

# Machine Learning for Predictive Maintenance: A Multiple Classifier Approach

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

**Abstract**—In this paper, a multiple classifier machine learning (ML) 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 the 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 tradeoffs in terms of frequency of unexpected breaks and unexploited lifetime, and then employing this information in an operating cost-based maintenance decision system to minimize 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 (ML), predictive maintenance (PdM), 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 decrease 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.

- 1) *Run-to-failure (R2F)*—where maintenance interventions are performed only after the occurrence of failures. This is obviously the simplest approach 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.
- 2) *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.
- 3) *PdM*<sup>1</sup>—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 prefailure 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 that describes the failures that happen on a machine due to the accumulative “wear and tear” effects of usage 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

Manuscript received March 17, 2014; revised June 30, 2014; accepted August 7, 2014. Date of publication August 18, 2014; date of current version June 02, 2015. This work was supported by the framework of Integrated Solutions for Agile Manufacturing in High-mix Semiconductor Fabs (INTEGRATE), a European Nanoelectronics Initiative Advisory Council project. Paper no. TII-14-0329.

G. A. Susto is with the Department of Information Engineering, University of Padova, 35131 Padova, Italy; and also with Statwolf Ltd., Dublin, Ireland (e-mail: gianantonio.susto@statwolf.com).

A. Schirru and S. Pampuri are with the National University of Ireland, Maynooth, Ireland; and also with Statwolf Ltd., Dublin, Ireland.

S. McLoone is with the Queen's University, Belfast, U.K.; and also with the National University of Ireland, Maynooth, Ireland.

A. Beghi is with the Department of Information Engineering, University of Padova, 35131 Padova, Italy.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TII.2014.2349359

<sup>1</sup>Other authors [8] refer to this class of maintenance approaches as condition-based maintenance.

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 the following two main classes.

- 1) *Supervised*—where information on the occurrence of failures is present in the modeling dataset.
- 2) *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) are 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 an ML perspective, supervised approaches require the availability of a dataset  $\mathcal{S}$

$$\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n \quad (1)$$

where a couple  $\{\mathbf{x}_i, y_i\}$  (called *observation*) contains the information related to the  $i$ th process iteration. Here, vector  $\mathbf{x}_i \in \mathbb{R}^{1 \times p}$  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.

- 1) If  $y$  assumes continuous values, a *regression* problem is obtained.
- 2) 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 (e.g., 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 nonfaulty 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, e.g., 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

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T \quad (2)$$

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,

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

When dealing with R2F data, each maintenance cycle ends with a failure, hence the available data contain  $N$  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<sup>2</sup>.

- 1) Frequency of Unexpected Breaks ( $\rho_{UB}$ )—percentage of failures not prevented.
- 2) Amount of Unexploited Lifetime ( $\rho_{UL}$ )—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.

<sup>2</sup>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.

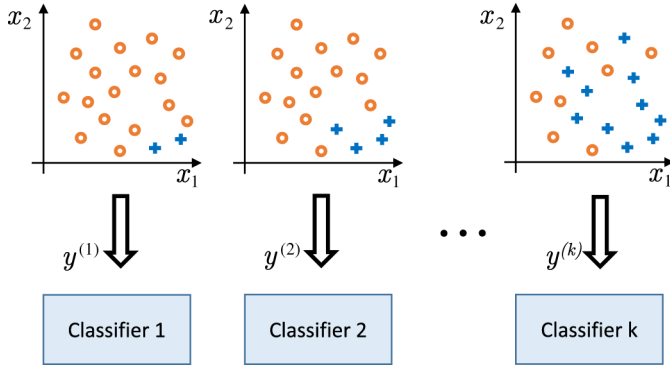


Fig. 1. Two-dimensional example for the  $k$  classifiers of the MC PdM methodology: the labels  $NF$  (orange circle) and  $F$  (blue plus) change with failure horizon  $m$  resulting in a different classification problem for each classifier.

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

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

As formulated above, the classification problem is not suitable for maintenance managements purposes, because it does not facilitate identification of policies that minimize  $J$  in (4).

