Machine Learning for Predictive Maintenance: A Multiple Classifier Approach

5 authors, including:

Gian Antonio Susto
University of Padova
129 PUBLICATIONS   **1,896** CITATIONS

SEE PROFILE

Simone Pampuri
University of Pavia
22 PUBLICATIONS   **880** CITATIONS

SEE PROFILE

Sean Mcloone
Queen's University Belfast
213 PUBLICATIONS   **3,610** CITATIONS

SEE PROFILE

Alessandro Beghi
University of Padova
227 PUBLICATIONS   **3,178** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   PhD Maynooth View project

Project   Virtual prototyping of controlled systems View project

Machine Learning for Predictive Maintenance: a Multiple Classifier Approach

Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone *Senior Member, IEEE*, Alessandro Beghi *Member, IEEE*

Abstract—In this paper a multiple classifier machine learning methodology for Predictive Maintenance (PdM) is presented. PdM is a prominent strategy for dealing with maintenance issues given the increasing need to minimize downtime and associated costs. One of the challenges with PdM is generating so called 'health factors' or quantitative indicators of the status of a system associated with a given maintenance issue, and determining their relationship to operating costs and failure risk. The proposed PdM methodology allows dynamical decision rules to be adopted for maintenance management and can be used with high-dimensional and censored data problems. This is achieved by training multiple classification modules with different prediction horizons to provide different performance trade-offs in terms of frequency of unexpected breaks and unexploited lifetime and then employing this information in an operating cost based maintenance decision system to minimise expected costs. The effectiveness of the methodology is demonstrated using a simulated example and a benchmark semiconductor manufacturing maintenance problem.

Index Terms—Classification Algorithms, Data Mining, Ion Implantation, Machine Learning, Predictive Maintenance, Semiconductor Device Manufacture.

I. INTRODUCTION

The increasing availability of data is changing the way decisions are taken in industry [17] in important areas such as scheduling [15], maintenance management [24] and quality improvement [6], [23]. Machine learning (ML) approaches have been shown to provide increasingly effective solutions in these areas, facilitated by the growing capabilities of hardware, cloud-based solutions,

and newly introduced state-of-the-art algorithms. At the same time the efficient management of maintenance activities is becoming essential to decreasing the costs associated with downtime and defective products [14], especially in highly competitive advanced manufacturing industries such as semiconductor manufacturing.

Approaches to maintenance management can be grouped into three main categories which, in order of increasing complexity and efficiency [22], are as follows: i. *Run-to-Failure (R2F)* - where maintenance interventions are performed only after the occurrence of failures. This is obviously the simplest approach to dealing with maintenance (and for this reason it is frequently adopted), but it is also the least effective one, as the cost of interventions and associated downtime after failure are usually much more substantial than those associated with planned corrective actions taken in advance. ii. *Preventive Maintenance (PvM)* - where maintenance actions are carried out according to a planned schedule based on time or process iterations. With this approach, also referred to as scheduled maintenance, failures are usually prevented, but unnecessary corrective actions are often performed, leading to inefficient use of resources and increased operating costs.
iii. *Predictive Maintenance (PdM)*[1] - where maintenance is performed based on an estimate of the health status of a piece of equipment [7]. PdM systems allow advance detection of pending failures and enable timely pre-failure interventions, thanks to prediction tools based on historical data, *ad hoc* defined *health factors*, statistical inference methods, and engineering approaches.

Among statistical inference based methods, those based on ML are the most suitable for dealing with modeling of high-dimensional problems, such as those arising in semiconductor manufacturing where hundreds or thousands of physical variables (pressures, voltages, currents, flows, etc.) act on the process [10], [26].

In this paper a new PdM methodology based on multiple classifiers is introduced for *integral type faults* (the most frequent in semiconductor manufacturing), a term which describes the failures that happen on a machine due to the accumulative 'wear and tear' effects of usage

---

[1]Other authors [8] refer to this class of maintenance approaches as Condition-Based Maintenance

and stress on equipment parts. Even if no direct evidence of process/machine degradation is available, PdM tools exploit process and logistic variables collected during production to identify the 'footprint' of this degradation in the data. The proposed methodology, referred to as *Multiple Classifier (MC)* PdM, can effectively deal with the unbalanced datasets that arise in maintenance classification problems, that is datasets where the observations relating to normal production greatly outnumber the observations associated with abnormal/faulty production [22]. It also allows planning of maintenance schedules using a statistical cost minimization approach.

The remainder of the paper is structured as follows: Section II provides a brief literature review and an introduction to ML-based approaches to PdM, while Section III is dedicated to presenting the proposed MC PdM methodology. Tuning guidelines and classification approaches are discussed in Section IV. Then in Section V, to demonstrate the principles and efficacy of MC PdM, results are presented comparing MC PdM with PvM for a benchmark semiconductor manufacturing maintenance problem, namely, changing of filaments in ion implantation tools. Finally, in Section VI, concluding remarks and possible extensions of the work are discussed.

II. LITERATURE REVIEW

Maintenance issues can be completely different in nature and the predictive information to be fed to the PdM module has in general to be tailored to the particular problem at hand, thus justifying the presence in the literature of many different approaches to PdM [12]. However, PdM-related solutions based on ML techniques seem to be among the most popular (see e.g. [5] and [21] with reference to semiconductor manufacturing applications).

ML-based PdM can be divided into two main classes:
i. *supervised* - where information on the occurrence of failures is present in the modeling dataset;
ii. *unsupervised* - where logistic and/or process information is available, but no maintenance related data exists.

The availability of maintenance information mostly depends on the nature of the existing maintenance management policy: in the case of R2F policies the data related to a *maintenance cycle* (the production activity between two successive failure events) is available and therefore supervised approaches can readily be adopted; on the other hand, when PvM policies are currently in place the full maintenance cycle may not be observable given the fact that maintenance is generally performed well in advance of any potential failure, and hence only unsupervised learning approaches are feasible. When possible, supervised solutions are evidently preferable: given the wide diffusion of R2F maintenance policies in industry, and hence the availability of suitable datasets, we consider in this work a supervised approach to PdM.

