Intelligent data-driven monitoring of high dimensional multistage manufacturing processes

Mohammadhossein Amini*

Industrial and Manufacturing Systems Engineering,
Kansas State University,
Manhattan, Kansas, 66506, USA
Email: mhamini@ksu.edu
*Corresponding author

Shing I. Chang

Industrial and Manufacturing Systems Engineering,
Kansas State University,
Manhattan, Kansas, 66506, USA
Email: changs@ksu.edu

Abstract: Recent advances in cyber-physical systems and the Internet of things (IoT) have enabled the possible development of smart production systems. However, the complexity of such a system has posed significant challenges for traditional quality engineering methods, especially in monitoring and diagnosis of system performance. The traditional practices for monitoring or controlling multistage systems either treat each stage as an individual entity or model all stages as a whole. The formal approach mainly focuses on the most critical stages while ignores information from the other stages. In contrast, the latter approach attempts to build one model to account for all stages. In a complex production system, this latter approach is impractical, if not impossible. This research provides a control strategy by proposing an intelligent process monitoring system for high dimensional multistage processes using predictive models built from historical data. A repository dataset is used to demonstrate the implementation of the proposed framework.

Keywords: multistage manufacturing systems; data-driven; process monitoring; smart manufacturing; quality engineering; machine learning; predictive modelling; semiconductor manufacturing.

Reference to this paper should be made as follows: Amini, M. and Chang, S.I. (2020) 'Intelligent data-driven monitoring of high dimensional multistage manufacturing processes', *Int. J. Mechatronics and Manufacturing Systems*, Vol. 13, No. 4, pp.299–322.

Biographical notes: Mohammadhossein Amini is a Lecturer at Olin Business School, Washington University in St. Louis. He has received a PhD and a Master's degree in Industrial Engineering from Kansas State University in 2019 and, 2015 respectively.

Shing I. Chang earned his PhD in Industrial and Systems Engineering from the Ohio State University in 1991. He is a professor in the Department of Industrial and Manufacturing Systems Engineering at Kansas State University. He currently teaches courses related to Quality Engineering and Big Data at both undergraduate and graduate levels. His research interests include profile monitoring, statistical process control implementation for big data applications, neural networks and fuzzy set applications in quality engineering, spatial data modelling, and multivariate data modelling and visualisation. He served as Department Editor (2003–2009) for online quality engineering for IIE Transactions. He is a Member of several editorial boards of international journals. He is a Senior Member of IIE and ASQ.

1 Introduction

This research aims to develop a multistage process monitoring system for high dimensional multistage processes using predictive classification models. A multistage system refers to a system where several steps are needed to produce a product or perform a service. Many different industries, such as semiconductor manufacturing, assembly lines, and additive manufacturing, are examples of multistage systems (Shi and Zhou, 2009). In multistage systems, each stage may have multiple characteristics. For instance, in a car assembly line, the body dimension inspection is a critical stage where coordinate measuring machines generate numerous data points (Ceglarek and Shi, 1995). Although all dimensions fit into their tolerance and no control chart indicates out of control. However, door fitting in a later assembly stage may leave significant gaps in some areas. Existing process control methods such as control charts do not pass the dimensions information into later stages due to the sheer data volume or dimensional constraints in the dataset itself.

This kind of general multistage process may constitute a high dimensional vector in which each element contains either the status or measures of a process parameter or quality characteristics at the time of measurement. Note that the timestamps for each measurement may be different in different stages but can be strung together. This high-dimensional vector can be used directly for process monitoring or diagnosis during production and post-production. While manufacturing processes have seen much improvement, process monitoring techniques such as control charting have not experienced transformative improvement since Shewhart (1930) introduced X-bar and R charts in the 1920s. Since then, many studies have been published to improve process monitoring techniques incrementally. Statistical models such as cumulative sum (CUSUM), and exponential weighted moving average (EWMA), and Multivariate techniques such as Hotelling T^2 (Hotelling, 1947), principal component analysis (PCA), and generalised likelihood ratio test (GLRT) (Amini and Chang, 2018) all have improved the field. However, these process monitoring methods fail to answer the challenges posed by future manufacturing environments where abundant sensor data on process parameters and semi-finished parts are readily available.

For example, one of the visions of Industry 4.0 calls for a smart quality management system leveraging the real-time use of process data to monitor product quality (Foidl and Felderer, 2015). Nowadays, data-driven approaches such as machine learning (ML) techniques or loosely called *AI* have been adopted for decision making. But process data has still been used in an isolated manner regarding process monitoring practices. Control charts are implemented only for critical quality characteristics rather than on process parameters, of which data is either thrown away or stored in massive databases. This phenomenon has been dubbed 'dark data' in that most data has never been used for any purpose. Some manufacturers only use process data in a 'fire-fighting' mode when data is dug out for root-cause analysis when there is a decline in product quality. To diagnose what parameters, which stage, and when such a discrepancy took place, process engineers have to examine archived process data, which may take a long time due to various reasons such as messy and unclean data, outdated data, complexity, and dimensionality issues (Amini and Chang, 2018).

Answering these challenges, researchers have provided classification-based process monitoring techniques to use the manufacturing data (Amini and Chang, 2018). However, these methods usually provide quality predictions at the end of the manufacturing process and therefore provide no chance to fix the problem during production. Moreover, data-driven approaches generally face different challenges such as high dimensionality, updating process, and rare faulty samples. Addressing these challenges, we propose a stage-wise process monitoring model that provides prognosis information related to the process parameters following the stage of the current production before the last stage is reached.

This paper contains the following sections. The next section provides a brief literature review of quality engineering and process monitoring studies for high dimensional multistage systems. Section 3 provides a list of challenges faced in applying data-driven approaches in the manufacturing systems. Then the proposed model for process monitoring is introduced. The proposed model is then applied to a repository dataset to show the operation of the proposed model. Finally, future studies and conclusions are presented in the last section.

2 Literature review

Multistage systems are ubiquitous in practice in various industries. However, quality control of such systems is very complicated since the variation of each stage does not solely depend on itself but may come from upstream stages. Figure 1 illustrates a diagram of the multistage system.

Manufacturers have been using traditional control charts to monitor the quality of their products since the 1920s. Shewhart (1930) converted a series of hypothesis testings into a graphic monitoring tool. Traditional statistical process control or monitoring (SPC or SPM) approaches are widely used because of their simplicity and applicability. However, in the area of high-tech manufacturing products, traditional methods of quality control are not effective due to the 'curse of dimensionality' (Friedman et al., 2001). Unlike the traditional methods where measurement is restricted to physical products or work in progress, process parameters offer ample opportunities for process monitoring and defect prevention. Since the number of parameters is usually vast, a high-dimensional

problem often renders traditional control charts ineffective. For example, the production of a CPU includes hundreds of processes and thousands of process parameters. Various techniques such as Hotelling T^2 (Hotelling, 1947), PCA, and GLRT are proposed to study multiple quality characteristics (Amini and Chang, 2018). Hotelling T^2 chart developed in 1947 are used where a $p \times 1$ sample vector with mean μ and covariance matrix Σ are known or can be estimated.

$$X^2 = (x-\mu)' \Sigma^{-1} (x-\mu) = \chi_p^2 \tag{2}$$

A constant c can be determined according to the desired type I and type II error to define the boundaries of the normal process when $X^2 < c$ the process of interest is in control. PCA often used to reduce the dimension of the sample vector and then, a monitoring technique is applied to the reduced dimension. Finally, GLRT is another method to detect changes in multivariate problems. It also can be used to incorporate time information into the change detection models. Suppose X_t as a p-dimensional sample obtained at time unit t for a process. Two following hypotheses are testing the process to detect the change that happens in time τ.

$$H_0 : X_1, X_2, \ldots, X_t \sim f_0(x) \tag{3}$$

$$H_1 : X_1, X_2, \ldots, X_\tau \sim f_0(x), \quad X_{\tau+1}, X_{\tau+2}, \ldots, X_t \sim f_1(x) \tag{4}$$