From a classification perspective, the formulation described in (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 Section III-B.

### B. MC PdM Concept

A possible approach to prevent 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  $\rho_{UB}$  and  $\rho_{UL}$ .

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 cycle (of length  $n_i$ ) we have

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

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

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

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 subsampling 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 hyperparameters 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 online operation of the MC PdM module are described in Fig. 2. Estimates of  $\rho_{UB}$  and  $\rho_{UL}$  are provided

### 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  $\rho_{UB} = [\cdot]$  and  $\rho_{UL} = [\cdot]$  (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  $\rho_{UB}$  and  $\rho_{UL}$  of  $f^{(j)}$   
10. Compute  $\rho_{UB}^*$  and  $\rho_{UL}^*$  as averaged over the  $Q_2$  simulations  
11.  $\rho_{UB} = [\rho_{UB}; \rho_{UB}^*]$  and  $\rho_{UL} = [\rho_{UL}; \rho_{UL}^*]$



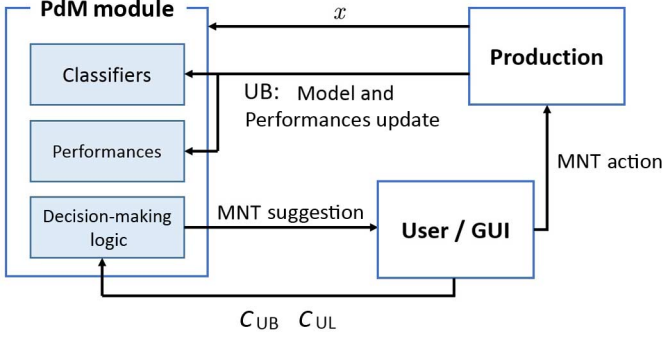


Fig. 2. Overview of how the MC PdM module integrates with the Fab.

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 an 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 are 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:

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

where  $\lfloor u \rfloor$  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

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

with

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

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: 1) support vector machines (SVMs); and 2)  $k$ -nearest neighbors ( $k$ -NN). This choice is motivated by the fact that it provides a comparison between two contrasting types of classifier: 1) a powerful, but computationally complex to estimate, model based classifier (SVM); and 2) a low-complexity nonparametric 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 nonlinear 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 (1) is available and the values  $\{-1, 1\}$  are assigned to the two classes which the data belong to, e.g.,

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

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

$$F_0 = \{x \mid f(x) = x\beta + \beta_0 = 0\} \quad (10)$$

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

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

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  $\Pi$  between the two classes is chosen: this can be rephrased as the maximization problem

$$\begin{aligned} & \max_{\beta, \beta_0, \|\beta\|} \Pi \\ & \text{subject to } y_i f(x) \geq \Pi, \quad i = 1, \dots, n. \end{aligned} \quad (12)$$

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

In real classification problems, classes are often nonseparable, 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

$$y(\mathbf{x}_i\beta + \beta_0) \geq \Pi - \xi_i^* = \Pi(1 - \xi_i) \quad \forall i \quad (13)$$

where the *slack variables*  $\xi_i$  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)

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

where the regularization parameter  $\gamma$  governs the tradeoff 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 nonlinear 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].

2) *k-Nearest Neighbors*:  $k$ -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 labeled according to the labels of its  $k$  closest neighboring 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 neighborhood: the choice of  $k$  is usually data-driven (often decided through 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

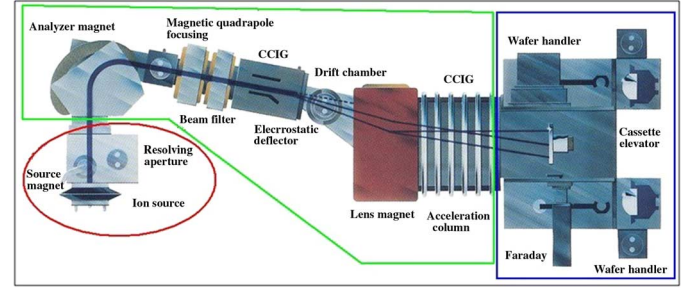


Fig. 3. Schematic of a generic ion implanter tool. The tool can be divided into three main parts: the Source ○, the Beamline Area □, and the End Station □.