From a ML perspective, supervised approaches require the availability of a dataset $\mathcal{S}$

$$\mathcal{S} = \{x_i, y_i\}_{i=1}^n, \tag{1}$$

where a couple $\{x_i, y_i\}$ (called *observation*) contains the information related to the i-th process iteration. Here, vector $x_i \in \mathbb{R}^{1 \times p}$ contains information related to the p variables associated with available process or logistic information. Depending on the type of output y two classes of supervised problem are possible:
i. if y assumes continuous values a *regression* problem is obtained;
ii. if y assumes categorical values a *classification* problem results.

In PdM problems, regression-based formulations generally arise when predicting the Remaining Useful Life of a process/equipment, directly [2] or indirectly (eg. through the computation of the conditional reliability [5]), while classification-based PdM formulations occur when seeking to discriminate between healthy and unhealthy conditions of the system being monitored [1].

While classification tools are a natural choice for distinguishing between faulty and non-faulty process iterations based on observed process data, they do not map naturally to health factors that can be extrapolated for maintenance related decision making, unlike for example, regression models of Remaining Useful Life. In the following a methodology based on multiple classifiers is presented to address this limitation.

III. MULTIPLE CLASSIFIER PDM

A. Classification for PdM

Let us suppose that data regarding N maintenance cycles are available, for a total of $n = \sum_{i=1}^N n_i$ machine runs. We define the matrix $X \in \mathbb{R}^{n \times p}$ containing all the collectible information

$$X = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix}^T \tag{2}$$

regarding p physical variables or logistic information on the process or on the tool that are used as inputs to the PdM module. Information on maintenance events is contained in the variable Y. If during the run the fault under consideration takes place, then the observation is marked as *faulty (F)*, and *not faulty (NF)* otherwise. Accordingly:

$$Y(i) = y_i = \begin{cases} F & \text{if iteration } i \text{ is faulty} \\ NF & \text{if iteration } i \text{ is not faulty} \end{cases}. \tag{3}$$

When dealing with R2F data, each maintenance cycle ends with a failure, hence the available data contains N samples for class F and $n - N$ samples for class NF. Based on X and Y, a classifier learns a decision rule $f(\cdot)$ that assigns one of the two classes $\{F, NF\}$ to each point in the input space $\mathbb{R}^p$.

The formulation presented above is weak from a PdM perspective in two aspects:
1. only the current process iteration is classified, i.e. no

fault prevention policy can be implemented.

2. no operating cost optimization policy is enforced.

As far as total operating cost minimization is concerned, two key metrics can be defined, namely[2]:

1) Frequency of Unexpected Breaks (ρ_{UB}) - percentage of failures not prevented;
2) Amount of Unexploited Lifetime (ρ_{UL}) - average number of process iterations that could have been run before failure if the preventative maintenance suggested by the maintenance management module had not been performed.

Different costs, c_{UB} and c_{UL}, can be associated with ρ_{UB} and ρ_{UL}, respectively, where generally $c_{UB} >> c_{UL}$, given that c_{UB} relates to unplanned interruption of production. It is not possible to simultaneously minimize both metrics, rather an optimal trade-off solution is sought through minimization of the total operating costs, as defined by the weighted sum

$$J = \rho_{UB}c_{UB} + \rho_{UL}c_{UL}. \qquad (4)$$

As formulated above, the classification problem is not suitable for maintenance managements purposes, because it doesn't facilitate identification of policies that minimise J in (4).

From a classification perspective, the formulation described in Eq. 3 also suffers from the fact that the dataset is very *unbalanced* or *skewed*, with N samples in class F and $n - N$ samples in class NF and $N \ll n$. Classification using unbalanced datasets generally results in poor prediction accuracy and generalization performance [9].

These issues are addressed by the MC PdM methodology presented in the next subsection.

B. MC PdM Concept

A possible approach to preventing unexpected failures is to consider a different classification problem where, instead of only labeling the last iteration of a maintenance cycle as F, we label as F the last m iterations. From a PdM perspective, this approach allows us to provide more conservative maintenance recommendations by choosing larger values for the failure horizon m. Moreover, by assigning more samples to the F class we reduce the skewness of the dataset (Nm samples in class F and $n - Nm$ samples in class NF).

This can be repeated for k different values of the horizon m. In the proposed MC PdM approach, k different classifiers run on the module, each one facing a different classification problem and therefore providing different maintenance management performance outcomes in terms of ρ_{UB} and ρ_{UL}.

The MC PdM scheme is depicted in Fig. 1. The j-th classifier is associated with the labels $Y^{(j)}$, where, considering the t-th process iteration of the i-th maintenance
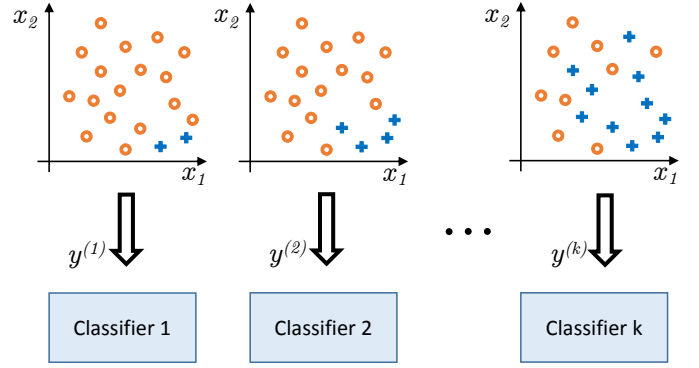

Fig. 1. 2-dimensional example for the k classifiers of the MC PdM methodology: the labels NF (orange circle) and F (blue plus) change with failure horizon m resulting in a different classification problem for each classifier

cycle (of length n_i) we have

$$y_t^{(j)} = \begin{cases} NF & \text{if } t \le n_i - m^{(j)} \\ F & \text{otherwise} \end{cases}, \qquad (5)$$

where $m^{(j)} \in \mathbb{N}^+$; the associated PdM performance of the j-th classifier is indicated by $\rho_{UB}^{(j)}$ and $\rho_{UL}^{(j)}$.

