

## A Methodology for Predictive Maintenance in Semiconductor Manufacturing

Peter Scheibelhofer<sup>1,2</sup>, Dietmar Gleispach<sup>1</sup>, Günter Hayderer<sup>1</sup> and  
Ernst Stadlober<sup>2</sup>

<sup>1</sup>ams AG, Unterpremstätten, Austria

<sup>2</sup>Institute of Statistics, Graz University of Technology, Austria

**Abstract:** In order to occupy a competitive position in semiconductor industry the most important challenges a fabrication plant has to face are the reduction of manufacturing costs and the increase of production yield. Predictive maintenance is one possible way to address these challenges. In this paper we present an implementation of a universally applicable methodology based on the theory of regression trees and Random Forests to predict tool maintenance operations. We exemplarily show the application of the method by constructing a model for predictive maintenance of an ion implantation tool. To fit the problem adequately and to allow a descriptive interpretation we introduce the remaining time until next maintenance as a response variable. By using R and adequately analyzing data acquired during wafer processing a Random Forest model is constructed. We can show that under typical production conditions the model is able to predict a recurring maintenance operation sufficiently accurate. This example shows that better planning of maintenance operations allows for an increase in productivity and a reduction of downtime costs.

**Zusammenfassung:** Die wichtigsten Herausforderungen mit denen man in der Halbleiterindustrie konfrontiert ist, um wettbewerbsfähig zu bleiben sind die Reduktion von Herstellungskosten und die Erhöhung der Produktivität. Die prädiktive Wartung (engl. *Predictive Maintenance*) ist ein möglicher Weg um sich mit diesen Aufgaben zu befassen. In dieser Arbeit stellen wir die Umsetzung einer universell anwendbaren Methodik basierend auf der Theorie der Regressionsbäume und der Random Forests vor, um Maschinenwartungen vorhersagen zu können. Exemplarisch für die Anwendung der Methodik konstruieren wir ein Modell für die prädiktive Wartung einer Maschine zur Implantierung von Ionen. Für eine adäquate und interpretierbare Modellierung des Problems führen wir die verbleibende Zeit bis zur nächsten Wartung als Zielgröße ein. Mit Hilfe des Software-Pakets R und adäquater Datenanalyse konstruieren wir ein Random Forests Modell unter Verwendung von Daten, die während der Wafer-Prozessierung akquiriert wurden. Wir können zeigen, dass ein wiederkehrender Wartungseingriff unter typischen Produktionsbedingungen mit einer ausreichenden Genauigkeit vorhergesagt werden kann. Dieses Beispiel zeigt, dass die bessere Planbarkeit von Wartungsarbeiten den Weg für Steigerungen der Produktivität und die Reduktion von Stillstandskosten ebnet.

**Keywords:** Applied Statistics, Regression Trees, Random Forests, Predictive Fault Detection, Advanced Process Control.

## 1 Introduction

Among the most important challenges in a semiconductor fabrication plant are the reduction of nonproductive wafers and wafer defects as well as the increase of throughput and uptime of the production equipments. Achieving these objectives requires both, the implementation of a satisfying fault detection and classification environment and advanced process control (APC). One typical goal of APC is the transition from preventive to predictive maintenance, defined as a model-based prediction of equipment faults.

Usually, a production step in a semiconductor process contains information of many, often nonlinearly related production parameters. Hence, for implementing model-based fault prediction a multivariate method needs to be able to capture such complex relationships.

In this paper we present an implementation of a universally applicable methodology for predictive maintenance based on the use of classification and regression trees (CART, see Breiman, Friedman, Olshen, and Stone, 1984) for both data analysing and modelling purposes. CART models offer an intuitive overview of a multivariate data set and are suitable for dealing with complex processes and nonlinear relationships. They are also able to recognize the parameters that are most important to a given regression problem. However, they suffer from high prediction variance. Therefore, for prediction purposes we use a method that utilizes an ensemble of CART models called *Random Forests* (see Breiman, 2001). The aggregation of a large number of different single models usually offers improved prediction accuracy.

