



A CPPS based on GBDT for predicting failure events in milling

Y. Zhang¹ · X. Beudaert² · J. Argandoña² · S. Ratchev¹ · J. Munoa²

Received: 16 December 2019 / Accepted: 11 September 2020 / Published online: 26 September 2020
© The Author(s) 2020

Abstract

Cyber-physical production systems (CPPS) are mechatronic systems monitored and controlled by software brains and digital information. Despite its fast development along with the advancement of Industry 4.0 paradigms, an adaptive monitoring system remains challenging when considering integration with traditional manufacturing factories. In this paper, a failure predictive tool is developed and implemented. The predictive mechanism, underpinned by a hybrid model of the dynamic principal component analysis and the gradient boosting decision trees, is capable of anticipating the production stop before one occurs. The proposed methodology is implemented and experimented on a repetitive milling process hosted in a real-world CPPS hub. The online testing results have shown the accuracy of the predicted production failures using the proposed predictive tool is as high as 73% measured by the AUC score.

Keywords Cyber-physical production systems (CPPS) · Production failure · Dynamic principal component analysis (DPCA) · Gradient boosting decision trees (GBDT) · Machine learning

1 Introduction

Cyber-physical production systems (CPPS) are mechatronic systems (physical part) monitored and controlled by software brains and digital information (cyber part) [1]. The presence of the high-technological information and communication component aims to provide factories with self-aware and self-adaptive production patterns to restore the monitored production process from the faulty status to the normal functional condition [2]. Tool condition monitoring (TCM), in particular, is a critical function of the CPPS, which allows early failure detection by continuous monitoring over the machine tool and the overall production process in order to increase the reliability and productivity of the machine tool and keep a high-quality control over the produced workpieces [3, 4]. Within the field of TCM, the investigation and development of the

failure predictive tool have attracted tremendous interests from researchers across different fields.

A wide range of failure prediction methods have been explored to date, which can be broadly categorised into model-based [5, 6] and data-driven techniques [7, 8]. A premise condition of the model-based methods is the prior knowledge of both the physical system and the mathematical foundation. The model-based techniques have demonstrated its suitability and efficiency in applications varying from fault detection for compressors and pumps to process monitoring for electronic induction [9–11] and automatic control systems in multiple manufacturing processes [12–14]. Recently, the reduced cost of machine sensors, the enormous growth in computational power and the tremendous surge of the cloud systems have driven more attention to the other data-driven alternative, underpinning which is the generalisation of statistical methods and machine learning algorithms at scale [15].

The data-driven methods are particularly interesting for CPPS which use data as a non-physical medium to communicate information carrying production process status to core control computers. Numerous data-driven methods have been developed since the first introduction of the concept in 1930 [15], generally classified into three mainstreams including the two multivariate statistical approaches: principal component analysis (PCA) [16, 17] and partial least square (PLS) [18, 19] and the third machine learning approach using algorithms such as neural networks [17, 20], [21], support vector machine

✉ Y. Zhang
y.zhang1205@gmail.com

✉ X. Beudaert
xbeudaert@ideko.es

¹ The Institute for Advanced Manufacturing, The University of Nottingham, Nottingham, UK

² Dynamics and Control, Ideko, Elgoibar, Basque Country, Spain

[22, 23] and tree-based models [24]. Both PCA and PLS are able to reserve the most significant variety of the original set of parameters, which are assumed to follow Gaussian distributions. PLS differs from PCA by taking the relationship between the inputs (the measured parameters and their variations) and outputs (machine statuses of interests such as production failure or faults) into account during the model training stage on the batch data [8].

Despite their low computational complexity, both PCA and PLS methods require further improvements in robustness and accuracy for predicting failures in a real production environment, where parameters appear dynamic (non-stationary) and nonlinear. The dynamic and kernel techniques have been investigated to complement the PCA and the PLS (resulting in dynamic PCA [25–27], dynamic PLS [18, 19, 28], kernel PCA [24, 29] and kernel PLS [30, 31]), and to tackle the system dynamics and nonlinearity respectively. A kernel PCA first maps the original input space to a higher dimensional nonlinear space and then implements the PCA on the converted space in order to obtain the nonlinear principal components. DPCA differs from PCA in applying the PCA on to a time lag shifted input space instead of the original input space. The two techniques can also be applied simultaneously in association with the PCA to handle systems which possess both of the common yet troublesome properties [24]. As for the machine learning algorithms, the support vector machine [22, 23] and neural networks [17, 20], [21] have been intensively explored solely or in combination with previously mentioned multivariate statistical models to resolve the nonlinearity and dynamics between parameters those do not follow Gaussian distributions. A tree-based model with the sliding window technique has been proposed in [32], which has been shown to provide an accurate prediction of production failures with less computational complexity compared with the above two algorithms.

The comparison of the mentioned methods from the perspectives of mainstreams and their main successors, assumptions on parameters and computational complexity, is presented in Table 1.

Considering the application of the data-driven approaches in milling processes within a CPPS environment, various parameters can be measured to reflect the real-time status of the machining process, with common ones including spindle power, vibration spectrum and severity, feed rate and spindle speed. However, due to the long manufacturing processes of work pieces, it is impractical for a technician to identify abnormal signals by continuously monitoring the process patterns. In addition, it is wise to give consideration to the relationship between the measured parameters in order to accurately determine the occurrence of the faulty process behaviour [33]. Therefore, in order to make the most of the real-time sensory data and provide the most useful information to the decision-makers, a computational efficient and economically affordable machine tool monitoring system is required to be developed within milling CPPS environment, while taking the system dynamics into consideration. This paper introduces a novel process monitoring strategy, composed of the DPCA to extract the auto-correlations in the dynamic milling process and to reduce the dimension of the parameter space, followed by the gradient boosting decision tree (GBDT). GBDT is a supervised machine learning model (a type of model that requires known inputs variables and labelled target variables to train), constituted by a series of decision trees with each successive tree attempting to correct the prediction error of the previous series. It has been increasingly popularised due to its accurate prediction power for large-scale datasets and low computational complexity. In addition to its prediction capability, the GBDT model also provides feature importance (a score that quantifies the contribution of any feature to the overall prediction decision) as a by-product and thus can be utilised to understand the relative importance of involved features.

The remainder of the paper is organised as follows: Section 2 describes the paradigm of the CPPS and gives a detailed description of the data collection mechanism. In Section 3, the concepts of the DPCA and the GBDT are presented. This is followed by the presentation of the machine tool predictive tool underpinned by the hybrid model of the

Table 1 The comparison of existing data-driven methods

Mainstream	Successors	Parameter assumptions	Computational complexity
PCA	DPCA	Dynamics	Median
	KPCA	Nonlinearity	Median
PLS	DPLS	Dynamics	Median
	KPLS	Nonlinearity	High
Machine learning algorithms	SVM	Nonlinearity	High
		Non-Gaussian	
	Tree-based model	Dynamics nonlinearity	Median
	Neural networks	Dynamics	High
		Nonlinearity	
		Non-Gaussian	

DPCA and the GBDT. The proposed predictive tool is then implemented and tested in a comprehensive real-world milling process with the results presented. The final section concludes the current research and presents potential future works.

2 The design of the cyber-physical production system

This section introduces the architecture of a cyber-physical production system (CPPS), which will then be implemented in Section 4 on a large milling machine. The framework of the CPPS is illustrated in Fig. 1.

2.1 Data collection

One important element of a CPPS is the shop floor machine tool that is equipped with sensors and associated devices to measure different features of the production process (e.g. spindle speed, spindle power and vibration level). This data is then processed and uploaded onto the cloud for storage. The other fundamental element is a modern machine tool with central control system like computer numerical controlled (CNC) to gain access to all the sensory data and perform a closed-loop control. The CNC also provides information about the current part program running on the machine, the current program line and the programmed feed rate and spindle speed. Moreover, the feed and spindle speed overrides are relevant for the monitoring platform as they correspond to the actions of the machine operator. In the industrial application analysed in this article, the machine operator can change the feed or spindle override if he or she detects any problem or stop the current

execution and use the manual mode to move the tool away from the work piece and change broken cutting edges during the machining operations.

To obtain additional and high-frequency information on the manufacturing process, additional sensors are connected to a real-time controller that can process the signals. Acceleration sensors have high sensitivities and can be used for chatter detection, condition monitoring or collision detection [34]. Two accelerometers are allocated on the spindle housing (the part holding the spindle) perpendicularly along the *x*-axis and *y*-axis respectively on a plane paralleling to the work piece to take measurements, including vibration severity and amplitude and acceleration jumps for both axes. The measurements are processed and passed to an industrial computer and then uploaded to the cloud server as part of inputs for the predictive tool to forecast production stops. Transferring all this high-frequency data to the cloud platform would be burdensome and costly, so local processing of the acceleration signals is performed to extract the most meaningful information. The fast Fourier transform is used to obtain the frequency spectrum of the vibration, and only the ten highest peaks are transferred to the cloud platform, the vibration severity is also computed for different frequency ranges (from 20 to 80 Hz and from 20 to 1000 Hz). The lower frequency range corresponds to the vibration frequencies of the complete machine structure and it is monitored with special attention because strong chatter vibrations can appear within this frequency range. The second vibration severity range covers the complete vibration spectrum. Hence, the vibration level can be well reflected with a limited amount of data without loss of material information.