The likelihood ratio is defined as

$$L = \frac{\prod_{i=1}^{\tau} f_0(x_i) \prod_{i=\tau+1}^{t} f_1(x_i)}{\prod_{i=1}^{t} f_0(x_i)} = \frac{\prod_{i=\tau+1}^{t} f_1(x_i)}{\prod_{i=\tau+1}^{t} f_0(x_i)} \tag{5}$$

where f_0, f_1 are identified as unknown probability densities for the in control (IC) and out of control (OC) process points. To find out the unknown τ, the generalised ratio can be defined to maximise the likelihood ratio in equation (5). The log of the generalised likelihood ratio is

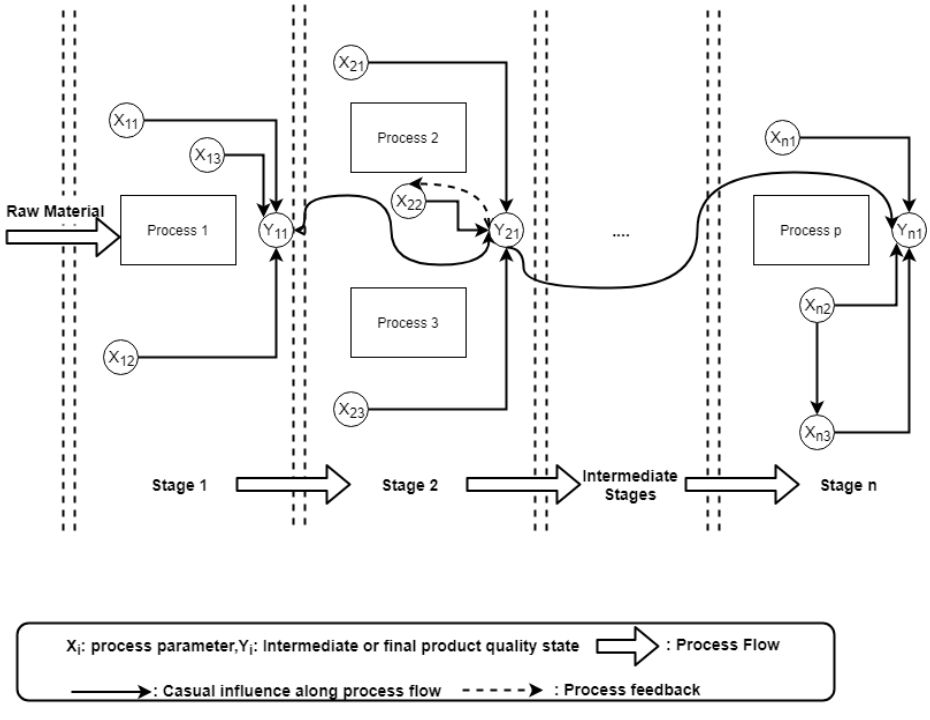
$$L_t = \max_{\tau} \sum_{i=\tau+1}^{t} ln \frac{f_1(x_i)}{f_0(x_i)} \tag{6}$$

Then, the signal is triggered when the decision parameter L_t exceeds a certain limit. This signal indicates that a change has taken place.

Despite the effectiveness of these methods, however, the traditional multivariable methods cannot be effectively applied in complex processes because they were designed to detect mean shifts of a moderate number of quality characteristics, usually less than 10 (Amini and Chang, 2018). Another major SPC innovation, since its inception was to identify the small process changes faster. Univariate control charts for this purpose include CUSUM and EWMA control charts. The key concept is to involve historical observations leading up to the current observation to expedite mean shifts or variance changes. These univariate control charts, like the X-bar and R charts, cannot be implemented effectively in cases where multiple quality characteristics or process parameters exist. In the multivariate environment, multivariate exponentially weighted moving averages (MEWMA) (Lowry et al., 1992) and multivariate cumulative sum

(MCUSUM) (Runger and Testik, 2004) control charts are appropriate. However, these methods either cannot detect off-target (*AKA* out-of-control (OC)) signals fast enough or cause unacceptable false alarm rates as the number of variables increases. Therefore, more efficient models are needed to tackle high dimensional process monitoring problems. Also, most SPC methods do not perform well in multistage applications because data from a multistage process is often considered as a whole without timestamps. Therefore, traditional multivariate SPC methods could not discriminate at which stage that a change takes (Shi and Zhou, 2009). Hence, the need for in-process monitoring for multistage systems can not be greater in the context of SPC.

Figure 1 A diagram of the multistage system (see online version for colours)



Jin and Shi (1999) first consider the modelling of a general multistage system where the critical quality characteristics of the product at stage k is represented by x_k as the following equation:

$$x_k = A_{k-1}x_{k-1} + B_k u_k + w_k \text{ and } y_k = C_k x_k + v_k \tag{7}$$

where $u_k, w_k,$ and v_k represent process error source, unmodelled error, and sensor noise, respectively. $A_{k-1}x_{k-1}$ represents the transformation of product quality deviations from station $k-1$ to station k, $B_k u_k$ represents the product deviations resulting from process errors at stage k and C_k maps the product quality states to quality measurements. The model has been used in many applications such as rigid-part assembly processes, compliant-part assembly processes, machining processes, and sheet stretch forming processes (Shi and Zhou, 2009). However, the physics of the process needs to be thoroughly studied to construct the process model.

Cause-selecting charts are other tools for monitoring the quality of the process in multistage systems. These charts have shown to be effective in finding the responsible stage in a faulty condition. Cause-selecting charts generally use univariate techniques, hence cannot handle high dimensional problems (Jin and Shi, 1999). Another commonly used method in a multistage system called multistream production is the group charts, which can be used to detect quality changes in identical, individual streams. However, this technique cannot perform diagnosis within stages because it only tracks the worst performance in a stage to analyse the process (Pan et al., 2016).

Data-driven approaches emerged as promising framework classes for SPC (Amini and Chang, 2018). These approaches include the use of classification-based models to group historical data into two: IC or OC. After learning from known patterns, trained models can predict the class IC or OC of a new dataset. Several studies have been proposed to monitor the process in the multivariate environment under two main categories (Amini and Chang, 2018). The first category utilised a method called the artificial contrast method (Tuv and Runger, 2003; Hu et al., 2007; Li et al., 2006; Hu and Runger, 2010; Deng et al., 2012; Hwang et al., 2007), which artificially produced OC points to balance the sample set.

Tuv and Runger (2003) first introduced the artificial contrast data to represent the OC points. The artificially generated data were random numbers generated by a uniform distribution. In this study, the range of IC points has been used to generate random numbers. Then, the generation of contrast data has been repeated for each parameter independently. A gradient boosting machine has been used in this study to classify the IC and OC points. Besides, in the case of high dimensionality, the authors recommended the reduction of the number of parameters using by a feature selection classifier.

Hwang et al. (2007) followed the previous work by Tuv and Runger (Tuv and Runger, 2003), where the generation of artificial contrast data was limited to one standard deviation of the target point. Also, the selected classifiers by Hwang et al. (2007) include random forest (RF) (Breiman, 2001) and regularised least square classifier (RLSC) (Poggio and Smale, 2005).

Hu et al. (2007) then introduced the concept of fine-tuning the artificial contrast data by incorporating prior knowledge of the manufacturing process (Tuv and Runger, 2003). While Tuv and Runger (2003) used a uniform distribution to generated artificial contrast data, Hu et al. (2007) instead generated the contrast data using the artificial contrast data in an intentional pre-defined direction. The tuned direction comes from the pre-knowledge of the process. The classifier used by this method is RF. Hu et al. (2007) claimed to obtain more precise results in terms of accuracy and false alarms.

Li et al. (2006) applied the artificial contrast concept in a change point detection problem where a vector of data points considered as features to capture the time when there was any change in the pattern. By using the likelihood ratio function (equation (5)), Hu and Runger (2010) incorporated the time element in the artificial contrast concept. The probability of each class by RF in each time unit is used to represent the functions in the likelihood ratio. Then, the EWMA chart using the obtained likelihood ratio is used to monitor the process.

Using the real-time data and the artificial contrast concept, Deng et al. (2012) introduced the real-time contrast concept to monitor a process. This proposed study used fixed-size new real-time observations to contrast the reference data (training set). By having a new observation window, a new classifier is trained for process monitoring. Like the traditional process monitoring studies, Deng et al. (2012) did not alter the

reference data (points labelled $y = 0$) while the new observations were defined as OC points. Hence, in normal process conditions, the error of the proposed method is expected to be high as both IC and OC points are following the same pattern. Once a shift in the process occurs, the error reduces. To identify the essential features, Deng et al. (2012) used RF as the classifier.