In the presented approach the k classifiers work in parallel and a maintenance event is triggered by the decision making logic based on the following operating costs minimization philosophy: given the current costs $c_{UB}(t)$ and $c_{UL}(t)$ at time t, the MC PdM suggests a corrective action when the j^*-th classifier

$$j^* = \arg \min_{j=1,\dots,k} J^{(j)}(t)$$
$$= \arg \min_{j=1,\dots,k} \rho_{UB}^{(j)}c_{UB}(t) + \rho_{UL}^{(j)}c_{UL}(t) \qquad (6)$$

outputs a label F classification.

The training procedure for MC PdM is sketched in Algorithm 1. The performance of each classifier is evaluated using Repeated Random Sub-Sampling Validation [16], also known as *Monte Carlo crossvalidation (MCCV)*: Q simulations are performed by randomly splitting the N maintenance cycles into a training dataset of $N_{TR} = \lfloor Nq \rfloor$ maintenance cycles and a validation dataset of $N_{VL} = \lceil N(1-q) \rceil$ maintenance cycles, with $0 < q < 1$. It has been shown [20] that MCCV is asymptotically consistent resulting in more pessimistic predictions of the test data compared with full crossvalidation. Algorithm 1 is applicable to any choice of classification algorithm. In general (as in the case of the two algorithms presented in Section IV-B) to set up the classifiers the tuning of one or more hyper-parameters is required. This is achieved through crossvalidation with the Missclassification Rate (MCR)

$$\text{MCR[\%]} = \text{Percentage of missclassified samples}$$

employed as a performance indicator.

Fab integration and on-line operation of the MC PdM module are described in Fig. 2. Estimates of ρ_{UB} and

---

[2]The R2F approach clearly guarantees a total of $\rho_{UB} = 1$ and $\rho_{UL} = 0$, while the performance of PvM and PdM approaches depend on some tuning parameters as will be illustrated in Section IV

Algorithm 1: MC PdM Training

Data: X, Y, k, q_1, q_2, Q_1, Q_2, $\{m^{(j)}\}_{j=1}^k$

Result: Classifiers $\{f^{(j)}\}_{j=1}^k$ and PdM performances

1. Let $\rho_{\text{UB}} = [\,\cdot\,]$ and $\rho_{\text{UL}} = [\,\cdot\,]$ (empty vectors)

for $j = 1$ *to* k **do**

  2. Compute $Y^{(j)}$ as in Eq. 5 **for** $i = 1$ *to* Q_1 **do**

    3. Randomly split the maintenance cycles between training and validation samples, keeping the ratio q_1

    4. Compute the MCR of the classifier with different hyper-parameters

  5. Chose the hyper-parameters based on the averaged MCR over the Q_1 simulations

  6. Compute $f^{(j)}$ with the selected hyper-parameters

  for $i = 1$ *to* Q_2 **do**

    7. Randomly split the maintenance cycles between training and validation, keeping the ratio q_2

    8. Compute $f^{(j)}$ with the selected hyper-parameters

    9. Compute ρ_{UB} and ρ_{UL} of $f^{(j)}$

  10. Compute ρ_{UB}^* and ρ_{UL}^* as averaged over the Q_2 simulations

  11. $\rho_{\text{UB}} = [\rho_{\text{UB}}; \rho_{\text{UB}}^*]$ and $\rho_{\text{UL}} = [\rho_{\text{UL}}; \rho_{\text{UL}}^*]$
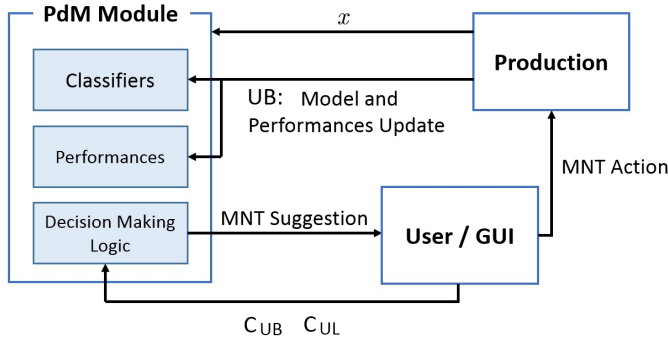

Fig. 2. Overview of how the MC PdM module integrates with the Fab

ρ_{UL} are provided by the historical and simulation performances of the PdM module, while c_{UB} and c_{UL} are provided by the user, and may be changed at each evaluation of the PdM module (i.e. unexpected breaks are much more costly during highly intense production periods than during normal production periods).

While undesirable from a operating cost perspective, when an unexpected break occurs during operation of a PdM module it means that a full maintenance cycle has been observed, and hence valuable new data is available with which to update the MC PdM module and related performance metrics. Therefore, as illustrated in Fig. 2, it is important to retain the facility to update the MC PdM module following deployment.

IV. IMPLEMENTATION DETAILS

A. Failure Horizon Selection

Performance of the MC PdM methodology increases with the number of classifiers, k, since each classifier provides more information on the health status of the process. Unfortunately k cannot be considered as a degree of freedom in the MC PdM design since it is constrained by the available computational and storage capabilities and by the algorithm chosen for the classification. In the following we propose two strategies for defining the failure horizons of the k classifiers.

The k classifiers are distributed in order to have equally spaced failure horizons in terms of the number of process iterations; thus the k horizons employed in (5) are assigned as follows

$$m^{(j)} = \left\lfloor \frac{(j-1)(M_1 - 1)}{k-1} \right\rceil + 1 \quad \text{for } j = 1, \ldots, k \quad (7)$$

where $\lfloor u \rceil$ is the closest integer to u and $M_1 > 0$ specifies the maximum failure horizon and is chosen based on the lengths of the maintenance cycles in the training dataset.