The measurements of the internal CNC sensors are collected every second, same as the pre-processed vibration signals

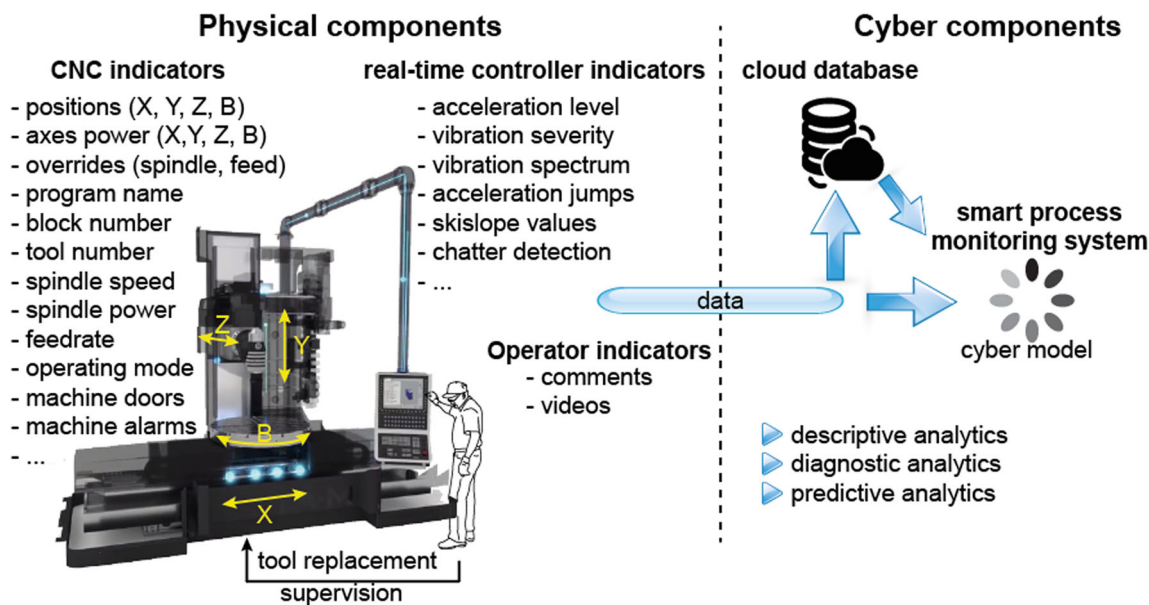


Fig. 1 Framework of the milling cyber-physical production system

coming from the real-time controller. Both slowly changing variables (e.g. tool parameters, spindle speed, program line number) and highly dynamic variables (e.g. vibration severity, spindle power) are transferred and stored in cloud-based databases to predict future production failures.

2.2 Data storage

A manufacturing operation can be complex and highly process-oriented. The real-time data collected from a large number of shop floor machine sensors is typically with high frequency, large volume and high dimension. All the collected data is required to be pre-processed and stored in a database that is capable of handling vast quantities of data with different types while keeping its integrity, resilience and security at a required level. For this purpose, a cloud-based database is utilised to store the pre-processed measurements. Apart from sensory data collected from shop floor machines, the machine alarms and the operator comments are also stored in the databases. Due to the variety of data types (structured data such as processed sensory data and unstructured data like operator comments), a NoSQL database, Amazon CouchDB [35], is used. The information stored in the database is accessible to any party with granted permission.

2.3 Data analytics: the machine tool monitoring system

The monitoring system offers online monitoring and supports adaptive controls over the undergoing production process by extracting available and necessary historical data from the cloud database together with the real-time data from the shop floor machine sensors. Both sources are used to realise different functionalities with embedded analytical models. The process monitoring system provides functionalities including the following:

- Descriptive analytics that analyses and monitors current and historical production status of the machine tools, such as the cutting tools used for specified part programs, days and machine tool. This functionality has been realised by a visual analytics commercial software (e.g. QlikView [36]).
- Prognostic analytics that detects faults having occurred and predicts failures that will happen in the near future based on the most updated sensory data. The real-time data collected from a wide range of machine sensors each second brings a new challenge to existing business software in terms of the prediction accuracy and speed.
- Diagnostic analytics that provide fault isolation and identification by locating the causes that result in the identified production failures. Tool wear and tool breakage are the

two main reasons that result in production stops and downtimes.

This research will focus on developing a novel predictive tool that benefits the prognostic functionality. Due to the large number of sensors and the frequency of measurements and collection (per second), the dataset is expected to belong to the big data scope (with high volume, high velocity and high variety). That is why an analytical model that could realise fast and accurate analysis such as the gradient boosting decision trees should be explored. The objective is to build a system able to automatically stop the machining process in the case of a faulty behaviour to help minimising human efforts and cost on production monitoring. Ideally, human operators will only be required to make decisions when the control system is not authorised to. The operator is also needed to provide feedbacks to the process monitoring system on false alarms or inappropriate suggested activities to improve the models.

3 The proposed predictive tool

This section introduces a novel hybrid model based on dynamic principal component analysis (DPCA) and the gradient boosting decision trees (GBDT) to predict failures in production processes.

3.1 Dynamic principal component analysis

The dynamic principal component analysis (DPCA) has been widely employed to detect faults in process monitoring since first introduced in [27]. In this research, the DPCA will be utilised to extract time-dependency within the same variable and to reduce the dimensionality of the input variables in the time domain. Like PCA, DPCA is a feature extraction technique which can achieve a set of reduced-dimensioned and uncorrelated features by linear combing the time lag shifted original inputs.

Considering an original dataset that contains m inputs and n observations, $X = [x_1, x_2, \dots, x_m]_{m \times n}$. In the case when the inputs are generated from a dynamic system, there exists time-dependency within each of the time series, x_i , meaning that the current value relies on the past values. Applying the standard PCA on the l time lag inputs $X_{(t:t-l)}$, where $X_{(t:t-l)} = [x_1(t), x_2(t), \dots, x_m(t), x_1(t-1), x_2(t-1), \dots, x_m(t-1), \dots, x_1(t-l), x_2(t-l), \dots, x_m(t-l)]_{m(t+1) \times n}$ will eliminate the time linearity within each x_i as well as exclude highly correlated inputs. PCA is a dimensional reduction technique that reduces the original dataset into a set of uncorrelated or orthogonal vectors through a linear transformation as mentioned in Section 1. By solving the eigenvalue problem,

$$\lambda v = \sum_{X_{(t:t-l)}} v \quad (1)$$

where $\Sigma_{X_{(t:l)}}$ is covariance of $X_{(t:l)}$ [37]. Eigenvectors with the first p largest eigenvalues are selected to keep the variance to a certain level and are then multiplied by $X_{(t:l)}$ to get reduced-dimensional inputs, $X_{cp} = [x_1(t), x_2(t), \dots, x_\alpha(t), x_1(t-1), x_2(t-1), \dots, x_\beta(t-1), \dots, x_1(t-l), x_2(t-l), \dots, x_\gamma(t-l)]_{p \times n}$ [37]. X_{cp} contains p variables, where $p < m * (l + 1)$. Note that as the dimension of the time lag shifted feature space $[X_{(t:l)}]_{(l+1) \times m}$ is $l + 1$ time more than the dimension of the original input space $[X]_{m \times n}$; the dimension of the selected principal components $[X_{cp}]_{p \times n}$, which is used as inputs to train the GBDT model, may be larger than the dimension of $[X]_{m \times n}$.

3.2 Gradient boosting decision trees

Gradient boosting decision tree (GBDT) is one of the supervised machine learning models that require both known input variable and target variables to utilise an algorithm to learn a function mapping from the inputs to the target. It has been deployed as the fundamental component of the embedded failure prediction methodology due to its capability in handling complex relationships and interaction effects between measured inputs automatically [38], providing better interpretability than other machine learning approaches like support vector machines or neural networks [39] and low computational complexity, which makes it realistic to be utilised and implemented to produce valuable prediction results in a real-world production environment [40–42].

A decision tree is an acyclic graph starting from a root node and ending at leaf nodes where each split node (either the root node or an intermediate node) is connected with two branches ('True' or 'False'). The split node represents a splitting condition and the leaf node represents the predicted values of outputs. The depth of a decision tree refers to the maximum number of layers of the tree. For the example tree shown in Fig. 2, it has two layers above the leaf nodes and thus has a depth of 2. Considering an input variable $x = \{x_1 = 7, x_2 = 1\}$,

the tree model will give a predicted target value $y = -1.2$, as x went to the true branch on the root node and false branch on the intermediate node. For a binary classification problem, the leaf node has a probability value ranging from 0 to 1 and a threshold will be chosen to determine its class type. A probability value above the threshold will be considered 'positive' as a failure event and below the threshold 'negative' as a non-failure event.

The GBDT constitutes a sequence of decision trees with each successive tree being built to correct the error of the precedent trees (as illustrated by Fig. 2). The full algorithm to construct the GBDT is detailed in Appendix 1.

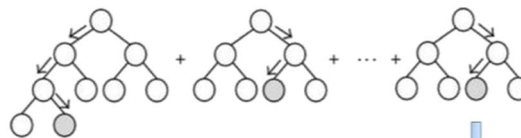
During the construction of the series of decision trees, the feature importance is computed explicitly, leading to a critical secondary product of a GBDT model that allows features to be ranked and compared with each other. The feature importance is a score quantifying the significance of a feature contributing to the structure and the prediction power of the sequence of decision trees. The higher the value, the more the feature is employed to make critical decisions within the GBDT model and vice versa.