Furthermore, such tree-based methods are nonparametric and distribution-free. A classical modelling approach often needs specific parametric and distributional assumptions that can be restrictive for our modelling purposes. A nonparametric modelling methodology is able to avoid these problems. This way, the methodology can be used for a wider range of applications.

We exemplarily show how this methodology can be utilized for predictive maintenance tasks by applying it to predict a recurring maintenance operation on an ion implantation tool. Predicting this production-time-consuming operation accurately allows for specific maintenance scheduling. It reduces tool downtime and improves the productivity of the equipment.

In the following section the method of Random Forests is summarized and an overview of its inherent variable importance measures is given. In the third section we show how the methodology is applied to the prototype tool. A predictive model is presented and we use test data to evaluate its performance. Finally some concluding remarks will be made. For further reading on predictive failure detection see Scheibelhofer (2011).

## 2 Review of Random Forests

A Random Forest for regression consists of an ensemble (or a *forest*) of regression tree models. As in Bagging (or “**B**ootstrap **A**ggregating”, see Breiman, 1996), a Random Forest does not use all of the given observations for constructing each tree but only a bootstrap sample (see Efron, 1979). Additional randomness comes from using only a random sample of predictors for determining each split in each tree. With this Random

Forests aim to reduce the variance of CART's fitted values and to improve the prediction error. The method is also able to measure its own performance by using the observations not selected by the bootstrapping (*out-of-bag* or OOB samples) to test the model's predictive power and calculate error rates (*OOB error*).

The basic steps of the algorithm are the following (see Berk, 2008): Let the response be a continuous variable,  $n$  be the number of given observations and  $m_{try}$  be the number of predictors used for each split in each tree.

1. Draw a bootstrap sample of size  $n$  from the data (random sample drawn with replacement).
2. Take a random sample of size  $m_{try}$  without replacement of the predictors.
3. Construct the first regression tree partition of the data, i.e. the first split and repeat step 2 for each subsequent split in the tree. Do not prune.
4. Drop the OOB data down the tree and store the assigned value, i.e. the mean of the terminal node in which the observation falls.
5. Iterate the steps 1 to 4 a large number of times, e.g. 500.
6. Use only the predicted values assigned to each observation when that observation was an OOB observation (i.e. not used to build the tree) to calculate the MSE.

Aggregating the results of single tree models reduces variance and produces more stable models. Furthermore the method does not overfit due to the law of large numbers as is proved in Breiman (2001).

Unlike with CART there is no graphical model output to visualize results and variable importance ranking. Although there are several graphical methods that aim to compensate this drawback, the procedure remains a black box.

By drawing a bootstrap sample of size  $n$  from the data, on average about one third of the samples are not used to build the corresponding tree as stated in Breiman (2001). These OOB samples are used to test each tree and deliver an internal estimation of the test set error (see Breiman, 2001, p. 11). On average each data point is among the out-of-bag sample around 36 % of the time as mentioned in Liaw and Wiener (2002). Furthermore, the prediction error observed using OOB cases approaches the true prediction error as the number of trees goes to infinity.

## 2.1 Variable Importance

Following the discussions in Berk (2008), Sandri and Zuccolotto (2006), and Breiman (2001) the following measures for determining variable importance are common in the regression case.

- **Measure 1:** Measure the reduction in the deviance each time the predictor is used to define a split. The sum of these reductions can then be used as importance measure for one particular tree and the average of all reductions over all tree models

describes the predictor's importance. The importance measure  $\text{Imp}^{(1)}$  for the predictor  $x_i$  is therefore defined as

$$\text{Imp}_{x_i}^{(1)} = \frac{1}{k} \sum_A d(x_i, A) 1_{\{x_i \in A\}},$$

where  $A$  is a node in each tree,  $d(x_i, A)$  is the reduction in the deviance induced  $x_i$  at node  $A$  and  $1_{\{x_i \in A\}}$  is an indicator function which is equal to 1 if  $x_i$  is selected for a split at node  $A$ .