An alternative approach for selecting the failure horizons is to express them as the percentage of a maintenance cycle, rather than a fixed number of process iterations. Thus, in this second approach we have

$$y_t^{(j)} = \begin{cases} NF & \text{if } t \le \lfloor n_i \left(1 - m^{(j)}\right) \rceil \\ F & \text{otherwise} \end{cases}, \quad (8)$$

with

$$m^{(j)} = \frac{(j-1)M_2}{k-1} \quad \text{for } j = 1, \ldots, k \quad (9)$$

where $M_2 \in (0, 1]$.

We remark that the choice of criteria used to define the failure horizons depends on the nature of the fault under investigation and, as such, there is no a-priori best solution.

B. Classification Algorithms

The MC PdM methodology presented does not impose any restrictions on the classification algorithm that can be adopted for the individual classifiers. Here, we consider two well known and widely used classification techniques, namely: Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NN). This choice is motivated by the fact that it provides a comparison between two contrasting types of classifier; a powerful, but computationally complex to estimate, model based classifier (SVM); and a low-complexity non-parametric classifier (k-NN). In the following the two selected classifiers are briefly presented.

1) Support Vector Machines: SVMs are probably the most popular approach to classification, thanks to their high classification accuracy, even for non-linear problems, and to the availability of optimized algorithms for their computation [3].

Consider the problem where two classes of data (F and NF in the problem at hand) have to be classified;

we initially consider the *separable case*, where the two classes can be linearly separated. Suppose that a training dataset $\mathcal{S}$ as in Eq. (1) is available and the values $\{-1, 1\}$ are assigned to the two classes which the data belong to, for example

$$y_i = \begin{cases} 1 & \text{if } i\text{-th sample} \in F \\ -1 & \text{if } i\text{-th sample} \in NF \end{cases}$$

We define the hyperplane F_0 in the $\mathbb{R}^p$ space as

$$F_0 = \{x \,|\, f(x) = x\beta + \beta_0 = 0\}, \tag{10}$$

where $\beta \in \mathbb{R}^p$ with norm $\|\beta\| = 1$. The classification is then based on the choice of $f(x)$ (and consequently of F_0): for a new sample $x^{\text{new}} \notin \mathcal{S}$, we classify

$$\widehat{y^{\text{new}}} = \begin{cases} 1 \ (\text{Class } F) & \text{if } f(x^{\text{new}}) > 0 \\ -1 \ (\text{Class } NF) & \text{if } f(x^{\text{new}}) < 0 \end{cases} \tag{11}$$

Since, by assumption, the two classes are separable, then it is possible to find a function $f(x)$ s.t.

$$y_i f(x) > 0 \quad \forall i;$$

The hyperplane yielding the largest margin Π between the two classes is chosen: this can be rephrased as the maximization problem

$$\max_{\beta, \beta_0, \|\beta\|} \Pi \\ \text{subject to} \quad y_i f(x) \geq \Pi, \ i = 1, \ldots, n. \tag{12}$$

We refer the interested reader to [4] for details on how (12) is solved.

In real classification problems, classes are often non-separable, i.e., the two categories overlap: SVMs can still be used in this case by allowing some samples to be on the 'wrong' side of the separation line. With respect to the separable case the optimization problem (12) is changed by modifying the constraint to:

$$y(x_i\beta + \beta_0) \geq \Pi - \xi_i^* = \Pi(1 - \xi_i) \quad \forall i, \tag{13}$$

where the *slack variables* ξ_i have to satisfy the conditions: $\xi_i \geq 0$ (the points on the wrong side of their margin are labeled with $\xi_i^* > 0$, while the points on the correct side have $\xi_i^* = 0$) and $\sum_{i=1}^{n} \xi_i \leq C \in \mathbb{R}$. The optimization problem is now (see [4] for a detailed solution):

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\| + \gamma \sum_{i=1}^{n} \xi_i \\ \text{subject to} \begin{cases} y(x_i\beta + \beta_0) \geq 1 - \xi_i, \ \forall i \\ \xi_i \geq 0 \end{cases}, \tag{14}$$

where the regularization parameter γ governs the trade-off between the margin width and the sum of values of the slack variables.

SVMs are usually employed in combination with *Kernel Methods* to further enhance the classification performance by allowing non-linear solutions. The computational cost for training a nonlinear SVM is generally between $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$, depending on the algorithm employed for its computation. For more detailed presentations on SVMs we refer interested readers to [19] and [25].
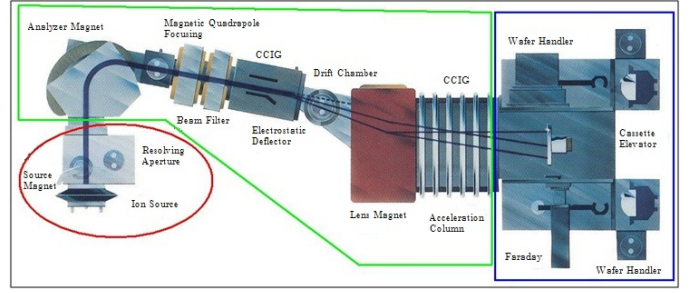

Fig. 3. Schematic of a generic Ion Implanter tool. The tool can be divided into three main parts: the Source ◯, the Beamline Area ▢, the End Station ▢

2) k-Nearest Neighbours: k-NN is probably the simplest approach to classification as it requires just computation of distances between samples.

In the *k*-NN procedure each point of the input space is labelled according to the labels of its k closest neighbouring samples (where distances are computed according to a given metric, often the Euclidean norm).

The only parameter that requires tuning in *k*-NN is k, the value of the number of samples in the considered neighbourhood: the choice of k is usually data-driven (often decided though cross-validation). Larger values of k reduce the effect of noise on the classification, but make decision boundaries between classes less distinct [28].

Optimized algorithms for computing k-NN have a computational cost of $\mathcal{O}(\log n)$.