TABLE I  
RECORDED TOOL VARIABLES

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

end station. The tungsten filament must be frequently replaced. Every time a filament is changed, the tool is down for approximately 3 h 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.

### B. Data Description

The available data consist of  $N = 33$  maintenance cycles, with the filament maintained using an 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 nonuniform 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 six features for each of the 31 time series, namely: 1) maximum; 2) minimum; 3) average; 4) variance; 5) skewness; and 6) kurtosis. After constant variables are discarded, a total of  $p = 125$  input variables are retained in the dataset.

TABLE II  
TESTED MAINTENANCE MANAGEMENT TECHNIQUES

Type	Based on:	Acronym
PvM	Mean $\mu$	PvM- $\mu$
	Median $\eta$	PvM- $\eta$
PdM	Linear SVM	PdM-lin
	Gaussian Kernel SVM	PdM-rbf
MC PdM	$k$ -NN	MC PdM-knn
	SVM	MC PdM-svm

### C. Comparison of SVM- and $k$ -NN-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 nonlinear 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 with MC PdM as follows.

1) *A Simulated PvM Policy:* PvM policies are usually based on the mean  $\mu$  or median  $\eta$  of the maintenance cycle lengths and on the definition of an optimal action threshold  $\tau_\mu^*$  or  $\tau_\eta^*$ . Once  $\tau_\mu^*$  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- $\mu$ ) is outlined in Algorithm 2 (the version based on the median PvM- $\eta$  is a natural extension).

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

---

#### Algorithm 2. PvM module

---

**Data:**  $Y, c_{UB}, c_{UL}$

**Result:** Maintenance Rule (defined by  $\tau_\mu^*$ ),  $\rho_{UB}^\mu$ ,  $\rho_{UL}^\mu$  and  $J^\mu(T^\mu)$

1. Compute  $\mu$ , 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  $\rho_{UB}^\mu = [\cdot]$  and  $\rho_{UL}^\mu = [\cdot]$  (empty vectors)
  - for**  $j = 1$  **to**  $N$  **do**
    4. Compute  $\rho_{UB}^\mu$  and  $\rho_{UL}^\mu$  for all entries in  $T^\mu$
    5.  $\rho_{UB}^\mu = [\rho_{UB}^\mu; \rho_{UB}^\mu]$  and  $\rho_{UL}^\mu = [\rho_{UL}^\mu; \rho_{UL}^\mu]$
    6.  $\rho_{UB}^\mu = \text{Mean}(\rho_{UB}^\mu)$  and  $\rho_{UL}^\mu = \text{Mean}(\rho_{UL}^\mu)$
    7. Compute  $J^\mu(\tau^\mu)$  for all entries in  $T^\mu$ :  $J^\mu(T^\mu)$
    8. Let  $\tau_\mu^* = \min_{\tau^\mu \in T^\mu} J^\mu(\tau^\mu)$
- 

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 can be performed when

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

and different choices of  $\tau$  affect the performance of the PdM in terms of  $\rho_{UB}$  and  $\rho_{UL}$ . 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: 1)  $k = 10$ ; 2)  $M_1 = 85$ .

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 of  $\rho_{UL}$  and  $\rho_{UB}$  to be effective. The criterion described in (5) and (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  $\rho_{UL}$  and  $\rho_{UB}$  than the one described in (8) and (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 optimized, namely, the variable  $\gamma$  in (14), while for RBF SVMs, two parameters need to be tuned ( $\gamma$  and the variance of the Gaussian kernels). In the case of  $k$ -NN, the parameter  $k$  (the size of the neighborhood) has to be set. The parameters we employed in the cross-validation procedure are: 1)  $q_1 = q_2 = 0.7$ ; 2)  $Q_1 = Q_2 = 1000$ .