- **Measure 2:** In every grown tree in the ensemble, the OOB data cases are dropped down and the mean squared error is computed to assess the model error. Then the values of predictor  $i$  are randomly shuffled and the OOB cases are dropped down again. The shuffled predictor should now be on average unrelated to the response. Iterate this procedure for each of the  $p$  predictors and compute

$$\text{Imp}_{x_i}^{(2)} = \frac{1}{K} \sum_{k=1}^K (\text{MSE}_i^{(k)} - \text{MSE}^{(k)}), \quad i = 1, \dots, p$$

with  $K$  being the number of trees,  $\text{MSE}_i^{(k)}$  is the mean squared error of the  $k$ th tree calculated using out-of-bag data when the  $i$ th predictor values are shuffled and  $\text{MSE}^{(k)}$  is the general mean squared error of the  $k$ th tree without shuffling.

If desirable, one can normalize  $\text{Imp}_{x_i}^{(2)}$  with the standard deviation of the differences:

$$\text{Imp}_{x_i}^{(2)*} = \frac{\text{Imp}_{x_i}^{(2)}}{\text{sd}(\{\text{MSE}_i^{(k)} - \text{MSE}^{(k)}\}_{k=1}^K)},$$

with the division not being done if the standard deviation is 0.

Further discussions on predictor relevance can be found for example in Hastie, Tibshirani, and Friedman (2001).

## 3 Case Study: Predictive Maintenance on an Implanter Tool

### 3.1 Motivation

Ion implantation is a single process step which occurs several times during wafer fabrication and is typically one of the most complex ones. The ion implantation tool (implanter) is used to impinge charged atoms or molecules (ions) upon the wafer to systematically change electrical characteristics of the wafer surface (see Wolf, 2003). Therefore, the ions are generated in an ion source and extracted in form of an ion beam. One part of the ion source of an implanter is the filament. It is stressed during the implanter operation and breaks on a highly irregular basis every few days. Figure 1 shows an ion source and different filament conditions. The breakdown and the resulting tool downtime leads to

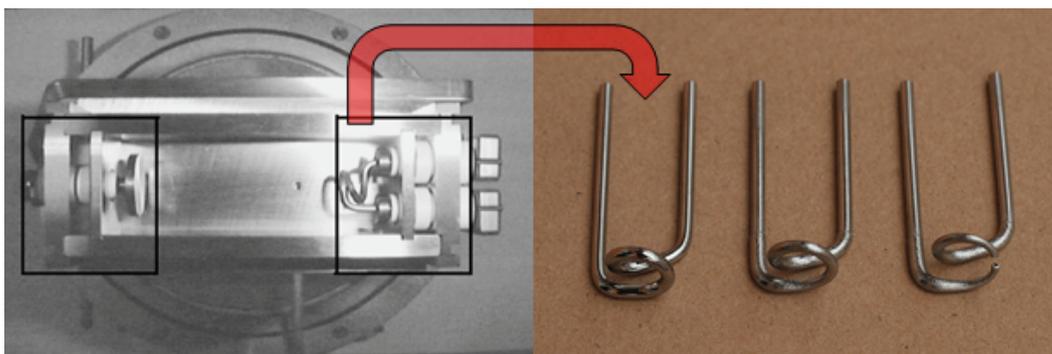


Figure 1: The ion source of an implanter tool (left) and different filament conditions.

a highly undesired loss in productivity. Thus, a well-defined point of time for changing the filament can increase the throughput and reduce downtime and maintenance costs. For example, knowledge of the remaining lifetime of the filament right before a weekend could prevent expensive weekend assignment of engineers and reduce manpower costs. Therefore we try to find a statistical model for predicting the filament break.

## 3.2 Data Analysis

Our initial data set consists of  $n_{initial} = 6781$  observations and  $p_{initial} = 20$  tool variables. The continuous nonconstant predictor variables are measured every time a new production unit starts on the implanter. All computation was done using R (see R Development Core Team, 2011). For the Random Forest analysis in R the downloadable `randomForest` package (see Liaw and Wiener, 2002) was used.

### 3.2.1 Response variable

For the initial set of historical process data a response variable suitable for the problem has to be assigned to every observation. As the exact time and date of each filament break is logged the remaining filament lifetime can be calculated for every historical data case. Thus, for our problem of predicting an imminent filament break we create the time-to-event response variable *NextPM*. For every observation in the historical data set, the corresponding *NextPM* value describes the exact hours left until the next recorded filament break. That means, the assigned response value describes the observed time span to the next maintenance event. These continuous values are highly intuitive and a prediction is easy to interpret for engineers.