A single decision tree is constructed in a top-down manner such that a node that improves the performance measure (weighted by the number of observations the node is responsible for) the most is located higher in a tree. Two commonly used performance measures are Gini impurity and entropy [43]. These two measures produce similar results with the choice made just based on preference. The feature importance is equivalent to the sum of the performance measures of the nodes within the single decision tree that contain the named feature. The importance of the feature for the GBDT model is then computed as the average across all of trees [44].

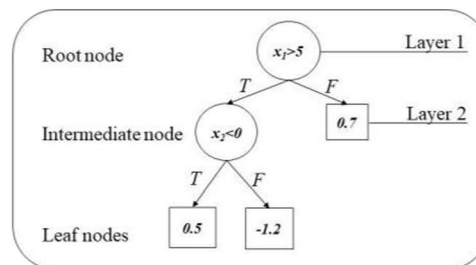
The GBDT will be utilised twice in the predictive tool, first for facilitating determining the right level of time lags for the DPCA followed by the feature generation step and later for achieving a production failure prediction model.

Fig. 2 Structure of the gradient boosting decision trees

A Gradient Boosting Decision Trees Structure



A Decision Trees Structure



3.3 The proposed production failure predictive tool for CPPS

The procedure of developing and deploying the proposed predictive tool is illustrated in Fig. 3a, b. Segments of historical data are extracted from the cloud database and are used for the offline analysis to train a predictive model for production failures. The predictive model is then fed with a set of unseen new data to foretell real-time production failures in the future. A more in-depth description of each of the steps is presented in this section.

3.3.1 Marking production stops and labelling failure events

A production stop can be identified according to a combination of measured variables which reflect the actions of the

machine operator: the program is paused, the machine operation goes to the manual operating mode and the operator door is opened for more than 1 min. However, the production failure (such as tool wear and tool breakage) happens before the operator stops the process. The machine is undergoing a certain critical status between a normal operation status and an established production stop, which is manually switched to by an operator after having observed unhealthy signals from the industrial computer or other physical characteristics of the machine tool. It is the critical status that is of our interest to predict. The measurements taken during the critical status will be labelled as failure events of the target variable.

An operator stops the process based on his/her experience by noticing the change in the chip formation (colour, shape, sparks), noise level or vibration level. The monitoring system is not capturing information about the chip formation or the

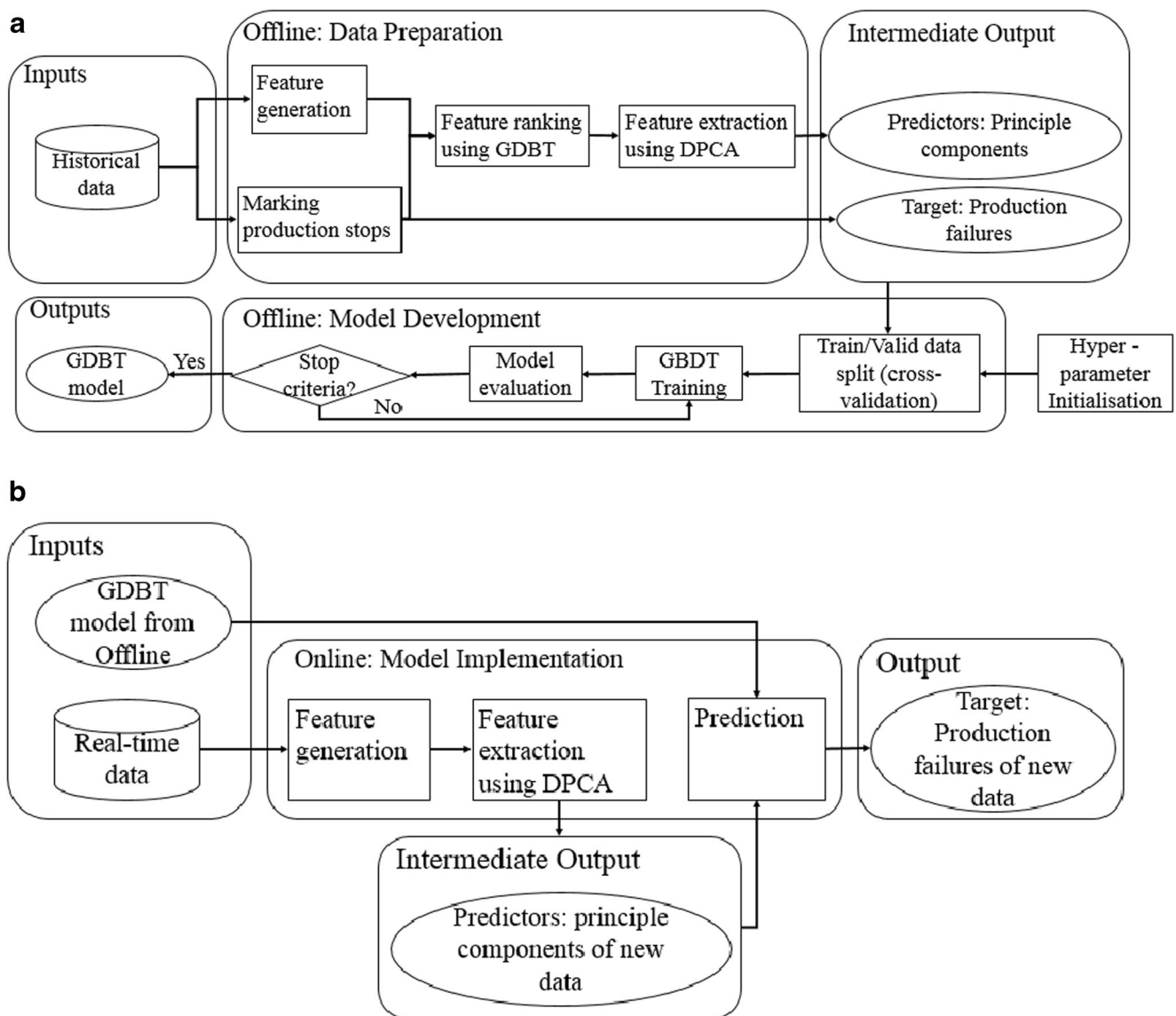


Fig. 3 a The offline model development for the predictive tool. b The online model deployment for the predictive tool

machining noise, but both the vibration and the spindle power are sensitive to the degradation of the cutting conditions. If the operator is next to the machine when the failure event occurs, he/she can stop in only a few seconds whereas it can take longer if he/she is less attentive to the cutting process or further from the control panel. The actual time varies but based on the experience of operators working on the shop floor in the experimental factory, an estimation of 60-s window is assumed for the duration of the critical status to cover the reaction time of a human operator to notice a potential production failure and the action time to switch off the machine tool or adjust the production pattern.

3.3.2 Feature generation

In addition to the input variables pulled from the cloud database, at the offline stage, new features are generated to include as much information as possible while keeping the past information in a compact way to assist in improving the performance of a prediction model. One classical approach is to use the rolling summary statistics of the physical variables, such as maximum, minimum, average, variance and skewness within a specified time period [45]. In this research piece, the maximum, average and variance values within 10-s, 30-s and 60-s windows are computed for the variables measured by the CNC and accelerometers.

3.3.3 Feature ranking using GBDT

The DPCA in the next step requires the determination of time lags, and the GBDT is first employed here to facilitate deciding the choice of the time lags. Using the generated feature space and the labelled production failures from the last steps, the importance score of the feature can be computed as a result of the constructed GBDT. By investigating importance score of generated features, the time window with the most highly important features will be chosen as the l time lags to shift the feature space to implement the DPCA.

3.3.4 Feature extraction using DPCA

In order to eliminate the auto-regression within a variable, up to l -seconds lag shifted features are generated and added to the newly generated feature space from the last step to perform DPCA analysis. Applying the PCA to the l -time lag shifted new feature space will eliminate features that are highly correlated and extract only ones that are orthogonal to each other while keeping the maximum variance of the new feature space. A number of principal components are determined to keep 85% of the total variance. The selected principal components are then used as inputs to train the GBDT model. This reduced-sized feature space can help to reduce the online analysis time and thus to detect potential production failures more

rapidly. However, one disadvantage of the DPCA is that, due to the time lag shifting and linear transformation (PCA) applied to the feature space, the resulted principal components will lose the physical interpretability.

3.3.5 GBDT model construction

The principal components achieved after the DPCA together with the corresponding target variable are split into a training part which is used to train the GBDT model and a validation part used to assess the performance of the trained model in order to avoid the overfitting problem (the model performs well on the training set, but poorly on the test set).

We will use the k -folder cross-validation technique for the training-validation separation. For k -folder cross-validation, the dataset is partitioned into k equal-size subsets. For the k subset, one set is retained for the model testing and the left $k-1$ sets are used for model training. The process repeats for k times and an average of all the k results are used to estimate the performance of the model. Considering the time-related nature of the data, the dataset is partitioned following a temporary manner. All transformed data from the previous steps is divided into k equal-size subsets, and the GBDT model that with the ‘best’ performance is trained.