V. EXPERIMENTAL RESULTS

A. Use Case

We test the proposed MC PdM methodology for replacing tungsten filaments used in Ion Implantation [13], one of the most important processes in semiconductor manufacturing fabrication. Ion Implantation is used to modify the electrical properties of wafers by injecting doping atoms. It is often considered a 'bottleneck' in production lines due to the high cost of the tool, making it a critical operation for throughput [11].

The components of a typical Implanter tool are illustrated in Fig. 3. The filament is part of the Ion Source section of the tool. During the process, the filament is heated and electrons are 'boiled' off the heated filament. The electrons are then accelerated in the beamline area and impinge on the target wafers in the End Station. The tungsten filament must be frequently replaced. Every time a filament is changed, the tool is down for approximately 3 hours making this the most important maintenance issue for process engineers working on the tool.

Several factors may influence the working duration of a filament. For example, it is known that high values of pressure, voltage and filament current can drastically reduce the lifetime of the filament. The operations of cleaning, installation and degasification can also have a fundamental impact on filament 'health' and duration.

Physical/Electrical Variable	Quantity
Current	9
Current Transfer Ratio	1
Deceleration	1
Flow	1
Number of Scans	1
Plasma Gun Emission	1
Position	2
Pressure	3
Quantity of Electric Charge	2
Scan Speed	1
Tilt Angle	2
Voltage	7

TABLE I
RECORDED TOOL VARIABLES

Type	Based on:	Acronym
PvM	Mean μ	PvM-μ
	Median η	PvM-η
PdM	Linear SVM	PdM-lin
	Gaussian Kernel SVM	PdM-rbf
MC PdM	k-NN	MC PdM-knn
	SVM	MC PdM-svm

TABLE II
TESTED MAINTENANCE MANAGEMENT TECHNIQUES

B. Data Description

The available data consists of $N = 33$ maintenance cycles, with the filament maintained using a R2F policy (i.e. each maintenance cycle consists of the data for the implanter tool from the installation of a new filament to the point that the filament breaks and the tool is stopped for maintenance), for a total of $n = 3671$ batches.

A total of 31 variables, as listed in Table I, were recorded from the ion-implantation tool during the n runs. These variables have a time series evolution during each run, with some presenting with a non-uniform sampling rate from observation to observation. In order to construct a design matrix to feed the classifiers, features must be extracted. A classical approach [18] to extracting such features is to rely on statistical moments. In this work we extracted 6 features for each of the 31 time series, namely: maximum, minimum, average, variance, skewness and kurtosis. After constant variables are discarded a total of $p = 125$ input variables are retained in the dataset.

C. Comparison of SVM and kNN Based Approaches

The MC PdM approach has been tested on the problem described above using the classification algorithms described in Section IV-B. We use MC PdM-*knn* and MC PdM-*svm* to denote MC PdM implemented with k-NN and SVM, respectively. In the case of SVM, we have employed Radial Basis Function (RBF) kernels to implement the non-linear classification boundaries in the linear framework presented in Section IV-B (at the expense of a more complicated tuning procedure, see [4]). A linear SVM implementation, denoted PdM-*lin*, is also considered.

The maintenance strategies investigated are summarized in Table II. A number of approaches are compared to MC PdM as follows:

1) A simulated PvM policy: PvM policies are usually based on the mean μ or median η of the maintenance cycle lengths and on the definition of an optimal action threshold τ_μ^* or τ_η^*. Once τ_μ^* is computed, then the PvM maintenance is triggered if

$$f^\mu = \text{iterations since last maintenance} > \mu - \tau_\mu^*, \quad (15)$$

and similarly for the median. The PvM policy based on the mean (PvM-μ) is outlined in Algorithm 2 (the version based on the median, PvM-η, is a natural extension).

Algorithm 2: PvM module

Data: Y, c_{UB}, c_{UL}
Result: Maintenance Rule (defined by $\tau_\mu*$), ρ_{UB}^μ, ρ_{UL}^μ and $J^\mu(T^\mu)$
1. Compute μ, the mean number of process iterations in a maintenance cycle
2. Define a set of threshold values $T^\mu \in \mathbb{R}^d$
3. Let $\rho_{\mathsf{UB}}^\mu = [\,\cdot\,]$ and $\rho_{\mathsf{UL}}^\mu = [\,\cdot\,]$ (empty vectors)
for $j = 1$ *to* N **do**
    4. Compute ρ_{UB}^μ and ρ_{UL}^μ for all entries in T^μ
    5. $\rho_{\mathsf{UB}}^\mu = [\rho_{\mathsf{UB}}^\mu; \rho_{\mathsf{UB}}^\mu]$ and $\rho_{\mathsf{UL}}^\mu = [\rho_{\mathsf{UL}}^\mu; \rho_{\mathsf{UL}}^\mu]$
6. $\rho_{\mathsf{UB}}^\mu = \text{Mean}(\rho_{\mathsf{UB}}^\mu)$ and $\rho_{\mathsf{UL}}^\mu = \text{Mean}(\rho_{\mathsf{UL}}^\mu)$
7. Compute $J^\mu(\tau^\mu)$ for all entries in T^μ: $J^\mu(T^\mu)$
8. Let $\tau^\mu* = \min_{\tau^\mu \in T^\mu} J^\mu(\tau^\mu)$

---

2) An SVM Classification distance-based PdM system: This is based on the PdM approach developed in [24] and involves exploiting the distance from a computed SVM decision boundary to define a metric for the distance an observed sample is from a faulty situation. The assumption is that the more stress on a machine, the closer statistically we get to a faulty situation and hence the distance from the separation boundary of the two SVM classes decreases. The resulting PdM module recommends that a corrective action be performed when

$$f(x) < \tau \in \mathbb{R}^+, \quad (16)$$

and different choices of τ affect the performance of the PdM in terms of ρ_{UB} and ρ_{UL}. We consider here both linear SVM (PdM-lin) and RBF kernel SVM (PdM-rbf) classifiers.