In order to use a historical data set to fit a model for *NextPM* the raw data of  $n_{initial}$  cases have to be adjusted. Time periods in which the implanter tool was shut down completely (i.e. no stressing of the filament) or where the filament was changed due to regular maintenance operations have to be filtered out of the *NextPM* calculation. Furthermore, we only consider lifetime values of 120 hours or less as this is the time span of interest for process engineers.

After purging the data,  $n = 1812$  observations or seven filament lifetime cycles remain. Figure 2 shows the adjusted *NextPM* values as well as their distribution. The

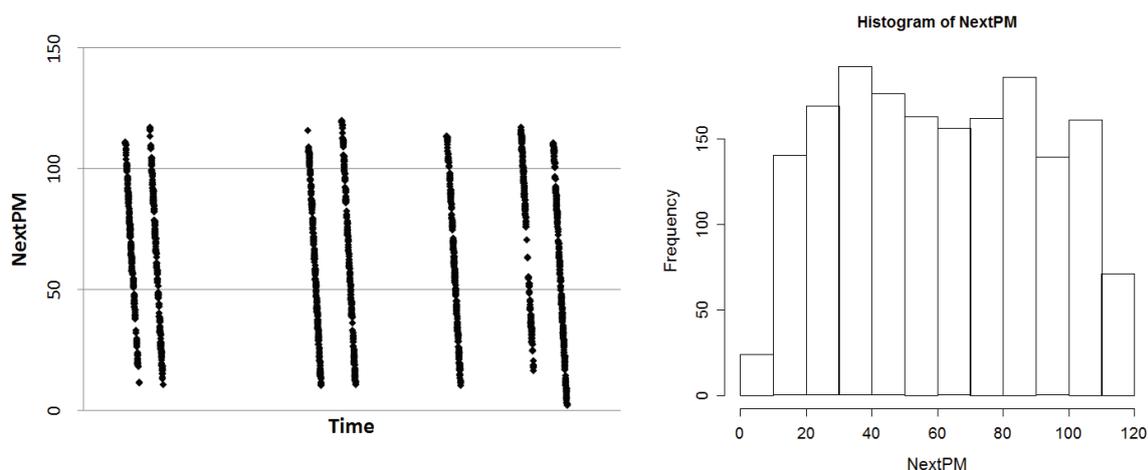


Figure 2: *NextPM* over time (left) and its sample distribution.

remaining gaps in the *NextPM* values are due to the implanter tool being idle but with the ion beam being activated, i.e. with the filament being stressed.

As we choose a Random Forest approach for modelling *NextPM* we can avoid any further distributional or parametric assumptions.

### 3.2.2 Predictors and their relationships

The  $i$ th *NextPM* value  $y_i$  can be seen as a mapping of the filament condition to the corresponding observation vector of  $p$  explanatory variables  $x_i = (x_{i1}, \dots, x_{ip})$ ,  $i = 1, \dots, n$ . Table 1 lists all initial variables along with a short description. Process engineers suggest that FIL\_I should be of high relevance for modelling the filament lifetime as it provides direct information of power running over the filament.

To determine contributing predictors we utilize the variable importance measures inherent to Random Forests as described in Section 2.2. This leads to Figure 3.

Another useful variable importance indicator are single regression tree models. Figure 4 shows the R output of a single CART model using the `rpart` package in R (for details on their construction see Therneau and J. (1997) and Hothorn and Zeileis (2012)). Similar results can be obtained by using trees based on unbiased recursive partitioning (see Hothorn, Hornik, and Zeileis, 2006).

The analysis suggests that FIL\_I is by far the most important predictor as also suggested by process engineers. Furthermore, GAS, EXT\_I, ION\_NAME, SUP\_I and BEAM can be regarded as top contributing variables. Further analysis of the change in a Random Forest model's OOB error induced by omitting one of the six predictors above shows that ION\_NAME is redundant due to no significant error reduction. Figure 5 shows the relationships of the remaining 5 predictors to *NextPM*. We denote the variable constellation FIL\_I, EXT\_I, SUP\_I, GAS, BEAM as  $M_0$ . FIL\_I is the only predictor with a linear relationship to *NextPM*. Process engineers suggested that the current FIL\_I value serves as a good indicator of the remaining filament lifetime. However, all other predictors show no clear visual relationship.