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 Figs. 4 and 5, the average performance of various maintenance approaches is reported in terms of  $\rho_{UL}$  and  $\rho_{UB}$  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  $\rho_{UL}$  and  $\rho_{UB}$  has been computed with various approaches, so that their performances are in a similar range when plotted in Figs. 4 and 5.

It can be appreciated how PdM and MC PdM outperform PvM approaches for both metrics. In the case of  $\rho_{UL}$ , 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 improved performance with higher  $m$  can be ascribed to the fact that these classifiers are estimated using more balanced (i.e., less skewed) datasets. In case of  $\rho_{UB}$ , MC PdM-svm consistently achieves the best results. Overall, when considering



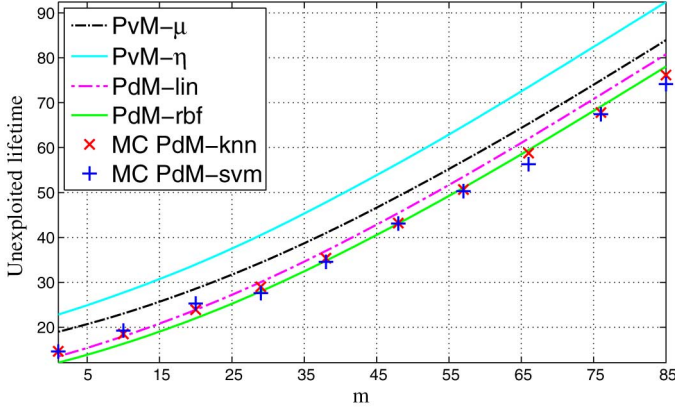


Fig. 4. Average  $\rho_{UL}$  over  $Q_2$  Monte Carlo simulations with the various maintenance management approaches.

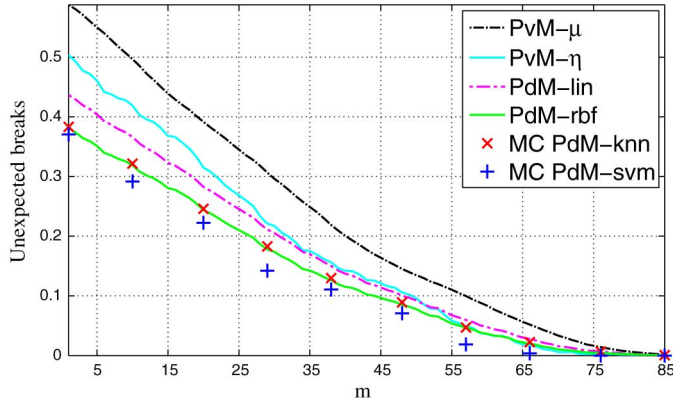


Fig. 5. Average  $\rho_{UB}$  over  $Q_2$  Monte Carlo simulations with the various maintenance management approaches.

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

$m$	$kNN$			$SVM$		
	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

both  $\rho_{UL}$  and  $\rho_{UB}$ , 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 various 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

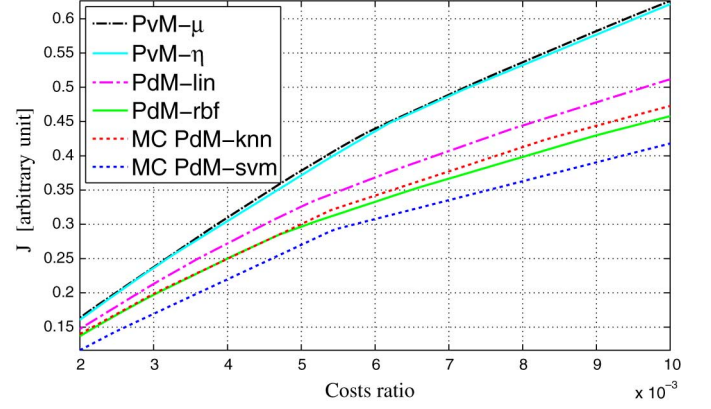


Fig. 6. Optimal value of  $J$  for the various maintenance strategies as a function of the ratio of the costs  $c_{UL}/c_{UB}$  (averaged results over  $Q_2$  Monte Carlo simulations).