D. Settings and Results

The parameters employed in the simulations of the MC PdM methodology are
- $k = 10$;
- $M_1 = 85$.

These values have been selected manually for illustrative purposes. Setting $M_1 = 85$ ensures that the PdM modules produce no unexpected breaks, while selecting $k = 10$ gives sufficient diversity in performance in terms
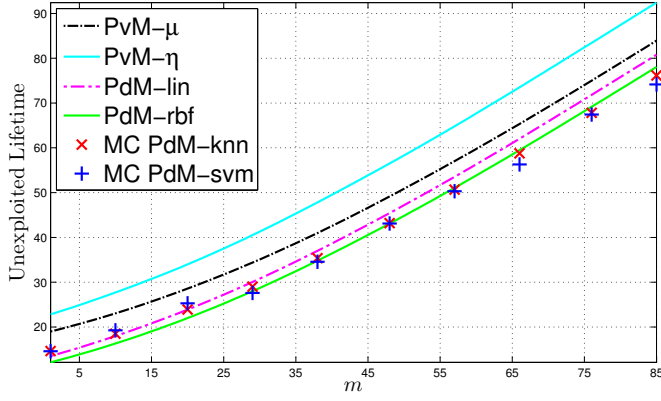
Fig. 4. Average ρ_{UL} over Q_2 Monte Carlo simulations with the various maintenance management approaches
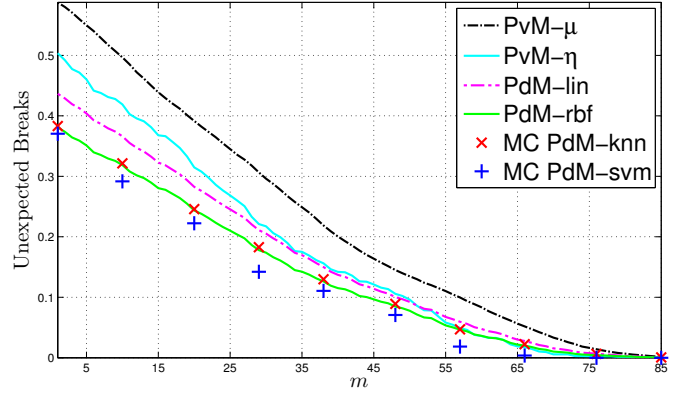

Fig. 5. Average ρ_{UB} over Q_2 Monte Carlo simulations with the various maintenance management approaches

of ρ_{UL} and ρ_{UB} to be effective. The criterion described in Eq. (5) and Eq. (7) was used to select the failure horizon for each classifier as this was found to provide a better distribution of results in terms of ρ_{UL} and ρ_{UB} than the one described in Eq. (8) and Eq. (9). In the interest of compactness, the results of this investigation are not included here.

As described in Algorithm 1, prior to application the hyper-parameters of the MC PdM classifiers must be tuned using cross-validation techniques. This also applies to the SVM classification distance based PdM approach. In the case of linear SVMs, only one parameter has to be optimised, namely, the variable γ in Equation (14), while for RBF SVMs two parameters need to be tuned (γ and the variance of the Gaussian kernels). In the case of k-NN the parameter k (the size of the neighbourhood) has to be set. The parameters we employed in the cross-validation procedure are:

- $q_1 = q_2 = 0.7$;
- $Q_1 = Q_2 = 1000$.

Cross-validation outcomes are not detailed here due to space constraints, but we report the fact that the optimal value for the parameter k in MC PdM-knn is consistently equal to 1 in the simulations. This may be due to the high dimensionality of the problem.

In Fig. 4 and 5 the average performance of the various maintenance approaches is reported in terms of ρ_{UL} and ρ_{UB} for different values of m over Q_2 Monte Carlo simulations. The thresholds of the PvM and PdM approaches are scaled to facilitate comparison with the MC PdM approaches. The linear scaling has been applied after ρ_{UL} and ρ_{UB} has been computed with the various approaches so that their performances are in a similar range when plotted in Fig. 4 and Fig. 5.

It can be appreciated how PdM and MC PdM outperform PvM approaches for both metrics. In the case of ρ_{UL} the MC PdM classifiers achieve similar performance to the PdM approaches, with PdM-rbf marginally superior for low values of m and MC PDM-svm marginally superior for higher values. The improves performance with higher m can be ascribed to the fact that these
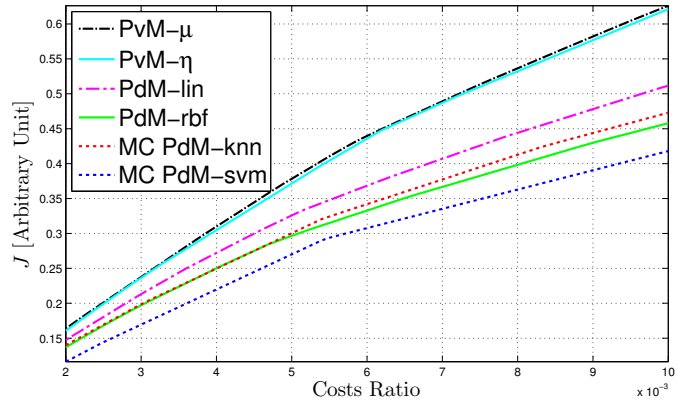

Fig. 6. Optimal value of J for the various maintenance strategies as a function of the ratio of the costs $c_{\mathsf{UL}}/c_{\mathsf{UB}}$ (averaged results over Q_2 Monte Carlo simulations)

classifiers are estimated using more balanced (i.e. less skewed) datasets. In the case of ρ_{UB}, MC PdM-svm consistently achieves the best results. Overall when considering both ρ_{UL} and ρ_{UB} MC PdM-svm guarantees better performance than MC PdM-knn, but this is at the expense of a much more time consuming tuning procedure.