The performance of the GBDT model can be justified by specified metrics. Since the failure events constitute merely a small portion of the target variable, it is sensible to choose a measuring metric that is diagnostic to the imbalanced nature of target variable. The area under the curve (AUC) is employed as the metric to evaluate the performance of the GBDT model during the model training on the historical data and to evaluate the performance of the prediction model on unseen data. In order to better explain the metric, the following definitions are introduced:

- True positive (TP): number of correct predictions for failure events.
- True negative (TN): number of correct predictions for non-failure events.
- False positive (FP): number of incorrect predictions for failure events.
- False negative (FN): number of incorrect predictions for non-failure events.

The AUC measures the area under the receiver operating characteristic (ROC) curve (as depicted in Fig. 8), which is a plot of the true positive rate (sensitivity), calculated as $TP/(TP + FN)$ against false positive rate ($1 - \text{specificity}$), calculated as $FP/(TN + FP)$. It is a plot which illustrates the diagnostic ability of a binary classifier as the threshold to discriminate the positive and negative events is varied. The diagonal dash line is a benchmark indicating relationship between the true positive rate and the false positive rate of a randomly

generated model. The closer a curve to the upper left corner and thus the closer to value 1 the AUC, the better the model performs. The best possible prediction model is yield in the upper left corner marked by the coordinate (0, 1) representing 0 false positive and 100% true positive rate. The AUC is equivalent to the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one [46].

The iteration of the GBDT algorithm stops if the measuring metric; i.e. the AUC score is not improved in 100 rounds or the maximum number of iterations is reached.

4 Case study: a milling process

4.1 The experimental setting

The performance of the proposed predictive tool is tested and demonstrated in a milling process. The experimental setup is organised following the cyber-physical production systems (CPPS) design presented in Section 2.

Figure 4 shows the hardware architecture that has been implemented to monitor the machining process. The milling machine is operated by a Siemens 840D numerical controller that offers a data service interface to read data from the CNC. An industrial computer has been added to the machine to pull the data every second from the CNC and send it to the cloud monitoring platform. This industrial computer is also connected through a Modbus communication to a real-time controller that processes the vibration measurements and outputs the vibration indicators (vibration severity, highest peaks of the vibration spectrum, etc.) every second. Finally, the historical data stored in the cloud is downloaded to a local machine that uses R programming language to train the GBDT model. The historical data contains 17 variables used as initial inputs variables; the explanation of each of the variable is presented in Table 2 in Appendix 2. In addition to these variables, the historical data also contains information about the machine operation mode (manual or machine operation), operator door mode (open or close) and program block number, which are used to deduce the production stops and then discarded after completing labelling the failure events. The trained GBDT model will be implemented on the industrial computer to forecast production failure events and provide potential production stop alerts in advance.

On the shop floor, the machine tool is operated in two shifts per day and always runs the same four machining programs to complete each work piece using 10 different types of tools. Figure 5 shows a schematic view of the work piece and a selection of the most problematic cutting tools. The total machining time by work piece is longer than 40 h. 40 days of machining operations are measured and used as the historical dataset. The program and cutting conditions are frozen, hence

the main source of process variation comes from the mechanical properties of the raw work piece and inserts that affect the tool wear. To minimise the risk of scrapping an expensive part, it is desirable to have a system which can stop the cutting process automatically when a failure is about to occur.

Figure 6 shows a typical vibration pattern obtained by superposing several executions of the machining program. It is not trivial for a human expert to identify a faulty behaviour based on a combined information of all the variables and it is impractical to set upper and lower boundaries for each variable along the 40 h of machining of the work piece to define the normal machining behaviour.

Using the criteria described in Section 3.3.1, production stops are marked for this process. The planned stops such as tool changing time and shifts between two-part programs, the starting and shutting-down periods of the machine tool (the first and last 5 min of a day) and small stops (within 1 min) are excluded from the unplanned production downtime. The proposed strategy to label the failure events tends to ensure that the instants marked as ‘failure’ are really representing problematic machining conditions. However, the learning algorithm can be perturbed by the fact that other problematic machining conditions are not marked. For example, large vibrations can appear during more than a minute if the operator is not able to stop the process rapidly or the machine can be stopped due to totally external reasons not related to the cutting process.

4.2 Feature generation and relative importance

The raw data set pulled from the cloud server contains records measured by 17 physical components every second (as presented in Table 2 in Appendix 2). The rolling statistics, i.e. mean, variance and maximum values within the 10-s, 30-s and 60-s time windows, are computed for all the 17 variables respectively.

The enriched features are normalised so that each feature variable follows a normal distribution with mean 0 and variance 1 using the formula: $z = \frac{x - \min(x)}{\max(x) - \min(x)}$, where x stands for the feature variable.

The GBDT is first utilised to achieve the relative importance of the newly generated features together with the original features in order to determine the best number of time lags for the DPCA. (Due to the high correlation among the features and auto-correlation within each feature, the prediction power of this first GBDT model is not ideal. Therefore, the DPCA has been deployed in the next step to eliminate the auto-correlations in order to achieve a better-performed predictive model.)

The features with the top 20 relative importance scores are shown in Table 3 in Appendix 3. The table can be utilised to provide a general idea of feature significance in order to assist

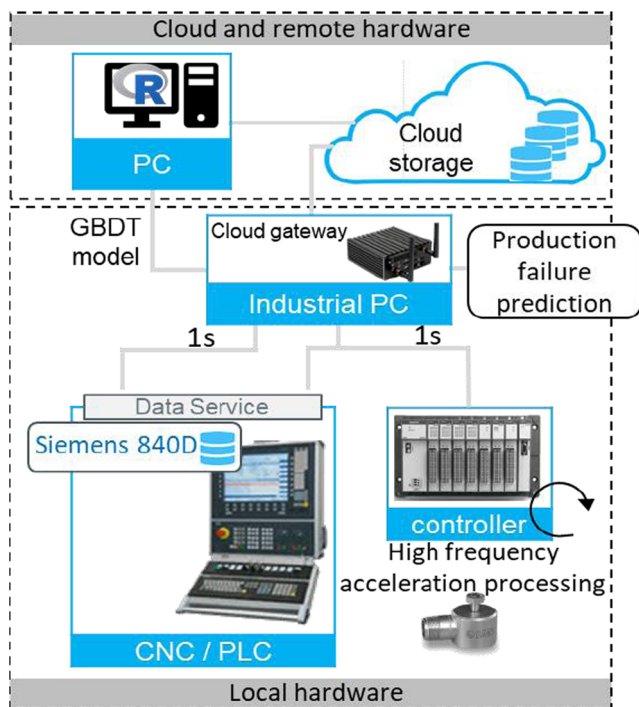


Fig. 4 Hardware and software architecture

allocating sensors to measure the most relevant machining properties in the future and most importantly, to facilitate determining the choice of the time lags in the DPCA. It is represented that the maximum vibration amplitude peak within 60 s and 30 s, and the maximum vibration severity from 20 to 80 Hz on the *y*-axis within 60 s plays the most important roles in predicting production failures of this case study. From the ten highest peaks of the vibration spectrum (40 data per second), the most significant elements correspond to the maximum peak in *x*- and *y*-directions. As the machine is more flexible in the *y*-direction, the second highest peak in the *y*-direction is also part of the relevant process variables. The rest of the frequency spectrum data are useful for the diagnostics of the process as they permit distinguishing the appearance of chatter vibrations, but they are not fed into the machine learning algorithm to reduce the size of the training dataset. Each

tool is used with a specific programmed spindle speed hence the programmed spindle speed gives redundant information. However, the measured spindle speed varies slightly during the machining operation due to the cutting forces. Large spindle speed variations can be caused by the modification of the tool work piece engagement or by a broken cutting edge. The spindle power also gives important information about the cutting process. As for the spindle speed, the variation of the spindle power can be related to the normal entry and exit in the material or to a broken tool. Considering the configuration of the machine and the shape of the part, the *x*-axis is almost not moving. Indeed, the feed movement is mainly given by the rotation of the B-axis hence it makes sense that the position of the *x*-axis does not lie within the group of the most relevant variables. The acceleration jumps are measured in the raw temporal signal. Large acceleration jumps are characteristics of the impacts of rapid machine movements. The most significant variables (e.g. variation amplitude in 30 s and 60 s, the max Y20–80 Hz in 60 s) to predict the production failures are not related to a specific part program, tool number or block number and therefore makes the predictive tool adaptive and flexible enough to be applied even if the machining program is modified.

Table 3 has illustrated that majority of the high ranked features are rolling statistics of the 60-s time window (12 out of 20). Therefore, from 1 s up to 60 s, time lag shifted feature space is generated for the implementation of DPCA.

4.3 Feature extraction using DPCA

A software has been developed using an open-source statistical language, R, to implement the predictive tool. An R package named ‘freedom’ [47] is utilised to extract principal components of the processed time series in order to eliminate the auto-correlation within a variable and to reduce the dimension of the new feature space and the GBDT models are developed using the R package named XGBoost [41, 48].