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 Figs. 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 being robust to different choices of  $\rho_{UL}$  and  $\rho_{UB}$ . 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. CONCLUSION

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 minimizing operating cost and can be applied to any maintenance problems characterized by integral type faults provided R2F historical data are 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 than 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 minimizing 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 Trans. Semicond. Manuf.*, vol. 25, no. 3, pp. 373–383, Aug. 2012.
- [2] S. Butler and J. Ringwood, “Particle filters for remaining useful life estimation of abatement equipment used in semiconductor manufacturing,” in *Proc. IEEE Conf. Control Fault Tolerant Syst.*, Oct. 2010, pp. 436–441.
- [3] C.-C. Chang and C. J. Lin, “Libsvm: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [4] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009.
- [5] A. Heng *et al.*, “Intelligent condition-based prediction of machinery reliability,” *Mech. Syst. Signal Process.*, vol. 23, pp. 1600–1614, 2009.
- [6] G. Koksai, I. Batmaz, and M. C. Testik, “A review of data mining applications for quality improvement in manufacturing industry,” *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13448–13467, 2011.
- [7] L. Krishnamurthy *et al.*, “Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the north sea,” in *Proc. Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2005, pp. 64–75.
- [8] R. Langone, C. Alzate, B. DeKetelaere, and J. A. K. Suykens, “Kernel spectral clustering for predicting maintenance of industrial machines,” in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, pp. 39–45, Apr. 2013.
- [9] S. S. Lee, “Noisy replication in skewed binary classification,” *Comput. Stat. Data Anal.*, vol. 34, no. 2, pp. 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 *Proc. 46th IEEE Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2013, pp. 3447–3456.
- [11] S.-Y. Lin and S.-C. Horng, “A classification-based fault detection and isolation scheme for the ion implanter,” *IEEE Trans. Semicond. Manuf.*, vol. 19, no. 4, pp. 411–424, 2006.
- [12] M. Luo, Z. Xu, H. L. Chan, and M. Alavi, “Online predictive maintenance approach for semiconductor equipment,” in *Proc. 39th IEEE Annu. Conf. Ind. Electron. Soc. (IECON)*, 2013, pp. 3662–3667.
- [13] C. M. McKenna, “A personal historical perspective of ion implantation equipment for semiconductor applications,” in *Proc. Conf. Ion Implant. Technol.*, 2000, pp. 1–19.
- [14] R. K. Mobley, *An Introduction to Predictive Maintenance*. London, U.K.: 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,” *J. Sched.*, vol. 4, no. 6, pp. 583–599, 2011.
- [16] R. R. Picard and R. D. Cook, “Cross-validation of regression models,” *J. Amer. Stat. Assoc.*, vol. 79, pp. 575–583, 1984.
- [17] General Electric Intelligent Platforms, *The Rise of Industrial Big Data*, Aug. 2012.
- [18] A. Schirru, G. A. Susto, S. Pampuri, and S. McLoone, “Learning from time series: Supervised aggregative feature extraction,” in *Proc. 51st IEEE Conf. Decision Control*, 2012, pp. 5254–5259.
- [19] B. Scholkopf and A. J. Smola, *Learning With Kernels*. Cambridge, MA, USA: MIT Press, 2002.
- [20] J. Shao, “Linear model selection by cross-validation,” *J. Amer. Stat. Assoc.*, vol. 88, pp. 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. Semicond. Manuf.*, vol. 19, pp. 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. Semicond. Manuf.*, vol. 25, pp. 638–649, 2012.
- [23] G. A. Susto, S. Pampuri, A. Schirru, A. Beghi, and G. DeNicolao, “Multistep virtual metrology for semiconductor manufacturing: A multilevel and regularization methods-based approach,” *Comput. Oper. Res.*, to be published.
- [24] G. A. Susto *et al.*, “A predictive maintenance system for integral type faults based on support vector machines: An application to ion implantation,” in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, 2013, pp. 195–200.
- [25] J. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural Process. Lett.*, vol. 9, no. 3, pp. 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,” *IEEE Trans. Semicond. Manuf.*, vol. 26, no. 3, pp. 319–327, Aug. 2013.
- [27] M. E. Tipping, “Sparse bayesian learning and the relevance vector machine,” *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.
- [28] M.-L. Zhang and Z.-H. Zhou, “ML-KNN: A lazy learning approach to multi-label learning,” *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.