The second category applies feature selection methods (Jiang and Tsui, 2008; Wang and Jiang, 2009; Zou and Qiu, 2009; Jin et al., 2017) to reduce the dimensionality of high-dimensional process monitoring problems. The developed T^2 statistic by Jiang and Tsui (2008) enables the identification of the responsible variables for OC points.

VS-MSPC is a variable selection based multivariate SPC control chart developed by Wang and Jiang (2009). The variable selection in the VS-MSPC is based on a penalised likelihood function. Then, the VS-MSPC method only monitors the selected variables. Wang and Jiang (2009) assumed that the simultaneous shift in multivariate problems usually happens in a limited number of variables. Based on the assumption, monitoring a small fraction of variables is then possible by the multivariate SPC methods. One of the limitations of the proposed method, however, is not being sensitive to small shifts.

Zou and Qiu (2009) used Lasso (Tibshirani, 1996) as the variable selection based model. Then, the MEWMA chart is proposed as the monitoring tool in the reduced problem. Like the previous studies, Zou and Qiu's approach (Zou and Qiu, 2009) comes with strong assumptions such as a limited number of variables to shift and normal and independent observations.

To incorporate cascade information, Jin et al. (2017) extended the previous work by Zou and Qiu (2009). In the cascade process, a process leads to several succeeding processes. Hence, when a shift in the process occurs, besides the root cause parameter, the following process parameters might be considered as responsible elements as well, where this might not always be the case. Jin et al. (2017) incorporated the cascade information using a Bayesian Network. Hence, the proposed method is called Lasso-BN, where BN stands for Bayesian Network. After identifying the truly responsible variables, a T^2 chart is used to monitor the process. The availability of the cascade relationship between parameters is assumed to be available and represented by a BN.

The variable selection models introduced by the researchers have shown promising in simplifying the high dimensional problem into a simplified environment. However, they have a compelling assumption as pre-known distributing of the observations which do not stand always. This assumption is relaxed in the first approach, where the classifier has performed the process monitoring. Where in the second approach, the monitoring techniques are performed by the traditional monitoring techniques such as Hotelling T^2 (Hotelling, 1947), EWMA, and CUSUM.

The studies, as mentioned earlier, generally follow the traditional SPC approaches where only product quality characteristics are considered in-process monitoring and quality assessment. Using data-driven techniques, Wuest et al. (2014) incorporated both product state and process state data for quality monitoring. The proposed study benefits from a wide range of supervised and unsupervised ML techniques. In the multistage production system illustrated by Wuest et al. (2014), changes in the product's physical shape are defined as the checkpoints. Kao et al. (2017) used several classification models to perform quality prediction in a multistage process. Also, Kao et al. (2017) have applied associate rule mining techniques to improve the prediction accuracy of the model. To verify the proposed method, authors have utilised a semiconductor dataset semiconductor manufacturing (SECOM) (Dheeru and Taniskidou, 2017).

Data-driven techniques have also been used by Uhlmann et al. (2017) to perform quality monitoring in a metal 3D printing process called selective laser melting. Several ML models, such as support vector machine, neural networks, Bayesian classifier, and nearest neighbours were applied to perform the monitoring task.

We developed a multi-layer classification process monitoring model (MLCPM) (Amini and Chang, 2018) to metal 3D printing, which is a multistage process considering its layer-by-layer nature of production. We adopt supervised and unsupervised models in MLCPM to control the quality of the print process while the print process has not reached its final layer. MLCPM provides solutions toward the high dimensionality of metal 3D print data using clustering and feature selection methods. However, like the previous studies, that application had no mechanism to update the training set in MLCPM, meaning that MLCPM would not function properly if it faces new, unseen patterns.

Through all these improvements in the process monitoring of multistage systems, there are still ample opportunities and challenges such as problem dimension, new unseen fault behaviours, and unbalanced classes to establish a research plan based on them. This research aims to develop an ML-based model that addresses these challenges, which will be discussed further in more detail in the following section.

3 Challenges

Developing a classification-based process monitoring technique for high dimensional multistage systems has three main challenges: high dimensionality, updating process, and rare OC points, as described in the following paragraphs.

3.1 High dimensionality

Today's production machines are often equipped with multiple sensors generating a large amount of process data at a torrential pace. The widening use of the Internet of things (IoT) has contributed to this trend (Morgan, 2014). ML techniques, a subset of 'artificial intelligence', has also contributed to the possibility of solving problems with high dimensionality since algorithms are used to autonomously learn from data (Marr, 2016). The production machines usually generate a considerable amount of data with a high dimension. As discussed earlier, coordinate measurement machines produce excellent measurements in a high dimension. Dealing with high dimensional data usually asks for more efforts and computations. Besides the computation time, data-driven models with high dimensions tend to overfit. Overfitting is a pervasive problem, especially in predictive models, where they perform very well in the training sets. However, they fail to generalise, meaning that they cannot predict unseen datasets well (Breiman, 2001). Hence, researchers usually avoid overfitting by reducing the dimension of problems by dropping correlated features. Dimension reduction techniques such as PCA, linear discrimination analysis (LDA), and penalised learners such as Lasso (Tibshirani, 1996) and Ridge Regression are examples of the dimension-reduction methods to avoid overfitting.

3.2 Updating process (cover unseen data patterns)

The training phase is a crucial part that makes a data-driven based model more accurate. In traditional SPC practice, the training dataset usually does not change. This practice is often referred to as Phase I SPC, in which data from a processing period deemed in control constitutes a training set. However, most data-driven approaches cannot perform accurately, facing unseen patterns. Hence, the training phase must be updated periodically to enable the model to cover new patterns. Further, new data deemed in the IC state should be included in the training set. In general, ML models benefit from more training samples for better accuracy.

3.3 Rare OC points (unbalance classes)

In traditional manufacturing processes, the reference data consisted of only IC points, and the process was monitored against the reference data to pinpoint faulty spots. However, data-driven approaches such as classification algorithms require both IC and OC observations in the training set. However, a healthy manufacturing process contains occasional OC conditions. Hence, a historical dataset consists of much more IC than OC data. This phenomenon causes an unbalance classification issue, especially in binary classification problems. Two approaches, including undersampling and oversampling, have been proposed in the literature (Chawla et al., 2002) to tackle this issue. In the undersampling method, the number of training data from the majority class is reduced to the level of a minority class. On the other hand, the oversampling method generates more samples from the minority class to those of the majority class. Artificial contrast data is one of the oversampling techniques applied to the process monitoring studies (Tuv and Runger, 2003). Despite the existence of several studies that use the artificial contrast data in process monitoring problems, a comprehensive study exploring several ML approaches facing unbalance samples is lacking. It is not certain whether the undersampling approach is better than the oversampling methods to tackle the unbalanced sampling problems in the content of process monitoring. Hence, both approaches should be investigated.

4 Proposed multistage process monitoring model

This research aims to study a system-wide process monitoring system based on predictive models. The proposed platform intends to monitor the manufacturing process and trigger necessary alarms while any process is heading into a detrimental quality outcome. The proposed framework is capable of process monitoring for high dimension multistage systems. The proposed method can be applied to various multistage systems, such as assembly lines and layer-by-layer additive manufacturing (Amini and Chang, 2018). The proposed model is an extension of our previously published work (MLCPM) for additive manufacturing in that each layer of print is considered as a production stage. Specifically, MLCPM provides a multi-layer predictive model process monitoring tool for the metal 3D print industry. MLCPM benefits from several supervised and unsupervised ML methods to tackle the prediction and high dimensionality problems. MLCPM is the base of this work. The proposed framework includes multiple stage-wise, predictive models that incorporate process parameters in a semiconductor production system. Since the

dimension of this kind of process parameters is enormous, several data reduction techniques are applied to make the problem less complex to save the time of computation. For example, the SECOM (Dheeru and Taniskidou, 2017) dataset used as the case study in Section 5 includes around 600 parameters. Hence, dimension reduction techniques are necessary to reduce computation costs and avoid overfitting.