The implementation of the DPCA requires premise of a properly defined hyper-parameter-the maximum time lag to

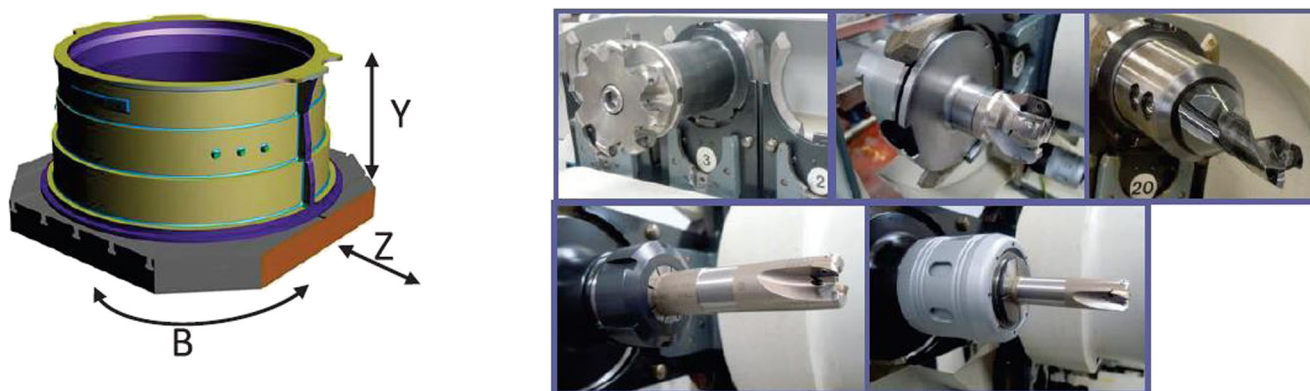


Fig. 5 Graphic representation of the work piece and different types of the problematic cutting tools

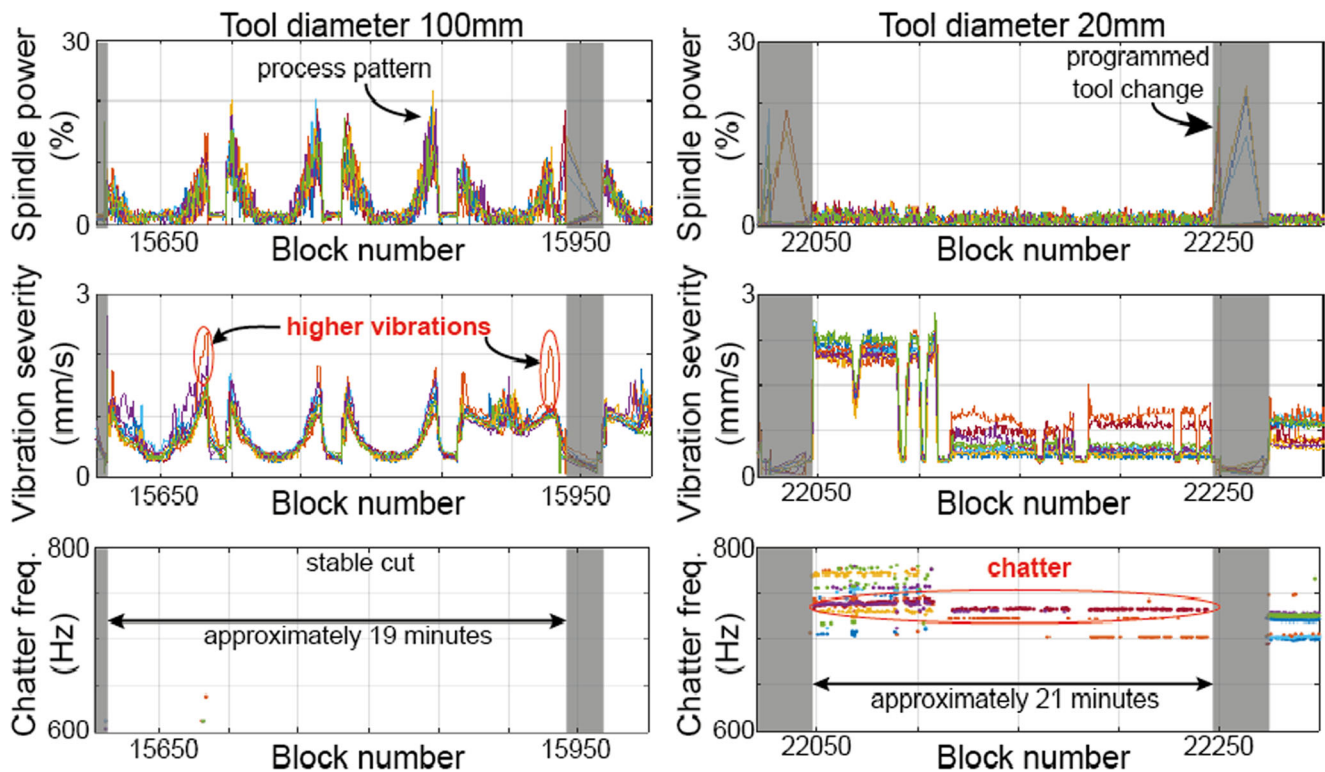


Fig. 6 Patterns of the experimental milling processes

shift the feature space. Employing the GBDT in the last step to rank the relative importance of generated summary statistics features has assisted to determine that the 60-s time window is the most relevant time lag to use. Principal components are extracted from the normalised 60-s time lagged shifted feature space and the number of principal components is determined such that 85% of the variance is retained. The resulted principal components space will have much lower dimensions compared with the time-shifted feature space. Unfortunately, due to the disadvantage of inexplicability of the resulted principal components, the inputs for the GBDT model do not represent any physical components properties or measurements and are thus lack interpretability. Finally, the input space composed of 82 principal components that are selected by the DPCA is used to train the GBDT model.

4.4 The development of the GBDT model

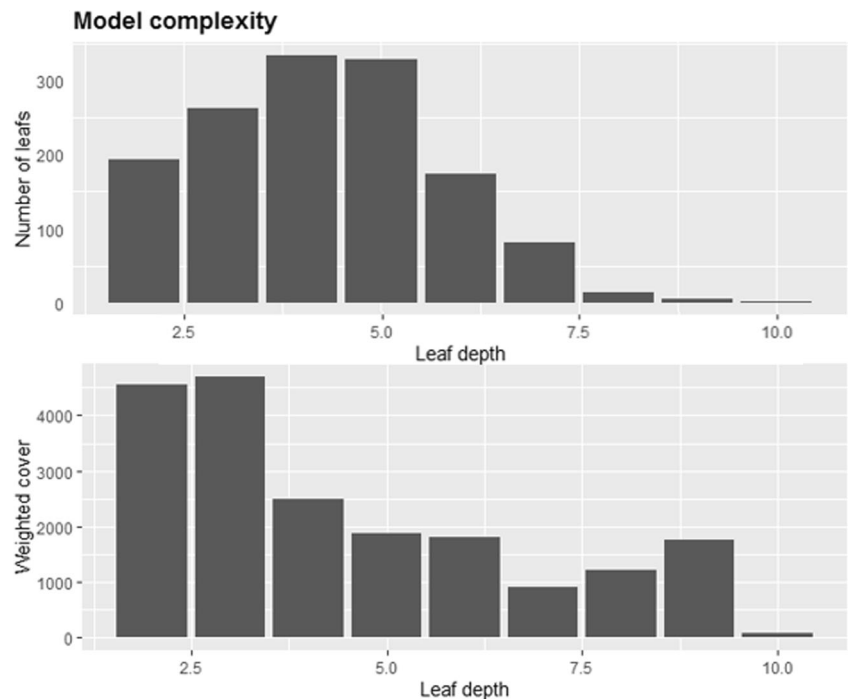
XGBoost is an efficient and scalable implementation program of GBDT that supports a wide range of built-in and self-defined objective functions (both regression and classification). It has proven to consume much fewer resources and perform 10 times faster than other GBDT packages by performing parallel computation automatically [41].

Data over 40 days are transferred and stored in an Amazon CouchDB database [35]. This data is then extracted by the developed software to train the predictive model. Initially, a set of hyper-parameters are defined to start a 4-folder cross-

validation as a premise to train a GBDT. The predefined hyper-parameters are as follows: the maximum number of decision trees, 10,000 (an upper bound of the number of decision trees in the GBDT), maximum decision tree depth, 10 (a stopping rule for constructing a single decision tree), and the learning rate, 0.005 (scale of the new tree when it is added the current decision tree sequence), minimal child weight, 60 (minimal number of observations after a node split). In this case study, we will only consider predicting the production failures. Thus, the objective of the GBDT model is to learn a binary classification model that could predict the probability of a failure event. The performance of the model is evaluated using the AUC score on the validation set during the cross-validation.

The GBDT algorithm (as presented in Algorithm 1 in Appendix 1) stops the iteration if the evaluation metric value has not improved in the next 100 rounds or the maximum number of iterations is reached. Training the GBDT model takes 340 iterations and the best model is obtained at iteration number 240 when the GBDT model has achieved a mean of 0.96 AUC score with 0.005 standard deviation on the training sets and 0.91 AUC score on the validation sets with 0.013 standard deviation. The model complexity is shown in Fig. 7, with the upper plot illustrating the distribution of the number of leaf nodes at a certain depth and the lower plot depicting the distribution of the average weighted number of cover (i.e. observations) terminated at a certain depth. The two-model complexity has facilitated the choices of the maximum depth

Fig. 7 Model complexity facilitating determining the choice of the maximum depth of the decision trees and the minimum child weight hyper-parameters



and the minimum child weight hyper-parameters mentioned in the last paragraph.

4.5 The online testing results

The developed GBDT is then deployed on the industrial computer to predict potential production failures. The fed real-time data of 21 days comes from both the CNC machine and the high-frequency controller (as illustrated in Fig. 5). The two segments of data coming from the low-frequency sensors and high-frequency acceleration signals are joined together as the original inputs for the online predictive tool.