The filament condition can also be considered dependent on the accumulated stress

Parameter	Description
ARC_I	Current between filament and source chamber (in amperes)
ARC_VOLTS	Voltage of ARC_I (in volts)
BEAM	Power of the ion beam (in amperes)
BEAM_ENERGY	Total energy that reaches the wafer (in electronvolts)
BEAM_I_RANGE	Categorical (6 levels), specifies unit calculation for BEAM current
CHAMBER_PRESSURE	Pressure in the source chamber (in torr)
DELTA	Time elapsed since last data point (in hours)
EXT_I	Power coming out of the source (in milliamperes)
EXT_VOLTS	Voltage of EXT_I (in kilovolts)
EXTRACTION	Target value of EXT_VOLTS (constant)
FIL_I	Current streaming over the filament (in amperes)
GAS	Pressure of gas bottle for storing the elements (in torr)
ION_NAME	Categorical (5 levels), atomic mass unit of element to implant
SCANNER_PRESSURE	Pressure of the scanner (in torr)
SOURCE_PRESSURE	Vacuum value, must correspond with GAS (in torr)
SUP_I	Current filtered out of extraction power (in milliamperes)
SUP_VOLTS	Voltage of SUP_I (in kilovolts)
X_AXIS	x axis position of source and extraction blind, values from 0-999
Y_AXIS	y axis position of source and extraction blind, values from 0-999
Z_AXIS	z axis position of source and extraction blind, values from 0-999

Table 1: Overview and descriptions of the predictors measured on an implanter tool.

over its lifetime. Therefore we also examine the effect of accumulation of variable measurements so far over a filament lifetime cycle. This should provide information on how machine usage up to the current observation point affects the filament lifetime. The associated variable importance analysis suggests that the accumulated version of SUP\_I (s.SUP\_I) is important. Furthermore, with the presence of accumulated variables in the tree-based interaction structure the importance of FIL\_I decreases. However, adding the accumulated variables to a model does not necessarily improve its practical use as will be shown in the next section (see Table 2).

### 3.3 Final Model

In order to determine the final model for our purged training data set, Random Forest models with a number of different variable constellations have been evaluated in terms of their prediction errors (RMSE). In order to test the practical performance of the models we use a separate test data set consisting of  $n_{test} = 674$  observations or two filament lifetime cycles. Table 2 shows a selection of the models considered with error rates for the training set (RMSE train), the test set (RMSE test) and an error rate smoothed with moving average of 5 (RMSE test MA(5)). The models have been evaluated with `set.seed(1)`, 1000 single tree models and the  $m_{try}$  value that returns the lowest OOB error for each model as determined by the function `tuneRF()` from the `randomForest` package.

Obviously, adding the accumulated variables only improves the RMSE on the training set, but the test cases mostly yield RMSE values worse than in the base model  $M_0$ .