Also, any ML model, including the proposed ones, requires updated training data for prediction accuracy. Moreover, the unbalanced nature of the process monitoring problem, that is more IC observations available than OC ones may cause too many false alarms (i.e., type I errors) or miss out defective prone processes (i.e., type II errors). Hence, we have investigated ways to mitigate this issue and explored multiple ML techniques to implement the proposed system. The ensemble of ML modelling and computations were implemented by python 2.7 using Scikit-Learn (Pedregosa et al., 2011) package.

4.1 Building predictive models to detect faults

This section provides the initial steps toward building the stage-wise process monitoring in high dimensional multistage systems. Predictive models have been widely used in the industry. Especially in the field of process monitoring, many researchers have proposed the use of predictive models to classify product quality. Most of the studies have considered product quality as either good or defective (binary classification). The predictive models are generally classifiers based on supervised models such as Bayesian network, random forest (Breiman, 2001), and support vector machine (Cortes and Vapnik, 1995) where a set of training data (includes both good and defective classes) are given to train a classifier. Then, the classifier can be used to classify a new observation into either the good or defective category. In general, assuming a problem with *n* samples and *m* parameters, a classification model can be modelled as:

$$Y = f(x) \text{ where } (x, Y) = (x_{i1}, x_{i2}, \ldots, x_{im}, Y_i), \quad i = 1, \ldots, n \tag{8}$$

where Y is the class set of data (target value of a quality characteristic), and x is the feature set (AKA process parameter or variable). In a manufacturing process, *x* represents the setting of a process parameter such as temperature, and *Y* is the quality state class (which could be 0 as good or 1 as defective). In some cases, the quality state can be more than two classes. In that situation, decision tree-based classifiers can work without modification. However, classifiers such as support vector machines (SVM) need to be modified for multilabel classification. Two commonly used methods include one-vs-rest and one-vs-one. Assuming *N* different classes, the one-vs-rest method trains one binary model for each class where classes are based on one class vs. the rest (seen as a single class). In other words, for $j \in \{1, \ldots, N\}$, a single classifier will be trained where the class of j will be seen as positive and the rest as negative. In this approach, a total of N classifiers are trained for all classes. On the other hand, the one-vs-one approach makes a binary classifier for each pair of classes. Therefore, total $N(N-1)/2$ classifiers need to be trained. Either method has its own advantage and disadvantage. One-vs-one is computationally expensive but does not cause imbalance problems where one-vs-rest method encounter imbalance problem and hence, cannot be solved with general classifiers such as generic SVM.

Process parameters could be either numerical or categorical. In a multistage system, the result of each stage is the input to the next stage in the system. Hence, each stage contributes to the final quality state. Depending on the applications, production time varies from seconds to days or even weeks. Hence, predicting the faulty process before the last process can save plenty of time and avoid costs. The first step to build predictive models is data collection of production parameters affecting the final quality state in all stages. Assuming a manufacturing process with *k* process parameters scattered in various production stages, a training dataset that contains *n* produced products can be listed in Table 1.

Table 1 A training dataset

	Production Parameters				
Part	*Parameter 1*	*Parameter 2*	*…*	*Parameter k*	*Target Value*
Part 1	x_{11}	x_{12}	…	x_{1k}	Y_1
Part 2	x_{21}	x_{22}	…	x_{2k}	Y_2
…	…	…	…	…	…
Part n	x_{n1}	x_{n2}	…	x_{nk}	Y_n

where x_{ij} is the measured value of the jth process parameter for part i and is the quality characteristic for part i. Using the training data as Table 1, we propose to establish a predictive ensemble platform based on *k* predictive models as follow:

$$\text{Predictive Models} \begin{cases} \text{Model 1: } Y = f_1(x) \text{ where } (x, Y) = (x_{i1}, Y_i) \\ \text{Model 2: } Y = f_2(x) \text{ where } (x, Y) = (x_{i1}, x_{i2}, Y_i) \\ \dots \\ \text{Model k: } Y = f_k(x) \text{ where } (x, Y) = (x_{i1}, x_{i2}, \dots, Y_i) \end{cases} \tag{9}$$

The proposed platform enables online process monitoring during the production process, where the data up to each point, can be used in the set of models to predict the final quality state. Therefore, *k* different assessments on the final quality state are performed to ensure the quality process.

Several classification models are promising candidates for the function *f*(*x*) in equation (9). Based on (Omar et al., 2013), K-Nearest Neighbours (KNN), Naïve Bayes (NB), Neural Network (NN), and SVM are effective classifiers for anomaly detection problems. Also, Logistic Regression (LR) and Random Forest (RF) are among the list of classifiers. In unbalance classification problems, accuracy is not the best evaluation measurement where specificity, sensitivity, and area under the curve (AUC) are more effective. Hence, we propose to compare the classifiers based on the specificity, sensitivity, and AUC evaluation. The specificity and sensitivity equations are as follow:

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} \tag{10}$$

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{11}$$

where true negative refers to the correctly classified bad parts, true positive refers to correctly classified good parts, false negative (type II error) refers to the parts that are truly bad while the model has misclassified them as good, and the false positive (type I error) refers to the truly good parts, however, the model has mistakenly classified them as bad. These measurements are usually placed in a confusion matrix for the evaluation of any classifier. For more details about the confusion matrix, the reader can refer to (Lancaster, 2003). AUC calculates the area under the receiver operating characteristic (ROC) curve, which is a figure that resulted from plotting true positive rate and false positive rate of classifiers. Higher the AUC score, the better the quality of prediction.

4.2 Tackle dimensionality problem

The proposed system can detect faulty products while the product is still in a production process but may generate many false alarms and disrupt normal production operations. This issue is more pronounced in a high dimensional problem because a vast number of parameters (K) are included. Hence, we propose the use of various techniques to reduce the time and computation. The dimension reduction can be applied in two different stages.

4.2.1 Stage clustering

We assume that the production process in a stage follows specific patterns. Therefore, instead of feeding the original raw data for modelling, clustering methods are proposed to identify these patterns. Model building time will be significantly reduced when the clustering is accomplished. Toward this goal, the data from each stage should be gleaned from Table 1. Note that curating data into the format shown in Table 2 is nontrivial because process parameter data comes in different timestamps at different stages. Assuming p parameters in stage j, Table 2 illustrates the production data for stage j where X_{ijk} represents the collected data for part i and parameter k in stage j.

Table 2 Data collected for stage j

	Production parameters in stage j			
Part	*Parameter 1*	*Parameter 2*	…	*Parameter p*
Part 1	X_{1j1}	X_{1j2}	…	X_{1jp}
Part 2	X_{2j1}	X_{2j2}	…	X_{2jp}
…	…	…	…	…
Part n	X_{nj1}	X_{nj2}	…	X_{njp}

To perform the clustering analysis, we propose to use the K-means algorithm in Scikit-Learn (Pedregosa et al., 2011) package available for Python 2.7. K-means is the widest clustering method that has been used in many applications. After applying the K-means in Table 2, a cluster will be assigned for each part's data in a stage. It should be noted that the K-means ++ method embedded in Scikit-Learn, and Euclidean method are chosen as initial centroid and distance method for K-means. Since the K-means algorithm needs the knowledge of the number of clusters, the 'elbow method' (Ketchen and Shook, 1996) is used to determine an adequate amount of clusters. The elbow method uses the

distortion within clusters to assess the efficient number of clusters. The elbow method is a graphical tool to detect an efficient number of clusters.

Most stages may have many possible production setting combinations. However, we assume that there are only small limited numbers of combinations that are used in production.

The K-means algorithm used first assigns each processed part in a given stage to a cluster based on equation (12). Then it is repeated for all parts and all stages to form the matrix illustrated in Table 3. The outcome of this step is that each part has one identified cluster for each stage.

$$C_{ij} = g(X_{ijk}), C_{ij} \in \left\{C_{1j}, C_{2j}, \ldots, C_{l_j j}\right\} \tag{12}$$

where g is the clustering function, X_{ijk} is the measurement of parameter k for part i in stage j, C_{ij} is the assigned cluster for part i within stage j and l_j is the number of clusters for stage j obtained by the elbow method.

