–4466.

[31] D. F. Specht, Probabilistic neural networks, Neural networks 3 (1) (1990) 109–118.

[32] D. F. Specht, Enhancements to probabilistic neural networks, in: Neural Networks, 1992. IJCNN., International Joint Conference on, Vol. 1, IEEE, 1992, pp. 761–768.

[33] K. Z. Mao, K.-C. Tan, W. Ser, Probabilistic neural-network structure determination for pattern classification, Neural Networks, IEEE Transactions on 11 (4) (2000) 1009–1016.

[34] C. Kramer, B. Mckay, J. Belina, Probabilistic neural network array architecture for ecg classification, in: Engineering in Medicine and Biology Society, 1995., IEEE 17th Annual Conference, Vol. 1, IEEE, 1995, pp. 807–808.

[35] M. T. Musavi, K. H. Chan, D. M. Hummels, K. Kalantri, On the generalization ability of neural network classifiers, Pattern Analysis and Machine Intelligence, IEEE Transactions on 16 (6) (1994) 659–663.

[36] R. D. Romero, D. S. Touretzky, R. H. Thibadeau, Optical chinese character recognition using probabilistic neural networks, Pattern Recognition 30 (8) (1997) 1279–1292.

[37] Y. Sun, M.-H. Horng, X. Lin, J.-Y. Wang, Ultrasonic image analysis for liver diagnosis, Engineering in Medicine and Biology Magazine, IEEE 15 (6) (1996) 93–101.

[38] J.-J. Quan, X.-B. Wen, X.-Q. Xu, Multiscale probabilistic neural network method for sar image segmentation, Applied Mathematics and Computation 205 (2) (2008) 578–583.

[39] T. Song, M. M. Jamshidi, R. R. Lee, M. Huang, A modified probabilistic neural network for partial volume segmentation in brain mr image, Neural Networks, IEEE Transactions on 18 (5) (2007) 1424–1432.

[40] D. Gerbec, S. Gasperic, I. Smon, F. Gubina, Allocation of the load profiles to consumers using probabilistic neural networks, Power Systems, IEEE Transactions on 20 (2) (2005) 548–555.

[41] D. Kim, D. H. Kim, S. Chang, Application of probabilistic neural network to design breakwater armor blocks, Ocean Engineering 35 (3) (2008) 294–300.

[42] A. Pande, M. AbdelAty, A computing approach using probabilistic neural networks for instantaneous appraisal of rearend crash risk, ComputerAided Civil and Infrastructure Engineering 23 (7) (2008) 549–559.

[43] E. D. beyli, Lyapunov exponents/probabilistic neural networks for analysis of eeg signals, Expert Systems with Applications 37 (2) (2010) 985–992.

[44] S. Wu, Abnormal pattern parameters estimation of control chart based on wavelet transform and probabilistic neural network, Springer, 2006, pp. 112–119.

[45] S. Lesany, A. Koochakzadeh, S. Fatemi Ghomi, Recognition and classification of single and concurrent unnatural patterns in control charts via neural networks and fitted line of samples, International Journal of Production Research (ahead-of-print) (2013) 1–16.

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

[47] R. S. Guh, Integrating artificial intelligence into on line statistical process control, Quality and Reliability Engineering International 19 (1) (2003) 1–20.

[48] R.-S. Guh, A hybrid learning-based model for on-line detection and analysis of control chart patterns, Computers & Industrial Engineering 49 (1) (2005) 35–62.

[49] A. Shaban, M. A. Shalaby, A double neural network approach for the identification and parameter estimation of control chart patterns, International Journal of Quality Engineering and Technology 3 (2) (2012) 124–138.

[50] A. D. McQuarrie, C.-L. Tsai, Regression and time series model selection, Vol. 43, World Scientific Singapore, 1998.

[51] N. R. Draper, H. Smith, E. Pownell, Applied regression analysis, Vol. 3, Wiley New York, 1966.

[52] J. O. Rawlings, S. G. Pantula, D. A. Dickey, Applied regression analysis: a research tool, Springer Science & Business Media, 1998.

[53] A. R. Gallant, Nonlinear statistical models, Vol. 310, John Wiley & Sons, 2009.

[54] A. Karatzoglou, A. Smola, K. Hornik, A. Zeileis, kernlab - an s4 package for kernel methods in r (2004). URL `http://www.jstatsoft.org/v11/i09/`

[55] R. C. Team, R: A language and environment for statistical computing. (2014). URL `http://www.R-project.org/`

[56] D. Pham, M. Wani, Feature-based control chart pattern recognition, International Journal of production research 35 (7) (1997) 1875–1890.

[57] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, The bees algorithm-a novel tool for complex optimisation problems, in: Proceedings of the 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006), 2006, pp. 454–459.

[58] D. Pham, M. Castellani, Benchmarking and comparison of nature-inspired population-based continuous optimisation algorithms, Soft Computing (2013) 1–33.

[59] M. Castellani, Evolutionary generation of neural network classifiersan empirical comparison, Neurocomputing 99 (2013) 214–229.