Due to the quality of the sensors and time used to transfer the signals and to record the measurement from the machine tool to CNC, there are less than 5% of records missing from measurement read from each of the CNC indicators. The missed value is filled with the last available value in the same variable. Measurements taken multiple times within the same second are averaged in order to keep only one record per second. The first 60 records containing unavailable values resulted from time lagging the features are deleted.

The GBDT model returns probabilities as output to distinguish failure events and non-failure events. The threshold of 0.2 is utilised to separate a non-failure and a failure event. This is due to the non-failure events that constitute 95% of all records in the dataset that is used to train the GBDT model and 0.2 is the 95th percentile of the predicted probabilities. The ROC curve of the GBDT model is shown in Fig. 8. The ROC shows the relationship of the sensitivity (true positive rate) against 1 – specificity (false positive rate). The accuracy

of the predictive tool measured by the AUC value for the unseen data is 73%.

Considering the production period shown in Fig. 9a, where the blue line represents the accumulated program block number, a production stop can be located starting from 17:26 and ending at 17:29 and another one starting from 17:50 and ending at 17:54, as the block number does not change and thus reflecting a production stop. The 60-s time interval before the stop starting points is marked by the red blocks. The

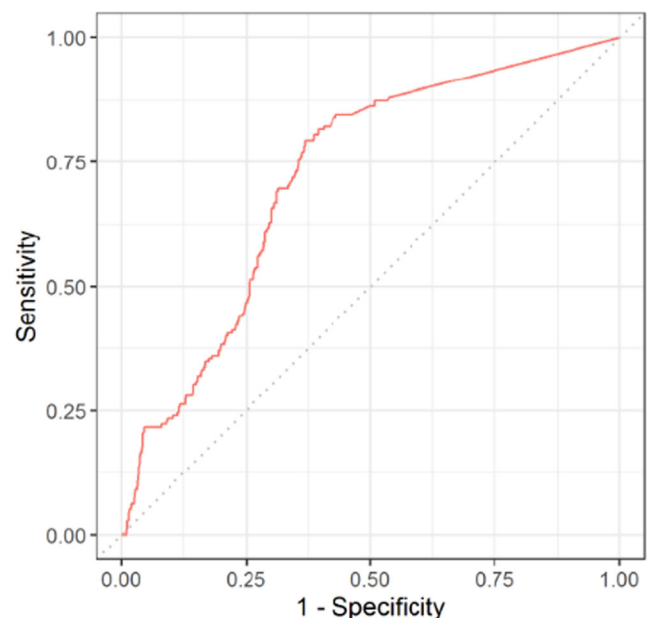


Fig. 8 Receiver operating characteristic (ROC) curve of the obtained GBDT model

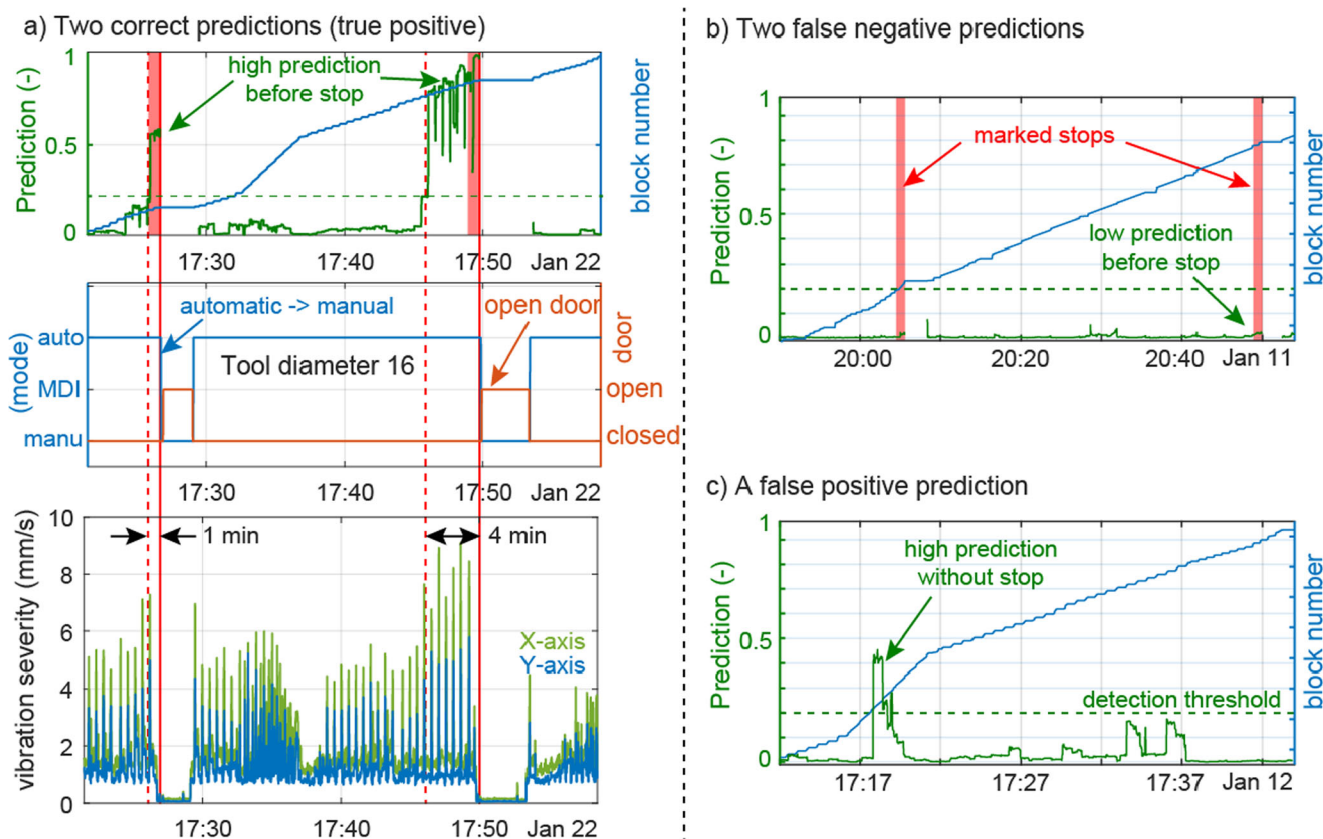


Fig. 9 Example production stops in the experimental CPPS. **a** Two correct predictions (true positive). **b** Two false negative predictions. **c** A false positive prediction

production failure might be associated with the increase of vibration severity before the stop.

It is not trivial even for an expert to identify the deviations from normal behaviours due to the large timeframe of the production and the large number of measured parameters. Indeed, several correct failure detections cannot be explained by the detailed analysis of the data by the expert. Unfortunately, the GBDT is also generating false negative and false positive detections as shown in Fig. 9b, c. However, most of the predictions correspond to true negative events given the chosen threshold.

The prediction precision of 73% implies that most of the production failures can be predicted accurately. Thus, it is believed that the cyber-physical model can be used to let the machine unattended while protecting the expensive parts. In that case, the predictive tool would automatically raise alerts of potential production failures and allow the operator to take appropriate countermeasures such as changing the cutting edges or pausing the production process.

5 Conclusion and future work

This paper has introduced a framework of CPPS, which gave a detailed description of the data acquisition mechanism from

the shop floor equipment, the data storage solution in a cloud-based database and data analytics that are used for process monitoring. Thereafter, a predictive tool as part of the CPPS has been proposed. The predictive model has utilised the dynamic principal component analysis (DPCA) to extract the auto-correlation within the same variable and to reduce the dimensionality of the input space such that only the inputs that contain the most meaningful information are retained for training the GBDT model. The GBDT is a powerful supervised machine learning algorithm that could deal with large-scale datasets with high efficiency. It has used the boosting ensemble technique by building a sequence of decision trees to achieve better prediction results. A classification GBDT model was developed to predict the production failure and has been tested on a real-world CPPS performing a machining process.

The GBDT has been utilised twice in the predictive tool earlier on the offline stage for facilitating determining a preferred time lag to shift the feature space as a premise of implementing the DPCA and later on the online stage to be trained properly as a classifier to forecast potential production stops. Considering the approach taken to separate the events into failure and non-failure classes and the fact that the failure events constitute merely 5% of all events in the training dataset, it is wise to choose a measuring metric

for the developed classification predictive model that is meanwhile suitable for imbalanced data. In this research piece, we have chosen the area under the ROC (AUC) score as the measuring metric. Testing results have shown the predictive tool can achieve 73% accuracy using the AUC score when foretelling the failure events on an unseen dataset. Despite the disadvantage of abolishing the physical meaning of a feature, the DPCA can eliminate the auto-correlation within a variable and thus enhance the performance of the following GBDT model and largely reduce the dimension of the time lag feature space and thus make the prediction process more efficient. An accurate and efficient productive tool makes it possible for a self-aware and self-adaptive CPPS that is able to adjust the production patterns before a failure event occurs, and therefore significantly reduce the downtime and associated production cost.

The main contribution of this paper lies in the introduction of a predictive tool by employing both the statistical method, DPCA, and the machine learning model, GBDT, and implementing the predictive tool to predict the production failures of a complex machining operation in a real-world CPPS.