Table 3 Classification matrix

	Production Stage (j)				
Part (i)	*Stage 1*	*Stage 2*	*...*	*Stage M*	*Target value (Y)*
Part 1	C_{11}	C_{12}	…	C_{1M}	Y_1
Part 2	C_{21}	C_{22}	…	C_{2M}	Y_2
…	…	…	…	…	…
Part *n*	C_{n1}	C_{n2}	…	C_{nM}	Y_n

Assuming total M production stages in Table 3, Y_i is the product quality characteristic for part i as it can be simply defined as 0 (for good) and 1 (for defective) part. Table 3 is then used as the training and testing set for predictive models (equation (9)). Hence, the total number of required predictive models drops from K to only M models. This step reduces computation time as the manufacturing processes usually include many processes parameters (K) while the number of stages (M) is limited.

Clustering reduces the parameters per stage into a limited number of classes (i.e., the assigned cluster). In other words, each cluster represents a production recipe in a production stage. The training set is then used to perform the clustering model. After training, the trained clustering model will assign an appropriate cluster to each new data. This process has a huge impact on model reduction. Then, instead of using a large number of process parameters, the assigned clusters can be used in the predictive models of equation (9). However, the stages that have only one process parameter may not need to go through the clustering process.

4.2.2 Stage selection

In multistage systems, not all the stages have the same impact on the final quality characteristic. We propose the use of a few selected highly important stages for building predictive models. Feature selection techniques can be applied to achieve this purpose. For example, RF provides the feature importance values in the predictive models. Then, a cut off number can be chosen to select only the most important stages. However, several

different methods, such as BN, LR, and KNN, can also be applied and compared to the RF algorithm. In this study, we utilise several classification models on all stages and then, using the best classification model in terms of AUC for the stage selection process (feature selection).

This step will reduce the number of predictive models when limited, or the most important models are selected. For example, in a production process containing 100 stages, RF can sort the importance of the stages regarding the product quality. Then, the limited number of those stages (for example, stage 20, stage 50, and stage 90) can be chosen to predict the product quality. Then, three predictive models will be generated where the first predictive model includes stages up to 20, but with stage 20, the most important stage. The second predictive model includes stages up to 50 but with the most significant stages, 20 and 50. The last predictive model includes the most significant staged up to 90 but with the most significant stages 20, 50, and 90. In general, assuming *m* significant stages in a production system with a total of *M* stages, *m* predictive models can be built as follow:

$$\text{Predictive Models} \begin{cases} \text{Model 1: } \hat{Y}_i = f_1(C_{i1}) \\ \text{Model 2: } \hat{Y}_i = f_2(C_{i1}, C_{i2}) \\ \dots \\ \text{Model m: } \hat{Y}_i = f_m(C_{i1}, C_{i2}, \dots, C_{im}) \end{cases} \tag{13}$$

where C_{ij} is the predicted cluster for part *i* in stage *j* and Y_i is the predicted quality for part *i*. Two proposed dimension reduction techniques will hugely reduce the time and effort of performing the proposed process monitoring framework.

4.3 Training update process and imbalance classes

As stated in Section 3, the update process and Imbalance classes are the two main challenges for the proposed framework. Once the predictive models are trained, new datasets in the deployment process can be evaluated. Then, this new dataset can be appended to the training set. So, the new training set grows as the new datasets join. Generally, more data improves the accuracy of data-driven approaches (Amini and Chang, 2018). However, the established models are based on the previous training sets, and they need to be evaluated using the updated dataset.

To address the update process, we propose to use the AUC score as a threshold scale. First, a threshold and trained models evaluation time should be specified. Since the AUC score can identify the misclassifications, a threshold should be set to trigger a warning when the AUC score goes below the threshold. At this point, clustering and prediction models should be trained again using the new training set. These updated models will reduce the chance of misclassification. However, while the models are under training, the process monitoring can still take place using the old models. As the training set grows, the training time will increase. But the model performance is expected to improve. A balance must be struck for how often to repeat the clustering and train process. The pseudo-code for the training update procedure can be seen as follow:

Specify the training_evaluation time and the threshold
Trigger training procedures.
Deploy the trained models on new coming samples.
Evaluate the new sample and add it to a separate dataset called temp_training.
Update the current time.
If time = training_evaluation time, then
    Deploy the trained models on the temp_training and obtain AUC.
    If AUC>= threshold then
continue
    Else
        add temp_training to the training set, trigger the training procedures using the new training set

Two general approaches include oversampling and undersampling, have been considered to address the problem of unbalanced classes. As the names suggest, the undersampling method aims to balance the classes by trimming the data in the majority group while the oversampling method seeks to increase the numbers of samples in a minority group. In this study, synthetic minority over-sampling technique (SMOTE) (Chawla et al., 2002) has been selected as the oversampling method as it does not merely randomly copy from the available points in the minority group, but also it creates synthetic minority class examples. The algorithm selects similar samples from the minority group (using distance methods) and creates an instance using interpolation of the points chosen. In this study, the SMOTE method is the one used in Scikit-Learn (Pedregosa et al., 2011) package for Python 2.7. These techniques will be applied to the SECOM (Dheeru and Taniskidou, 2017) dataset in the numerical example section.

4.4 Model deployment (process monitoring)

Once the clusters and predictive models are generated, new production data can be evaluated by the trained models. First, K-means is applied to data in each stage to create clusters. Then, when the first significant stage is reached, the first predictive model (equation (13)) will predict the initial result. This prediction provides the likelihood of the quality outcome while the semi-finished product is still in the manufacturing process. If the result is a success, the process continues up to the next significant stage, where the second predictive model can be applied to predict the quality. Otherwise, engineers have enough time to change the process parameters so that this semi-finish part may have a better chance to be good. The same procedure is repeated until all of the crucial stages are reached.

The main difference between the proposed model and the other studies is that the proposed model provides an evaluation of the product while the product is still in the production process. The other ML methods reviewed in the literature all wait until the very end to generate a prediction. Generally, at the end production stage, product quality has been determined, and there is no chance to go back and adjust process parameters to improve product quality. Details of the proposed method will be shown in a numerical example in the next section.

5 A numerical example

The proposed model is applied to the SECOM dataset – a semiconductor manufacturing dataset extracted from the UCI repository lab (Dheeru and Taniskidou, 2017). Each row of the dataset contains production parameters and the final quality stage. SECOM consists of 1567 examples, each with 591 features and a label containing the classification of the quality characteristic. The classifications reported as +1 stands for a defective part while –1 is for a good part. In all, 104 parts were identified as defectives (+1), and 1463 parts were good (–1). It is clear that this dataset is very unbalanced in terms of the classes presented.

5.1 Data preprocessing

Among 591 features, 116 of them are fixed (meaning its row value does not change) and therefore do not contribute any information toward this classification problem. Hence, they were dropped from this study. A manufacturing sensor generates each data point according to Arif et al. (2013). Data columns representing process parameters can be divided into five groups, each representing a manufacturing workstation. Besides, there are missing data reflected as empty cells in the dataset. In this study, all empty cells were filled with the most frequently occurred number within each feature.

The first step is to divide all columns of data based on five production stages, as stated in Kao et al. (2017). Table 4 demonstrates the process parameter data within all stages. For example, the first stage contains parameter readings from parameter 1 to parameter 107. After splitting the data, K-means can be applied to assign clusters to parts within each stage.

Table 4 SECOM dataset

	Stage 1				...	*Stage 5*			
Part	*Parameter 1*	*Parameter 2*	...	*Parameter 107*	...	*Parameter 493*	*Parameter 494*	...	*Parameter 590*
Part 1	3030.93	2564.00	…	0	…	10.0167	2.9570	…	0
Part 2	3095.78	2465.14	…	0	…	10.0167	3.2029	…	208.2045
…	…	…	…	…	…	…	…	…	…
Part 1567	2944.92	2450.76	…	0.0009	…	10.0167	2.7756	…	137.7844

5.2 Clustering

Before applying K-means, the efficient number of clusters for each stage data should be identified. According to the elbow method, 4,3,3,4, and 4 are the most efficient number of clusters for stages 1 to 5, respectively. For example, according to the elbow chart in Figure 2, stage 5 can be divided into 4 clusters.

After identifying the clusters, K-means can be applied on stage data to assign a cluster to each part within a stage. Then using identified clusters, a classification matrix can be formed as shown in Table 5 based on the SECOM dataset. Each value under production stage columns in Table 5 represents the index of the assigned cluster for the respected data points.