**Gian Antonio Susto** received the M.S. degree (*cum laude*) in control systems engineering and the Ph.D. degree in information engineering from the University of Padova, Padova, Italy, in 2009 and 2013, respectively.

He is currently a Postdoctoral Associate with the University of Padova, and a Chief Operating Officer with Statwolf Ltd., Dublin, Ireland. He was a Visiting Student at the University of California, San Diego, CA, USA, from 2008 to 2009, and at the National University of Ireland, Maynooth, Ireland, in 2012, and was an Intern Researcher with Infineon Technologies Austria AG, Villach, Austria, in 2011. His research interests include manufacturing data analytics, machine learning, gesture recognition, and partial differential equations control.

Dr. Susto has been recipient of the IEEE Conference on Automation Science and Engineering (IEEE-CASE) Best Student Conference Paper Award in 2011, the IEEE Semiconductor Equipment and Materials International Advanced Semiconductor Manufacturing Conference (IEEE/SEMI-ASMC) Best Student Paper Award in 2012, and the IEEE Multi-Conference on Systems and Control (IEEE-MSC) Best Student Paper Award in 2012.





**Andrea Schirru** received the M.S. degree (*cum laude*) in computer science and the Ph.D. degree in information engineering from the University of Pavia, Pavia, Italy, in 2008 and 2011, respectively.

He is currently a Postdoctoral Associate with the National University of Ireland, Maynooth, Ireland, and Chief Technology Officer with Statwolf Ltd., Dublin, Ireland. From 2010 to 2011, he was an Intern Researcher with Infineon Technologies Austria AG, Villach, Austria.

Dr. Schirru has been the recipient of the International Conference on Informatics in Control, Automation, and Robotics (ICINCO) Best Student Paper Award in 2011, and the Best Student Presentation at the Intel European Research and Innovation Conference in 2010.



**Simone Pampuri** received the M.S. degree in computer science and the Ph.D. degree in information engineering from the University of Pavia, Pavia, Italy, in 2009 and 2012, respectively.

He is currently a Postdoctoral Associate with the National University of Ireland, Maynooth, Ireland, and Chief Executive Officer with Statwolf Ltd., Dublin, Ireland. He was an Intern Researcher with the Infineon Technologies Austria AG, Villach, Austria, in 2011. His research interests include manufacturing data analytics, machine learning, and

sentiment analysis.



**Seán McLoone** (S'94–M'96–SM'02) received the M.E. and Ph.D. degrees in control engineering from Queens University Belfast (QUB), Belfast, U.K., in 1992 and 1996, respectively.

He is currently a Professor and Director of the Energy Power and Intelligent Control (EPIC) Research Cluster, QUB. His research interests include general area to data-based modeling 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.

Dr. McLoone is a Past Chairman of the U.K. and Republic of Ireland (UKRI) Section of the IEEE.



**Alessandro Beghi** (M'06) received the Laurea degree (*cum laude*) in electrical engineering and the Ph.D. degree in control systems engineering from the University of Padova, Padova, Italy, in 1989 and 1993, respectively.

In 1994, he joined the Department of Information Engineering, University of Padova, where he is currently an Associate Professor of Control Systems Theory. He held Visiting Positions with universities and international research centers both in Europe and in the USA. He is the author of more than 130 publi-

cations in journals, books, and conference proceedings. He is the Co-Inventor of five international patents on applications of advanced control techniques. His research interests include a wide range of control applications, from control of fusion devices and guidance algorithms for virtual vehicles, to control of heating, ventilation, air conditioning, and refrigeration (HVAC&R) systems; adaptive optics systems; and semiconductor manufacturing processes.

Dr. Beghi is an Associate Editor for the Conference Editorial Board and Co-Chair of the Technical Committee on Power Generation.