The following aspects can be further explored in future work:

- The developed predictive tool can be further extended to diagnose varying reasons that lead to a production stop such as tool breakage, over-heating, high vibration. The predictive tool in this manuscript has deployed a binary target variable including only failure and non-failure events. Instead of a two-class target, it is possible to label the target as multiple categories by means of recognising and recording the manufacturing failure that lead to the production stop. For example, if the production stop is caused by the tool breakage, then the target variable in the critical stage will be labelled as ‘tool wear’; if it is the over-heating that result in the shut-down of the manufacturing process, the target variable will be categorised as ‘high temperature level’. The choices of the manufacturing failure can differ depending on the experimental setting scenario and thus require manual initialisation as an input to train the GBDT model.
- As described in Section 3.3.2, observations within the 1 min prior to an established production stop are labelled as the failure events of the target variable. It is assumed that before entering the established stop status, the manufacturing process is undergoing a critical status from the normal production status. It is the critical status of the manufacturing process that the predictive tool is forecasting. However, the 1-min time interval has been chosen by the subject matter experts (SMEs) based on their

experience of manoeuvring machine tools, whereas the actual critical status can go longer or shorter under different cases. Therefore, it will require a more scientific way to support or improve this judgemental decision. One possible approach is to utilise an unsupervised model (a type of model that only contains inputs variable) to classify the machine status to several different clusters using the inputs variables. The number of clusters can be manually determined based on SMEs’ experience (e.g. it is sensible to assume that the output the unsupervised model accommodate 3 categories which include the normal status, the critical status and the stop status of the machining tool) or by a data-driven method (such as the elbow method), in which case, the number of clusters will be determined numerically. Since an unsupervised model trains a model without using labelled output, it does not need to bother to mark production stops and label records in the critical status judgementally. Meanwhile, by attempting a different type of model, it is possible to further enhance the prediction power of the predictive tool with a 73% accuracy level.

Acknowledgements The authors are grateful to the two anonymous reviewers for their valuable comments to help improve the quality of the manuscript.

Funding This research work is conducted as part of the H2020 European project MC-SUITE (Grant No. H2020-2015-680478) and is partially funded by the MIRAGED project (CER-20191001 CDTI-CERVERA).

Data availability Please see the shared Github repository ([GitHub Link](#)).

Compliance with ethical standards

Code availability Please see the shared Github repository ([GitHub Link](#)).

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

Appendix 1: GBDT algorithm for a classification problem

With a K-class classification problem, the target variable Y takes values in a set $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_K\}$. Within the data science applications, the interest often lies in finding the class probability, $p_k(x)$, $k = 1, 2, \dots, K$ represented by a logistic function as shown in Eq. (2) [44]:

$$p_k(x) = \frac{e^{F_k(x)}}{\sum_{l=1}^K e^{-F_l(x)}} \quad (2)$$

which ensures that $0 \leq p_k(x) \leq 1$ and $\sum_{l=1}^K p_k(x) = 1$.

Suppose $F(x)$ is the function/classifier that is learned using the GBDT algorithm, the fundamental schema for the GBDT algorithm is then represented in Algorithm 1.

Algorithm 1: Gradient Boosting Decision Tree for Classification [44]

1. Initialise $F_{k0}(x) = 0, k = 1, 2 \dots K$.
 2. For $m = 1$ to M , where M is the maximum iteration number.
 - a. Set $p_k(x)$ as represented by Equation 2.
 - b. For $k = 1, 2 \dots K$:
 - i. Compute $r_{ikm} = y_{ik} - p_k(x_i), i = 1, 2 \dots n$ (n is the total number of observations).
 - ii. Fit a decision tree for the new target γ_{ikm} giving each terminal region, $R_{jkm}, j = 1, 2 \dots J_m$.
 - iii. Compute

$$\gamma_{jkm} = \frac{K - 1}{K} \frac{\sum_{x \in R_{jkm}} r_{ikm}}{\sum_{x \in R_{jkm}} |r_{ikm}|(1 - |r_{ikm}|)}, \quad j = 1, 2 \dots J_m$$
 - iv. Update $F_{km}(x) = F_{k,m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jkm} I(x \in R_{jkm})$, where $I(x \in R_{jkm})$ is the index function.
 3. Return $\widehat{F}_x(x) = F_{kM}(x), k = 1, 2 \dots K$
-

Appendix 2: Feature names and explanation measured by sensors

Table 2 Feature names used as original inputs pulled from the cloud server

Feature name	Explanation	Measured by	Unit
ToolFactor	The type of the cutting tool	CNC	-
Spindle_speedActual	Actual spindle speed	CNC	rpm
Cnc_Override_Axis	Override percentage to control the spindle speed	CNC	%
Axis_FeedRate_actual	Actual velocity at which the cutter is advanced against the work piece	CNC	mm/min
Axis_B_positionActualMCS_deg	Actual position of the rotation degree of the work piece	CNC	degree
Axis_Y_positionActualMCS_mm	Actual position of the work piece along the y -axis	CNC	mm
Axis_Z_positionActualMCS_mm	Actual position of the work piece along the z -axis	CNC	mm
Spindle_Power	Spindle power	CNC	kw
IC3_X_Accel_Max_Discontinuity	Acceleration jumps along the x -axis	Accelerometer	mm/s ²
IC3_X_FFT_Peak_Amplitude_1	Peak vibration amplitude at range 20 Hz–80 Hz along the x -axis	Accelerometer	mm/s
IC3_X_Severity_range_1	Vibration severity at range 20 Hz–1000 Hz along the x -axis	Accelerometer	mm/s
IC3_X_Severity_range_2	Vibration severity at range 20 Hz–80 Hz along the x -axis	Accelerometer	mm/s

Table 2 (continued)

Feature name	Explanation	Measured by	Unit
IC3_Y_Accel_Max_Discontinuity	Acceleration jumps along the y-axis	Accelerometer	mm/s ²
IC3_Y_FFT_Peak_Amplitude_1	Peak vibration amplitude along the y-axis	Accelerometer	mm/s
IC3_Y_FFT_Peak_Amplitude_2	2nd peak vibration amplitude along the y-axis	Accelerometer	mm/s
IC3_Y_Severity_range_1	Vibration severity at range 20 Hz–80 Hz along the y-axis	Accelerometer	mm/s
IC3_Y_Severity_range_2	Vibration severity at range 20 Hz–1000 Hz along the y-axis	Accelerometer	mm/s

Appendix 3: Features with the top 20 importance scores ranked by the GBDT

Table 3 Features with the top 20 importance scores ranked by the GBDT

Rank	Feature name	Importance score	Statistics features (60 s)	Statistics features (30 s)	Statistics features (10 s)	Original
1	IC3_Y_FFT_Peak_Amplitude_1, max60	0.08	✓			
2	IC3_Y_Severity_range_2, max60	0.07	✓			
3	IC3_Y_FFT_Peak_Amplitude_1, max30	0.06		✓		
4	Axis_B_positionActualMCS_deg, ampli60	0.04	✓			
5	Axis_X_positionActualMCS_mm, max60	0.04	✓			
6	IC3_Y_Severity_range_2, max30	0.04		✓		
7	Axis_B_positionActualMCS_deg, ampli30	0.04		✓		
8	IC3_Y_FFT_Peak_Amplitude_2, max60	0.04	✓			
9	Axis_B_positionActualMCS_deg	0.03				✓
10	IC3_Y_Accel_Max_Discontinuity, max60	0.03	✓			
11	Spindle_speedActual, ampli60	0.03	✓			
12	Axis_B_positionActualMCS_deg, ampli10	0.03			✓	
13	Axis_Y_positionActualMCS_mm, aver60	0.03	✓			
14	Axis_Y_positionActualMCS_mm, max60	0.03	✓			
15	Axis_FeedRate_actual, ampli60	0.03	✓			
16	IC3_X_FFT_Peak_Amplitude_1, max60	0.02	✓			
17	IC3_Y_Accel_Max_Discontinuity, max30	0.02		✓		
18	IC3_X_Severity_range_1, max30	0.02		✓		
19	Axis_Y_positionActualMCS_mm, aver30	0.02		✓		
20	IC3_X_Accel_Max_Discontinuity, max60	0.01	✓			

Explanation of suffix: *max60* maximum value within 60 s, *ampli60* variance within 60 s, *aver60* mean within 60 s, *max30* maximum value within 30 s, *ampli30* variance within 30 s, *aver30* mean within 30 s, *max10* maximum value within 10 s, *ampli10* variance within 10 s, *aver10* mean within 10 s