Figure 2 Elbow method for stage 5 (see online version for colours)

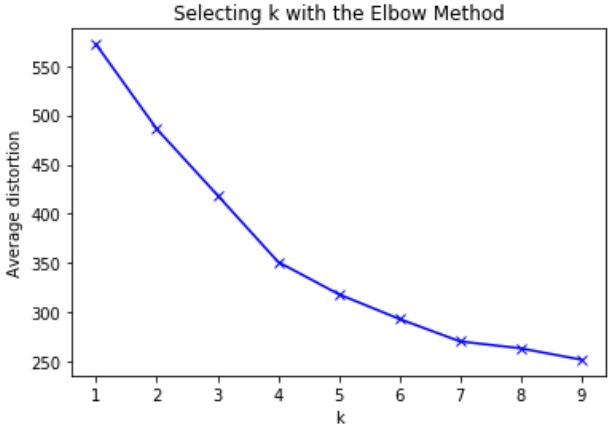


Table 5 Classification matrix

	Production Stage					
Part	*Stage 1*	*Stage 2*	*Stage 3*	*Stage 4*	*Stage 5*	*Target value (Y)*
Part 1	0	0	2	1	3	–1
Part 2	4	3	0	1	3	+1
…	…	…	…	…	…	…
Part 1567	1	0	0	3	4	–1

5.3 Classification

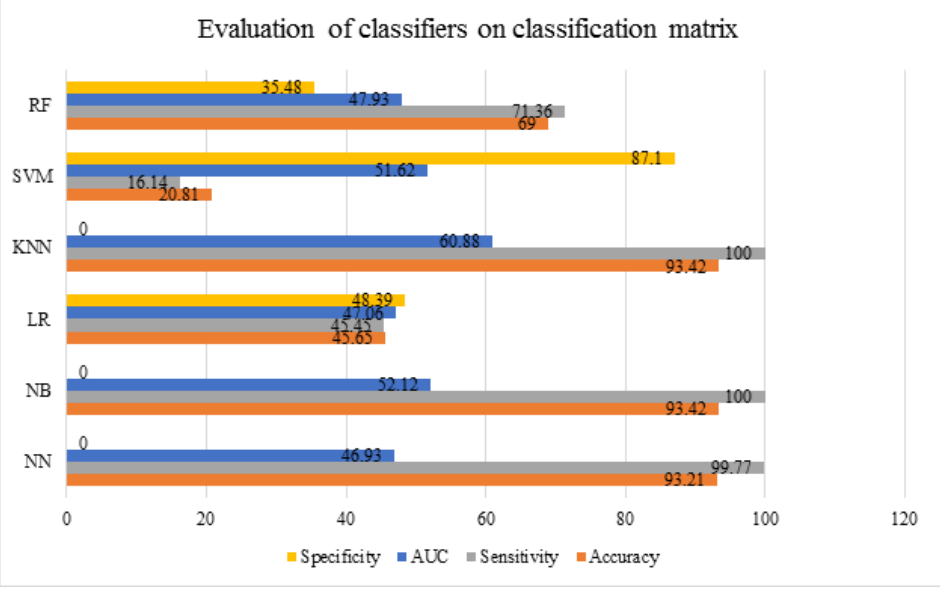
Table 5 is a sample of the input array into classification models. The classification type of this problem is binary and, hence, a handful number of classification models can be applied. We have implemented KNN, NB, NN, LR, RF, and SVM on the classification matrix. Figure 3 shows the evaluation of these models on the classification matrix from Table 5. Evaluation of models is based on sensitivity, specificity, and AUC scores. The classifiers were trained on 70% of the data, while 30% has been reserved for validation. It is noted that the training and testing sets were randomly selected.

Due to the imbalanced nature of the dataset, a balanced classification is required in all classifiers except for NN, KNN, and NB since these classifiers do not adjust weights based on a number of classes. In each set of evaluations (i.e., on original data, with undersampling, and with oversampling), the classifiers have been tuned to perform the best in terms of the AUC score. In addition, we keep an eye on the classification ability to detect both classes (healthy products and failed products) and would like to peek a classifier that has the best AUC considering the detection of both classes. The hyperparameter tuning on each classifier has been performed by the H2O autoML platform (H2O.ai, 2016).

Among all classifiers listed in Figure 4, RF performs the best in terms of classifying all groups. Although KNN, NN, and NB perform better in terms of accuracy, however, they have misclassified all the bad parts (zero specificity). In an imbalanced sampling problem, AUC, specificity, and sensitivity are better tools to evaluate a model than

accuracy. Figure 3 also shows that those classifiers that do not use class weights do not perform well in terms of specificity (higher scores are favoured) since the false-negative (the wrong prediction of +1 group, i.e., predicted as good while it was defective) is high. In SPC, this statistic is often referred to as a Type II error.

Figure 3 Evaluation of classifiers (RF: random forest, SVM: support vector machine, KNN: K nearest neighbourhood) (see online version for colours)



The next step is to tackle the imbalance problem of the dataset. As discussed before, two general approaches are oversampling and undersampling (Chawla et al., 2002). To solve the imbalance classification, we applied techniques using imblearn package (Lemaître et al., 2017) available for both Python versions 2 and 3. It should be noted that in all of the discussed classification models, oversampling and undersampling methods were applied to the training set (70% of the data) but not on the testing set (30%).

Undersampling was applied by imblearn (Lemaître et al., 2017) set where data was bootstrapped from the majority class with the same size of the minority class. Figure 4 was generated by the evaluation of models with the newly trained model (trained with the undersampling method). Figure 4 shows clearly that the undersampling method improves the training of the classifiers in terms of specificity and sensitivity scores. However, the accuracies of the models were dropped dramatically. For example, the NN had an accuracy of 93.21 before undersampling, but the new rate is only 50.11. This phenomenon is due to the fact that by adjusting the samples, accuracy reflects a more realistic number than before.

Next, oversampling was applied to the original data using SMOTE (Chawla et al., 2002) embedded in imblearn (Lemaître et al., 2017). Figure 5 was generated using the oversampling technique.

Figure 4 Evaluation of classifiers after applying the undersampling method (see online version for colours)

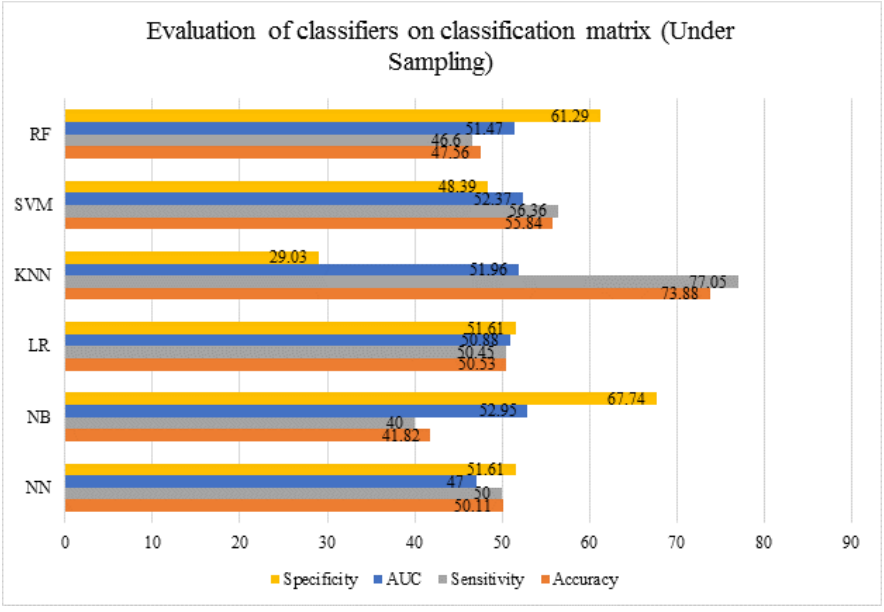
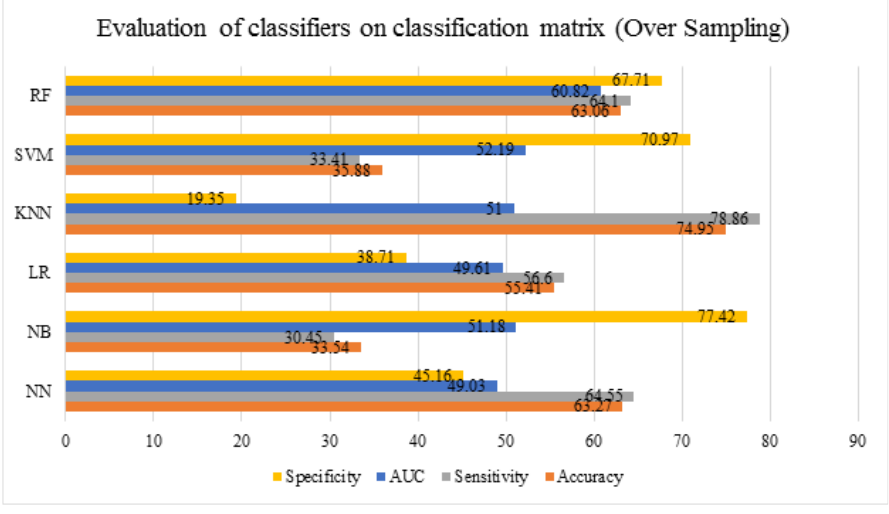