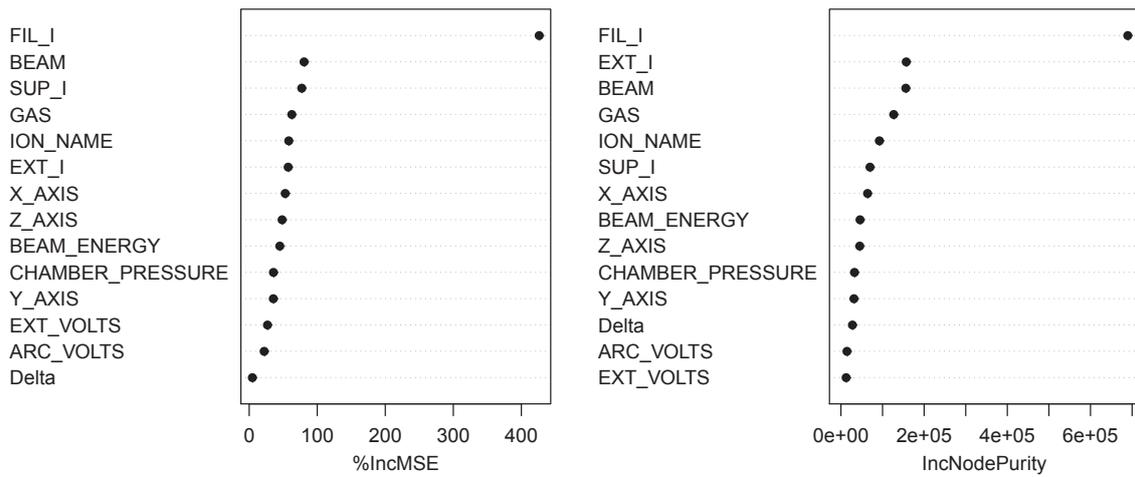


Figure 3: Importance measures  $\text{Imp}_{x_i}^{(1)}$  (left) and  $\text{Imp}_{x_i}^{(2*)}$  of a Random Forest model.

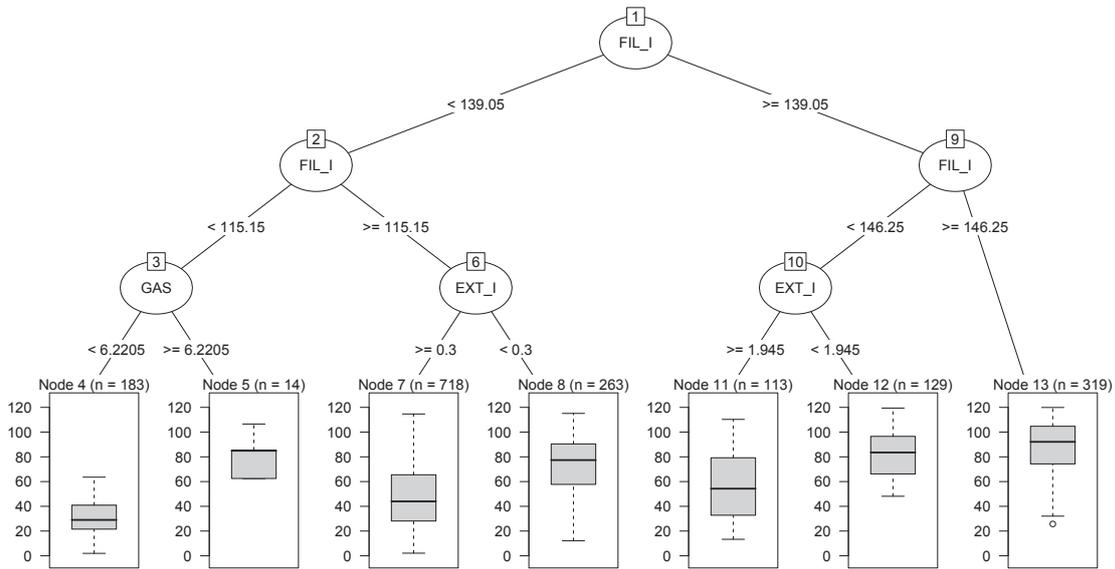


Figure 4: Regression tree model constructed using binary recursive partitioning routines as implemented in the R package rpart and plotted using routines from the R package partykit.

However, adding the accumulated measurements  $s.SUP\_I$  gives error rate improvements compared to  $M_0$ . The constellation  $M_{10}$  (five variables) yields a better training error (7.85 vs. 13.2 hours) and test error rate (11.18 and 10.78 hours) by replacing  $SUP\_I$  with its accumulated version  $s.SUP\_I$ .

Thus we use  $M_{10}$  as final model. The  $m_{try}$  value resulting in the lowest OOB error is calculated to be  $m_{try} = 4$ . The model plot on the right hand side of Figure 6 shows that a number of 400 trees is reasonable. More trees do not yield a significant OOB error reduction.

A model with  $m_{try} = 4$  and 400 trees explains about 93 % of the variance of  $NextPM$  and its RMSE is 7.85 hours.