In Table III the average classification accuracy (100% - MCR), precision and recall are reported for the various

	kNN			SVM		
m	Accuracy	Precision	Recall	Accuracy	Precision	Recall
1	98.51	69.27	61.69	98.52	69.34	62.96
10	98.47	64.09	67.85	98.47	63.14	70.84
20	98.39	58.00	75.42	98.37	56.61	77.77
29	98.29	53.22	81.72	98.33	54.46	85.79
38	98.15	48.27	87.03	98.16	48.85	88.95
48	97.95	43.31	91.09	97.95	43.36	92.94
57	97.74	39.44	95.30	97.74	39.61	98.15
66	97.52	35.96	97.76	97.57	36.96	99.63
76	97.26	32.74	99.36	97.26	32.86	100.00
85	97.03	30.23	99.94	97.08	30.08	100.00

TABLE III
AVERAGE ACCURACY, PRECISION AND RECALL FOR THE MC PDM MODULES (PERCENTAGE RESULTS) OVER Q_2 MONTE CARLO SIMULATIONS

classifiers that make up MC PdM-knn and MC PdM-svm. In computing these values a process iteration with a broken filament is considered as the positive condition, while an uninterrupted process iteration is considered as the negative condition. It can be observed that as the fault horizon m increases, precision decreases (more false positives, i.e. unexploited lifetime) while recall increases (less false negatives, i.e. unexpected breaks).

In Fig. 6 the performance of the six maintenance approaches is plotted as a function of the ratio of the costs, c_{UL}/c_{UB}. Here, J is the minimum total cost achieved with each approach and is computed based on the average Monte Carlo results reported in Fig. 4 and 5. The values of the costs ratio c_{UL}/c_{UB} were chosen with the aid of process experts in order to have plausible results for the problem at hand.

These results highlight that MC PdM-svm consistently guarantees better performance than any of the other approaches considered. This is the most important outcome of the present work, since it demonstrates that the proposed maintenance methodology achieves the lowest operating costs while at the same time being robust to different choices of ρ_{UL} and ρ_{UB}. MC PdM-knn consistently delivers lower operating costs than classical PvM techniques and exhibits comparable performance to PvM-svm, especially for low values of the ratio c_{UL}/c_{UB}.

VI. CONCLUSIONS

A novel multiple classifier PdM system for integral type faults has been presented. The multiple ML classifiers work in parallel to exploit the knowledge of the tool/logistic variables at each process iteration in order to enhance decision making. The proposed tool guarantees improved maintenance management decisions in terms of minimising operating cost and can be applied to any maintenance problems characterised by integral type faults provided R2F historical data is available or can be collected.

The MC PdM module is robust to variations in the operating costs associated with unexpected breaks and unexploited equipment part lifetime that can occur over time. It also allows the user to dynamically change maintenance policy based on current costs and needs of manufacturing production. Process engineers are provided with a tool that enables them to adjust the performance and the action policy so that the total operating cost is minimized. Moreover, the presence of several classifiers allows the proposed tool to be employed as a health factor indicator in a more natural and straightforward fashion that with other PdM approaches. Specifically, the k classifiers provide an estimate of the current status of the maintenance issue, enhancing process understanding and allowing cost aware maintenance management decisions at any time.

The proposed methodology has been demonstrated for a semiconductor manufacturing Ion-Implanter related maintenance task and shown to guarantee better performance than classical PvM approaches and a single SVM classifier distance based PdM alternative. It has also been shown in this case study that SVMs offer superior performance to k-NN classifiers when implementing MC-PdM, and that in general MC-PdM-knn also consistently outperforms PvM approaches.

The high dimensionality of datasets typical of PdM problems encountered in semiconductor manufacturing and other manufacturing industries, suggests that parsimonious (or sparse) classification approaches such as Relevance Vector Machines [27] could provide competitive results. This will be the subject of a future extension of the present work.

We have seen how the choice of the fault horizon m strongly affects the performance of the corresponding classifier in MC PdM. It may be the case that some areas of the $[\rho_{UB}, \rho_{UL}]$ space are more 'interesting', given the associated costs, than others in the context of minimising total operating costs. One approach to cope with this issue is to employ a two steps procedure, where classifiers assigned to an uninteresting area of $[\rho_{UB}, \rho_{UL}]$ are reallocated in the second training iteration to more relevant areas of the cost space.

REFERENCES

[1] R. Baly and H. Hajj. Wafer classification using support vector machines. *IEEE Transactions on Semiconductor Manufacturing*, 25(3):373–383, 2012.

[2] S. Butler and J. Ringwood. Particle filters for remaining useful life estimation of abatement equipment used in semiconductor manufacturing. In *IEEE Conference on Control and Fault-Tolerant Systems*, pages 436–441, 2010.

[3] C.-C. Chang and C.J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[4] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning. Data Mining, Inference and Prediction.* Springer, 2009.

[5] A. Heng, A. Tan, J. Mathew, N. Montgomery, D. Banjevic, and A. Jardine. Intelligent condition-based prediction of machinery reliability. *Mechanical Systems and Signal Processing*, 23:1600–1614, 2009.

[6] G. Koksal, I. Batmaz, and M.C. Testik. A review of data mining applications for quality improvement in manufacturing industry. *Expert Systems with Applications*, 38(10):13448 – 13467, 2011.

[7] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis. Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the north sea. In *SenSys*, 2005.

[8] R. Langone, C. Alzate, B. DeKetelaere, and J.A.K. Suykens. Kernel spectral clustering for predicting maintenance of industrial machines. In *Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on*, pages 39–45, April 2013.

[9] S.S. Lee. Noisy replication in skewed binary classification. *Computational statistics & data analysis*, 34(2):165–191, 2000.

[10] B. Lenz and B. Barak. Data mining and support vector regression machine learning in semiconductor manufacturing to improve virtual metrology. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 3447–3456. IEEE, 2013.

[11] S.-Y. Lin and S.-C. Horng. A classification-based fault detection and isolation scheme for the ion implanter. *IEEE Transactions on Semiconductor Manufacturing*, 19(4):411–424, 2006.