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

References

- Cruz Salazar LA, Ryashentseva D, Lüder A, Vogel-Heuser B (2019) Cyber-physical production systems architecture based on multi-agent's design pattern—comparison of selected approaches mapping four agent patterns. *Int J Adv Manuf Technol* 105:4035. <https://doi.org/10.1007/s00170-019-04226-8>
- Severson K, Chaiwatanodom P, Braatz RD (2016) Perspectives on process monitoring of industrial systems. *Annu Rev Control* 42: 190–200. <https://doi.org/10.1016/j.arcontrol.2016.09.001>
- Caggiano A (2018) Cloud-based manufacturing process monitoring for smart diagnosis services. *Int J Comput Integr Manuf* 31:612–623. <https://doi.org/10.1080/0951192X.2018.1425552>
- Gao R, Wang L, Teti R et al (2015) Cloud-enabled prognosis for manufacturing. *CIRP Ann* 64:749–772. <https://doi.org/10.1016/j.CIRP.2015.05.011>
- Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN (2003) A review of process fault detection and diagnosis: part I: quantitative model-based methods. *Comput Chem Eng* 27:293–311. [https://doi.org/10.1016/S0098-1354\(02\)00160-6](https://doi.org/10.1016/S0098-1354(02)00160-6)
- Gao Z, Dai X, Breikin T, Wang H (2008) Novel parameter identification by using a high-gain observer with application to a gas turbine engine. *IEEE Trans Ind Inf* 4:271–279. <https://doi.org/10.1109/TII.2008.2007802>
- Stavropoulos P, Chantzis D, Doukas C, Papacharalampopoulos A, Chrystoulouris G (2013) Monitoring and control of manufacturing processes: a review. *Procedia CIRP* 8:421–425. <https://doi.org/10.1016/J.PROCIR.2013.06.127>
- Yin S, Ding SX, Xie X, Luo H (2014) A review on basic data-driven approaches for industrial process monitoring. *IEEE Trans Ind Electron*.
- Campos-Delgado DU, Espinoza-Trejo DR (2011) An observer-based diagnosis scheme for single and simultaneous open-switch faults in induction motor drives. *IEEE Trans Ind Electron* 58:671–679. <https://doi.org/10.1109/TIE.2010.2047829>
- Akrad A, Hilairet M, Diallo D (2011) Design of a fault-tolerant controller based on observers for a PMSM drive. *IEEE Trans Ind Electron* 58:1416–1427. <https://doi.org/10.1109/TIE.2010.2050756>
- Huang S, Tan KK, Lee TH (2012) fault diagnosis and fault-tolerant control in linear drives using the Kalman filter. *IEEE Trans Ind Electron* 59:4285–4292. <https://doi.org/10.1109/TIE.2012.2185011>
- Gertler J (1998) Fault detection and diagnosis in engineering systems. CRC Press, New York
- Ding SX (2013) Model-based fault diagnosis techniques. Springer London, London
- Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2016) Introduction to diagnosis and fault-tolerant control. In: Biagnosis and fault-tolerant control. Springer Berlin Heidelberg, Berlin, pp 1–35
- Das A, Maiti J, Banerjee RN (2012) Process monitoring and fault detection strategies: a review. *Int J Qual Reliab Manag* 29:720–752. <https://doi.org/10.1108/02656711211258508>
- Wise BM, Gallagher NB (1996) The process chemometrics approach to process monitoring and fault detection. *J Process Control* 6:329–348. [https://doi.org/10.1016/0959-1524\(96\)00009-1](https://doi.org/10.1016/0959-1524(96)00009-1)
- Dong D, McAvoy TJ (1996) Nonlinear principal component analysis—based on principal curves and neural networks. *Comput Chem Eng* 20:65–78. [https://doi.org/10.1016/0098-1354\(95\)00003-K](https://doi.org/10.1016/0098-1354(95)00003-K)
- Muradore R, Fiorini P (2012) A PLS-based statistical approach for fault detection and isolation of robotic manipulators. *IEEE Trans Ind Electron* 59:3167–3175. <https://doi.org/10.1109/TIE.2011.2167110>
- He X, Wang Z, Liu Y, Zhou DH (2013) Least-squares fault detection and diagnosis for networked sensing systems using a direct state estimation approach. *IEEE Trans Ind Informatics* 9:1670–1679. <https://doi.org/10.1109/TII.2013.2251891>
- Martínez-Arellano G, Terrazas G, Ratchev S (2019) Tool wear classification using time series imaging and deep learning. *Int J Adv Manuf Technol* 104:3647–3662. <https://doi.org/10.1007/s00170-019-04090-6>
- Dong J, Subrahmanyam KVR, Wong YS, Hong GS, Mohanty AR (2006) Bayesian-inference-based neural networks for tool wear estimation. *Int J Adv Manuf Technol* 30:797–807. <https://doi.org/10.1007/s00170-005-0124-8>
- Gao C, Jian L, Luo S (2012) Modeling of the thermal state change of blast furnace hearth with support vector machines. *IEEE Trans Ind Electron* 59:1134–1145. <https://doi.org/10.1109/TIE.2011.2159693>
- Kothuru A, Nooka SP, Liu R (2018) Application of audible sound signals for tool wear monitoring using machine learning techniques in end milling. *Int J Adv Manuf Technol* 95:3797–3808. <https://doi.org/10.1007/s00170-017-1460-1>
- Choi SW, Lee I-B (2004) Nonlinear dynamic process monitoring based on dynamic kernel PCA. *Chem Eng Sci* 59:5897–5908. <https://doi.org/10.1016/J.CES.2004.07.019>
- Choi SW, Lee C, Lee JM, Park JH, Lee IB (2005) Fault detection and identification of nonlinear processes based on kernel PCA. *Chemom Intell Lab Syst* 75:55–67. <https://doi.org/10.1016/j.chemolab.2004.05.001>
- Pöllänen K, Häkkinen A, Reinikainen SP, Rantanen J, Minkkinen P (2006) Dynamic PCA-based MSPC charts for nucleation prediction in batch cooling crystallization processes. *Chemom Intell Lab Syst* 84:126–133. <https://doi.org/10.1016/j.chemolab.2006.04.016>
- Ku W, Storer RH, Georgakis C (1995) Disturbance detection and isolation by dynamic principal component analysis. *Chemom Intell Lab Syst* 30:179–196. [https://doi.org/10.1016/0169-7439\(95\)00076-3](https://doi.org/10.1016/0169-7439(95)00076-3)
- Cui P, Li J, Wang G (2008) Improved kernel principal component analysis for fault detection. *Expert Syst Appl* 34:1210–1219. <https://doi.org/10.1016/J.ESWA.2006.12.010>
- Guo K, San Y, Zhu Y (2012) Nonlinear process monitoring using wavelet kernel principal component analysis. In: 2012 International Conference on Systems and Informatics, ICSAI 2012
- Rosipal R, Trejo LJ (2002) Kernel partial least squares regression in reproducing kernel hilbert space. *J Mach Learn Res* 2:97
- Rosipal R (2003) Kernel partial least squares for nonlinear regression and discrimination. *Neural Netw World*
- Guh R-S, Shiue Y-R (2008) An effective application of decision tree learning for on-line detection of mean shifts in multivariate control charts. *Comput Ind Eng* 55:475–493. <https://doi.org/10.1016/J.CIE.2008.01.013>

33. Godreau V, Ritou M, Chové E, Furet B, Dumur D (2018) Continuous improvement of HSM process by data mining. *J Intell Manuf* 30:1–8. <https://doi.org/10.1007/s10845-018-1426-7>
34. Teti R, Jemielniak K, O'Donnell G, Dornfeld D (2010) Advanced monitoring of machining operations. *CIRP Ann* 59:717–739. <https://doi.org/10.1016/J.CIRP.2010.05.010>
35. CouchDB - JSON based Web database powered by TurnKey GNU/Linux (HVM). In: AWS Marketpl. <https://aws.amazon.com/marketplace/pp/TurnKey-GNULinux-CouchDB-JSON-based-Web-database-p/B01KVGYDF0>. Accessed 15 Apr 2020
36. Troyansky O, Gibson T, Leichtweis C (2015) QlikView your business: an expert guide to business discovery with QlikView and Qlik Sense. Wiley, Hoboken
37. Hörmann S, Kidzi L, Hallin M (2015) Dynamic functional principal components. *J R Stat Soc Ser B Stat Methodol* 77:319–348. <https://doi.org/10.1111/rssb.12076>
38. Elith J, Leathwick JR, Hastie T (2008) A working guide to boosted regression trees. *J Anim Ecol* 77:802–813. <https://doi.org/10.1111/j.1365-2656.2008.01390.x>
39. Guelman L (2012) Gradient boosting trees for auto insurance loss cost modeling and prediction. *Expert Syst Appl* 39:3659–3667. <https://doi.org/10.1016/J.ESWA.2011.09.058>
40. Friedman JH Greedy function approximation: a gradient boosting machine. *Ann Stat* 29:1189–1232. <https://doi.org/10.2307/2699986>
41. Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, California, USA
42. Si S, Zhang H, Keerthi SS, et al (2017) gradient boosted decision trees for high dimensional sparse output. In: Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia
43. Will (2016) Decision tree flavors: Gini index and information gain. In: Learn by Mark. <http://www.learnbymarketing.com/481/decision-tree-flavors-gini-info-gain/>. Accessed 30 May 2020
44. Hastie T, Tibshirani R, Friedman JH, Jerome H (2009) The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer, New York
45. Duo A, Basagoiti R, Arrazola PJ, Aperribay J, Cuesta M (2019) The capacity of statistical features extracted from multiple signals to predict tool wear in the drilling process. *Int J Adv Manuf Technol* 102:2133–2146. <https://doi.org/10.1007/s00170-019-03300-5>
46. Fawcett T (2006) An introduction to ROC analysis. *Pattern Recogn Lett* 27:861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>
47. Hormann S, Kidzinski L (2017) Frequency domain based analysis: dynamic PCA. <https://cran.r-project.org/web/packages/freqdom/freqdom.pdf>. Accessed 31 May 2020
48. Chen T, He T, Michael B, et al Extreme gradient boosting. In: CRAN. <https://cran.r-project.org/web/packages/xgboost/xgboost.pdf>. Accessed 2 Sep 2020

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