Figure 5 Evaluation of classifiers after applying under the oversampling method (see online version for colours)



Similar to undersampling, the oversampling method by SMOTE (Chawla et al., 2002) improves the training of the classifiers in term of specificity and sensitivity. But unlike undersampling, the accuracy values did not drop too much from the original analysis. As shown in Figure 5, we concluded that RF performs the best in terms of the overall consideration of accuracy, specificity, sensitivity, and AUC metrics. Therefore, oversampling by SMOTE has been selected as the imbalanced approach. Table 7 summarises the evaluation of all different combinations.

After establishing the primary predictive model, the next step is to select the most critical stages to reduce the number of predictive models. RF has an inherent feature property that can weigh the features based on the amount of information they provide toward detecting the patterns of classes. Table 6 shows the importance of stages in the selected classifier.

Table 6 Importance factor for production stages

Stages	*Stage 1*	*Stage 2*	*Stage 3*	*Stage 4*	*Stage 5*
Importance	0.1101	0.3580	0.108	0.1305	0.2934

Table 7 Evaluation of all models

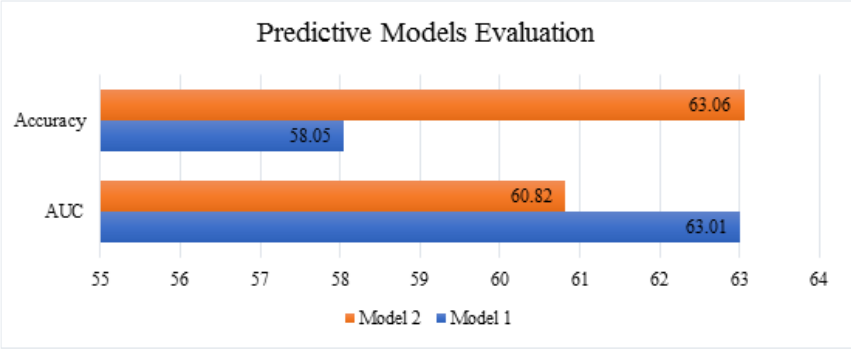
	Original data				*Under sampling*				*Over sampling*			
Model	*Accuracy*	*Sensitivity*	*Specificity*	*AUC*	*Accuracy*	*Sensitivity*	*Specificity*	*AUC*	*Accuracy*	*Sensitivity*	*Specificity*	*AUC*
RF	69	71.36	35.48	47.93	47.56	46.6	61.29	51.47	63.06	64.1	67.71	60.82
SVM	20.81	16.14	87.1	51.62	55.84	56.36	48.39	52.37	35.88	33.41	70.97	52.19
KNN	93.42	100	0	60.88	73.88	77.05	29.03	51.96	74.95	78.86	19.35	51
LR	45.65	45.45	48.39	47.06	50.53	50.45	51.61	50.88	55.41	56.6	38.71	49.61
NB	93.42	100	0	52.12	41.82	40	67.74	52.95	33.54	30.45	77.42	51.18
NN	93.21	99.77	0	46.93	50.11	50	51.61	47	63.27	64.55	45.16	49.03

A significance level of 0.25 is chosen based on the respected importance values for all stages in Table 6. Then, stages 2 and 5 are selected as the most important stages toward predicting the parts' final quality. The stage selection helps to reduce the number of predictive models. This feature selection stage may be trivia in this example. However, it would be significant for other applications such as 3D prints where thousands of layers are required to finish a product. Besides, more predictive models will provide more chances to catch the faulty process. On the other hand, it may also increase false alarm or false-negative rates and time computation. Therefore, there is a tradeoff between these considerations. In this example, two classifiers can be built, as shown in equation (14).

$$\text{Predictive Models} \begin{cases} \text{Model 1: } Y_i = f_1(C_{i1}, C_{i2}) \\ \text{Model 2: } Y_i = f_2(C_{i1}, C_{i2}, \ldots, C_{i5}) \end{cases} \tag{14}$$

The first model includes data from the first two stages, where the second model includes all production stages data. Model 2 was trained previously by applying RF using the SMOTE method on 70% of the data, while 30% was reserved for validation. The same procedure was done for model 1, while only two stages were considered. Figure 6 shows the metrics for both models. These two models perform the diagnostic and prognosis tasks as model 1 alarms regarding a suspected production while the process is continuing and model 2 can be used to tune the process parameters to avoid failure before reaching the final production step.

Figure 6 Evaluation of stage-based predictive models (see online version for colours)



The computations have been performed on an AMD 3700X and 32GBs of ram. Regarding the computation time, the major part is devoted to the preprocessing that has taken 892.33 s while training and testing on the selected model have taken 19.7 s in total. However, evaluation of the quality for a new sample in either model 1 or 2 (equation (14)) is almost 0 s. Hence, the assessment of a newly produced part while it is under production is possible with the fast testing computation.

After establishing the predictive models, the process monitoring phase or Phase II of SPC can be initiated. Specifically, a new part proceeds to the production line. When this new product reaches stages 1 and 2, the K-means models associated with each stage will assign an appropriate cluster to each. Then, the first predictive model would provide a prediction on the final quality state (–1 for a good and +1 for a defective product). If the prediction results as good (–1), then the process continues up to the last stage to provide a prediction using model 2. Otherwise, process engineers have plenty of time to control the process to prevent producing a faulty product by examining historical data of those parts having the same pattern as this new part in the first two stages but still ending to be good part at the end of stage 5. Process engineers can then use the machine settings in stages 3, 4, and 5 of the good parts for this new part.

6 Conclusions and future studies

This research provides a process monitoring framework for high-dimensional, multistage processes. The proposed framework is capable of providing a prognosis of product quality and mitigation strategies before the production of a product is finished. Today's manufacturing processes are much more complicated. Sensors embedded throughout multiple stages of the production processes generate a massive amount of data in high dimensions. The recent development in Industry 4.0 (Foidl and Felderer, 2015) and the IoT (Morgan, 2014) have enabled the environment in which the proposed framework may become a reality.

Traditional quality engineering methods cannot be implemented effectively in this modern-day production environment. Most process parameters are often not used for decision making. Control charting is usually implemented independently throughout the production stages. The recent development in ML methods may provide solution strategies. Although various existing classification-based process monitoring techniques

have been applied for multistage processes, these methods usually provide quality predictions at the end of the manufacturing process and offer no chance to fix potential problems during production. Also, no study has provided any comprehensive model to address issues related to high dimensions, unbalanced training dataset, and the computational speed of big data. Addressing those challenges, the proposed research provides a stage-wise process monitoring approach that provides opportunities for engineers to fix potential process problems before a product reaches its last production stage.

Besides the benefits of the proposed model, however, most models come with limitations, and the proposed model is not an exception. First, although the proposed models can affectively detect and reduce the faulty product, however, it cannot explain which process parameters are the root causes of the faculty conditions. This limitation appears as the proposed method aims to reduce the dimensionality of the problem by using a clustering method. The clustering step advances the whole process in terms of computation time and complexity; however, the proposed models then lose the ability to track the effects of original process parameters on the final quality. Then, a future study is necessary to address this issue further. Second, ML models, in general, require enough training data for satisfactory results and tend to detect the patterns more accurately with having more data. Hence, in this research, the limited available data points have negatively affected the performance of selected models. This limitation, however, may be alleviated as more production data becomes available. Since modern manufacturing machines are now producing a huge amount of data during production states, abundant data points would further improve the performance of the proposed ML models.