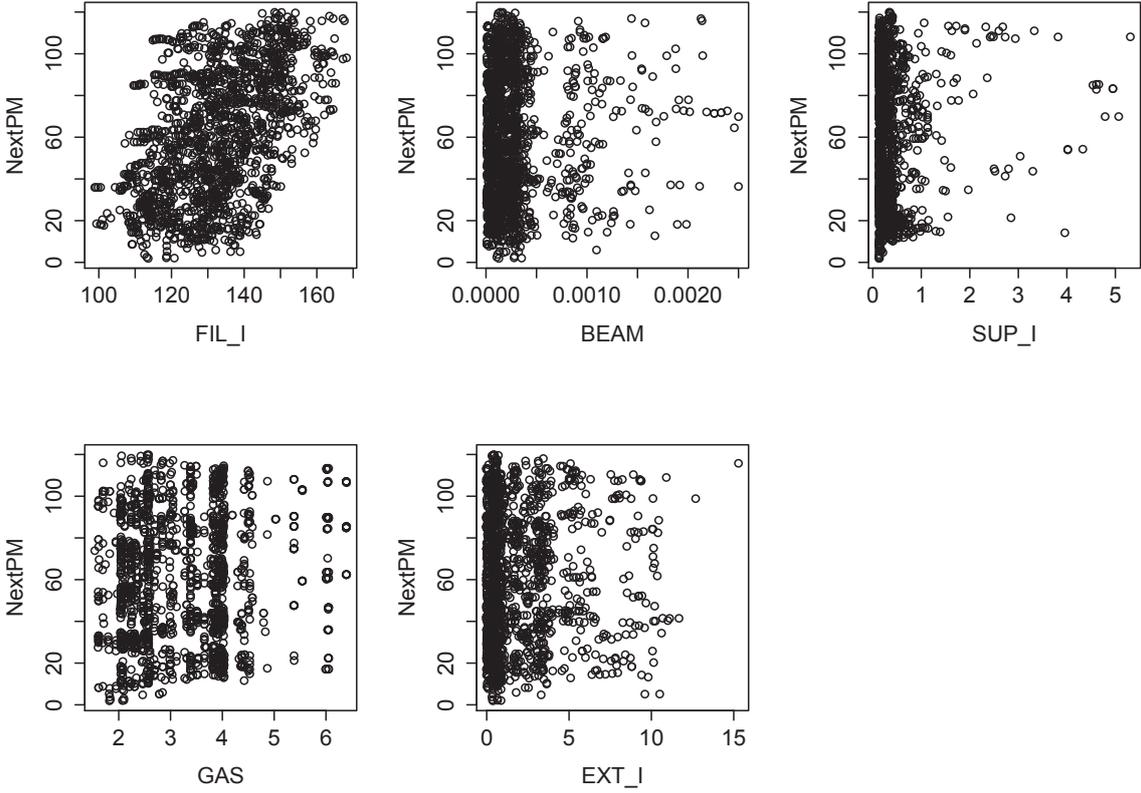


Figure 5: Scatterplots of the continuous predictors FIL\_I, BEAM, SUP\_I, GAS and EXT\_I with the response *NextPM*.

Furthermore, the model serves as a statistic for the remaining lifetime of a filament. Based on the constructed model, the  $i$ th future observation  $\tilde{x}_i = (\tilde{x}_{i1}, \dots, \tilde{x}_{i5})$  of the contributing five predictors (with one accumulated predictor) can be assigned a predicted value  $\hat{y}_i$ . This  $\hat{y}_i$  is always calculated by using the trained model i.e. by averaging over the prediction of the single tree ensemble as determined above.

### 3.4 Results

Using the constructed lifetime statistic which is our trained model, the  $i$ th predicted value  $\hat{y}_i^{test}$  is the result of the model predictions of the  $i$ th test observation  $x_i^{test} = (x_{i1}^{test}, \dots, x_{i5}^{test})$ . Given  $x_i^{test}$ , the corresponding prediction  $\hat{y}_i^{test}$  is the terminal node prediction that  $x_i^{test}$  is assigned on average over the constructed ensemble of 400 trees.