[12] M. Luo, Z. Xu, H.L. Chan, and M. Alavi. Online predictive maintenance approach for semiconductor equipment. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pages 3662–3667. IEEE, 2013.

[13] C.M. McKenna. A personal historical perspective of ion implantation equipment for semiconductor applications. In *Conference on Ion Implantation Technology*, pages 1 – 19, 2000.

[14] R.K. Mobley. *An Introduction to Predictive Maintenance.* Butterworth-Heinemann, 2002.

[15] L. Mönch, J.W. Fowler, S. Dauzère-Pérès, S.J. Mason, and O. Rose. A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *Journal of Scheduling*, 14(6):583–599, 2011.

[16] R.R. Picard and R.D. Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79:575–583, 1984.

[17] General Electrics Intelligent Platforms. The rise of industrial big data, August 2012.

[18] A. Schirru, G.A. Susto, S. Pampuri, and S. McLoone. Learning from time series: Supervised aggregative feature extraction. In *51st IEEE Conference on Decision and Control, Proceedings*, pages 5254–5259. IEEE, 2012.

[19] B. Scholkopf and A.J. Smola. *Learning with Kernels*. MIT Press, 2002.

[20] J. Shao. Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88:486–494, 1993.

[21] Y.-C. Su, F.-T. Cheng, M.-H. Hung, and H.-C.Huang. Intelligent prognostics system design and implementation. *IEEE Trans. on Semiconductor Manufacturing*, 19:195–207, 2006.

[22] G.A. Susto, A. Beghi, and C. DeLuca. A predictive maintenance system for epitaxy processes based on filtering and prediction techniques. *IEEE Trans. on Semiconductor Manufacturing*, 25:638–649, 2012.

[23] G.A. Susto, S. Pampuri, A. Schirru, A. Beghi, and G. DeNicolao. Multi-step virtual metrology for semiconductor manufacturing: a multilevel and regularization methods-based approach. *Computers & Operations Research*, (0):–, 2014.

[24] G.A. Susto, A. Schirru, S. Pampuri, D. Pagano, S. McLoone, and A. Beghi. A predictive maintenance system for integral type faults based on support vector machines: An application to ion implantation. In *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*, pages 195–200. IEEE, 2013.

[25] J. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[26] K. Tamaki and S. Kaneko. Multiparametric virtual metrology model building by job-shop data fusion using a markov chain monte carlo method. 2013.

[27] M.E. Tipping. Sparse bayesian learning and the relevance vector machine. *The journal of machine learning research*, 1:211–244, 2001.

[28] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

Andrea Schirru received the M.S. degree (cum laude) in computer science and the Ph.D. in Information Engineering from the University of Pavia, Pavia, Italy, in 2008 and 2011. He is currently a post-doctoral associate at National University of Ireland, Maynooth and Chief Technology Officer at Statwolf Ltd.

He has been an Intern Researcher at Infineon Technologies Austria AG, Villach, Austria (2010-2011), and during his studies he has been awarded with the ICINCO Best Student Paper Award (2011) and the Best Student Presentation at the Intel European Research and Innovation Conference (2010).

Simone Pampuri received the M.S. degree in computer science and the Ph.D. in Information Engineering from the University of Pavia, Pavia, Italy, in 2009 and 2012. He is currently a post-doctoral associate at National University of Ireland, Maynooth and Chief Executive Officer at Statwolf Ltd.

He has been an Intern Researcher at Infineon Technologies Austria AG, Villach, Austria (2011). His research interests include manufacturing data analytics, machine learning and sentiment analysis.

Seán McLoone (S'94, M'96, SM'02) received a Master of Engineering degree with Distinction and a Ph.D. degree in Control Engineering from Queens University Belfast (QUB), Belfast, Ireland, in 1992 and 1996, respectively. He is currently Professor and Director of the Energy Power and Intelligent Control (EPIC) Research Cluster at QUB.

His research interests lie in the general area to data based modelling and analysis of dynamical systems. This encompasses techniques ranging from classical system identification, fault diagnosis, and statistical process control to modern computational intelligence based adaptive learning algorithms and optimization techniques. He is a Past Chairman of the UK and Republic of Ireland (UKRI) Section of the IEEE.
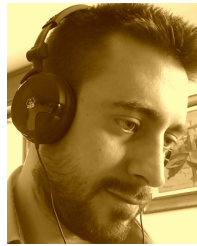
Gian Antonio Susto (S'11) received the M.S. degree (cum laude) in control systems engineering and the Ph.D. in Information Engineering from the University of Padova, Padova, Italy, in 2009 and 2013. He is currently a post-doctoral associate at University of Padova and Chief Operating Officer at Statwolf Ltd.

He has been a visiting student at the University of California, San Diego (2008-09), and at National University of Ireland, Maynooth (2012) and an Intern Researcher at Infineon Technologies Austria AG, Villach, Austria (2011). During his studies he has been awarded with IEEE-CASE Best Student Conference Paper Award (2011), IEEE/SEMI-ASMC Best Student Paper Award (2012) and IEEE-MSC Best Student Paper Award (2012). His research interests include manufacturing data analytics, machine learning, gesture recognition and partial differential equations control.

Alessandro Beghi (M'06) received the laurea degree cum laude in Electrical Engineering in 1989 and the Ph.D. degree in Control Systems Engineering in 1993, both from the University of Padova, Italy. In 1994 he joined the Department of Information Engineering, University of Padova, Italy, where he is currently an Associate Professor of Control Systems Theory.

He held visiting positions at universities and international research centers both in Europe and in the USA. He is a member of the IEEE Control System Society, serving also as Associate Editor for the Conference Editorial Board and co-chair of the Technical Committee on Power Generation. He has worked on a wide range of control applications, from control of fusion devices, guidance algorithms for virtual vehicles, to control of HVAC&R systems, Adaptive Optics systems, and Semiconductor Manufacturing processes. He is the author of more than 130 publications in journals, books, and conference proceedings. He is co-inventor of 5 International Patents on applications of advanced control techniques.