For future research, we propose to build a prognosis model such as a Bayesian Network based on the established stage-based predictive models. Since the proposed model generates limited clusters for multiple stage-based modelling, a prognosis model can benefit from tremendous dimension reduction. This prognosis model can be used to suggest process parameter settings in unfinished stages to prevent defective production. Once a part is identified to be potentially faulty, a search procedure is necessary then to find the best possible production settings in the unfinished stages to guide the faulty part to bring it back into healthy conditions. Since the number of parameters is large, an efficient approach is necessary to perform a fast and efficient computation and search among all possible production settings. A prognosis model or algorithm is required to provide potential impactful process parameters and proper settings for control purposes.

References

Amini, M. and Chang, S. (2018) 'A review of machine learning approaches for high dimensional process monitoring', *Proceedings of the 2018 Industrial and Systems Engineering Research Conference*, Orlando, FL.

Amini, M. and Chang, S.I. (2018) 'MLCPM: a process monitoring framework for 3D metal printing in industrial scale', *Computers and Industrial Engineering*, Vol. 124, pp.322–330.

Arif, F., Suryana, N. and B., Hussin (2013) 'Cascade quality prediction method using multiple PCA+ ID3 for multistage manufacturing system', *IERI Procedia*, Vol. 4, pp.201–207.

Breiman, L. (2001) 'Random forests', *Machine Learning*, Vol. 45, No. 1, pp.5–32.

Ceglarek, D. and Shi, J. (1995) 'Dimensional variation reduction for automotive body assembly', *Manufacturing Review*, Vol. 8, No. 2, pp.139–154.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002) 'SMOTE: synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research*, Vol. 16, pp.321–357.

Cortes, C. and Vapnik, V. (1995) 'Support-vector networks', *Machine Learning*, Vol. 20, No. 3, pp.273–297.

Deng, H., Runger, G. and Tuv, E. (2012) 'System monitoring with real-time contrasts', *Journal of Quality Technology*, Vol. 44, No. 1, pp.9–27.

Dheeru, D. and Taniskidou, E.K. (2017) *UCI Machine Learning Repository*, URL http://archive.ics.uci.edu/ml

Foidl, H. and Felderer, M. (2015) 'Research challenges of Industry 4.0 for quality management', *International Conference on Enterprise Resource Planning Systems*, Springer, Munich, Germany.

Friedman, J., Hastie, T. and Tibshirani, R. (2001) *The Elements of Statistical Learning*, Vol. 1, Springer Series in Statistics, New York.

H2O.ai (2016) *Python Interface for H2O. 2016*, H2O.ai: H2O.ai.

Hotelling, H. (1947) Multivariate Quality Control Illustrated by the Air Testing of Sample Bombsites, McGraw-Hill, New York, p.111.

Hu, J. and Runger, G. (2010) 'Time-based detection of changes to multivariate patterns', *Annals of Operations Research*, Vol. 174, No. 1, pp.67–81.

Hu, J., Runger, G. and Tuv, E. (2007) 'Tuned artificial contrasts to detect signals', *International Journal of Production Research*, Vol. 45, No. 23, pp.5527–5534.

Hwang, W., Runger, G. and Tuv, E. (2007) 'Multivariate statistical process control with artificial contrasts', *IIE Transactions*, Vol. 39, No. 6, pp.659–669.

Jiang, W. and Tsui, K-L. (2008) 'A theoretical framework and efficiency study of multivariate statistical process control charts', *IIE Transactions*, Vol. 40, No. 7, pp.650–663.

Jin, J. and Shi, J. (1999) 'State space modeling of sheet metal assembly for dimensional control', *Journal of Manufacturing Science and Engineering*, Vol. 121, No. 4, pp.756–762.

Jin, Y., Huang, S., Wang, G. and Deng, H. (2017) 'Diagnostic monitoring of high-dimensional networked systems via a LASSO-BN formulation', *IISE Transactions*, Vol. 49, No. 9, pp.874–884.

Kao, H-A., Hsieh, Y-S., Chen, C-H. and Lee, J. (2017) 'Quality prediction modeling for multistage manufacturing based on classification and association rule mining', *MATEC Web of Conferences*, EDP Sciences, Kending, Pingtung, Taiwan.

Ketchen, D.J. and Shook, C.L. (1996) 'The application of cluster analysis in strategic management research: an analysis and critique', *Strategic Management Journal*, Vol. 17, No. 6, pp.441–458.

Lancaster, F.W. (2003) 'Precision and recall', in Bates, M.J. and Maack, M.N. (Eds.): *Encyclopedia of Library and Information Science*, Lib-Pub., pp.2346–2351.

Lemaître, G., Nogueira, F. and Aridas, C.K. (2017) 'Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning', *The Journal of Machine Learning Research.*, Vol. 18, No. 1, pp.559–563.

Li, F., Runger, G.C. and Tuv, E. (2006) 'Supervised learning for change-point detection', *International Journal of Production Research*, Vol. 44, No. 14, pp.2853–2868.

Lowry, C.A., Woodall, W.H., Champ, C.W. and Rigdon, S.E. (1992) 'A multivariate exponentially weighted moving average control chart', *Technometrics*, Vol. 34, No. 1, pp.46–53.

Marr, B. (2016) *A Short History of Machine Learning*, Forbes, Editor. 2016, Forbes, pp.1–2.

Morgan, J. (2014) *A Simple Explanation of 'The Internet of Things', Leadership* [cited 2018 12/1/2018]; Available from: http://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#475fab4e6828

Omar, S., Ngadi, A. and Jebur, H.H. (2013) 'Machine learning techniques for anomaly detection: an overview', *International Journal of Computer Applications*, Vol. 79, No. 2, pp.33–41.

Pan, J-N., Li, C-I. and Wu, J-J. (2016) 'A new approach to detecting the process changes for multistage systems', *Expert Systems with Applications*, Vol. 62, pp.293–301.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. and Dubourg, V. (2011) 'Scikit-learn: machine learning in python', *Journal of Machine Learning Research*, Vol. 12, October, pp.2825–2830.

Poggio, T. and Smale, S. (2005) 'The mathematics of learning: dealing with data', *2005 International Conference on Neural Networks and Brain*, IEEE, Beijing, China.

Runger, G.C. and Testik, M.C. (2004) 'Multivariate extensions to cumulative sum control charts', *Quality and Reliability Engineering International*, Vol. 20, No. 6, pp.587–606.

Shewhart, W. (1930) 'Economic quality control of manufactured product 1', *Bell System Technical Journal*, Vol. 9, No. 2, pp.364–389.

Shi, J. and Zhou, S. (2009) 'Quality control and improvement for multistage systems: a survey', *IIE Transactions*, Vol. 41, No. 9, pp.744–753.

Tibshirani, R. (1996) 'Regression selection and shrinkage via the lasso', *Journal of the Royal Statistical Society Series B*, Vol. 58, No. 1, pp.267–288.

Tuv, E. and Runger, G. (2003) 'Learning patterns through artificial contrasts with application to process control', *WIT Transactions on Information and Communication Technologies*, p.29.

Uhlmann, E., Pontes, R.P., Laghmouchi, A. and Bergmann, A. (2017) 'Intelligent pattern recognition of a SLM machine process and sensor data', *Procedia Cirp.*, Vol. 62, pp.464–469.

Wang, K. and Jiang, W. (2009) 'High-dimensional process monitoring and fault isolation via variable selection', *Journal of Quality Technology*, Vol. 41, No. 3, pp.247–258.

Wuest, T., Irgens, C. and Thoben, K-D. (2014) 'An approach to monitoring quality in manufacturing using supervised machine learning on product state data', *Journal of Intelligent Manufacturing*, Vol. 25, No. 5, pp.1167–1180.

Zou, C. and Qiu, P. (2009) 'Multivariate statistical process control using LASSO', *Journal of the American Statistical Association*, Vol. 104, No. 488, pp.1586–1596.