The coefficient of determination  $R^2$  of the model is 0.88. It is estimated internally by

$$R^2 = 1 - \frac{1}{n_{test}} \frac{\sum_{i=1}^{n_{test}} (y_i^{test} - \hat{y}_i^{test})^2}{\text{Var}(y_i^{test})},$$

where  $y_i^{test}$  is the observed *NexPM* value corresponding to  $x_i^{test}$ . By taking moving average values (MA) over the last five fitted values instead of the fitted values itself the resulting root mean squared error can be reduced to 10.78 hours. In the critical time frame of

Model	Variables in the model	RMSE			#variables
		train	test	test MA(5)	
M0	FIL_I, EXT_I, SUP_I, GAS, BEAM	<b>13.20</b>	<b>14.61</b>	<b>13.58</b>	5
M1	s.FIL_I, EXT_I, SUP_I, GAS, BEAM	3.84	18.13	18.18	5
M2	s.FIL_I, s.EXT_I, SUP_I, GAS, BEAM	2.13	20.58	20.66	5
M3	s.FIL_I, s.EXT_I, s.SUP_I, GAS, BEAM	1.37	19.73	19.81	5
M4	s.FIL_I, s.EXT_I, s.SUP_I, s.GAS, BEAM	1.22	19.49	19.62	5
M5	s.FIL_I, s.EXT_I, s.SUP_I, s.GAS, s.BEAM	1.09	19.25	19.39	5
M6	FIL_I, s.EXT_I, s.SUP_I, s.GAS, s.BEAM	1.11	17.15	17.33	5
M7	FIL_I, EXT_I, s.SUP_I, s.GAS, s.BEAM	1.53	15.66	15.89	5
M8	FIL_I, EXT_I, SUP_I, GAS, BEAM, s.FIL_I, s.EXT_I, s.SUP_I, s.GAS, s.BEAM	1.17	18.94	19.09	10
M9	FIL_I, s.EXT_I, SUP_I, GAS, BEAM	7.59	21.82	21.68	5
M10	FIL_I, EXT_I, s.SUP_I, GAS, BEAM	<b>7.85</b>	<b>11.18</b>	<b>10.78</b>	5
M11	FIL_I, EXT_I, SUP_I, s.GAS, BEAM	4.10	15.29	15.52	5
M12	FIL_I, EXT_I, SUP_I, GAS, s.BEAM	6.36	19.29	19.25	5
M13	FIL_I, EXT_I, SUP_I, GAS, BEAM, s.ARC_VOLTS	3.39	16.87	17.06	6
M14	FIL_I, EXT_I, SUP_I, GAS, BEAM, s.BEAM_ENERGY	4.46	12.13	12.40	6
M15	FIL_I, EXT_I, SUP_I, GAS, BEAM, s.EXT_VOLTS	3.49	13.74	13.95	6
M16	FIL_I, EXT_I, SUP_I, GAS, BEAM, s.CHAMBER_PRESSURE	8.20	21.93	21.67	6
M17	FIL_I, EXT_I, SUP_I, GAS, BEAM, s.BEAM_ENERGY, s.EXT_VOLTS	2.76	12.88	13.14	7
M18	s.FIL_I, s.EXT_I, s.SUP_I, s.ARC_VOLTS, s.BEAM	1.08	19.57	19.69	5
M19	FIL_I, EXT_I, s.SUP_I, GAS, BEAM, s.EXT_I	5.08	18.62	18.64	6
M20	FIL_I, s.EXT_I, s.SUP_I, GAS, BEAM	5.04	18.78	18.80	5

Table 2: Results of Random Forest models with different variable constellations. Accumulated variable measurements are denoted s.name, e.g. s.SUP\_I. All RMSE values are in hours.

72 hours before the actual filament break the root mean squared error is 10.8 hours. An error of this magnitude makes the constructed model applicable for tool monitoring and predictive maintenance on the implant.

A graphical comparison of observed test response and the model output can be observed in Figure 7.

In the first cycle (left plot in Figure 7) we observe a low prediction variance for all filament lifetimes and an accurate prediction overall. For the second cycle the prediction variance is constantly higher with mostly under-estimation and an over-estimation of the prediction near the filament breakdown.

Long-running real time testing of the model has shown that several consecutive predictions below 30 hours indicate that a filament breakdown is imminent.

For further improvements of the prediction one can expand the model by applying

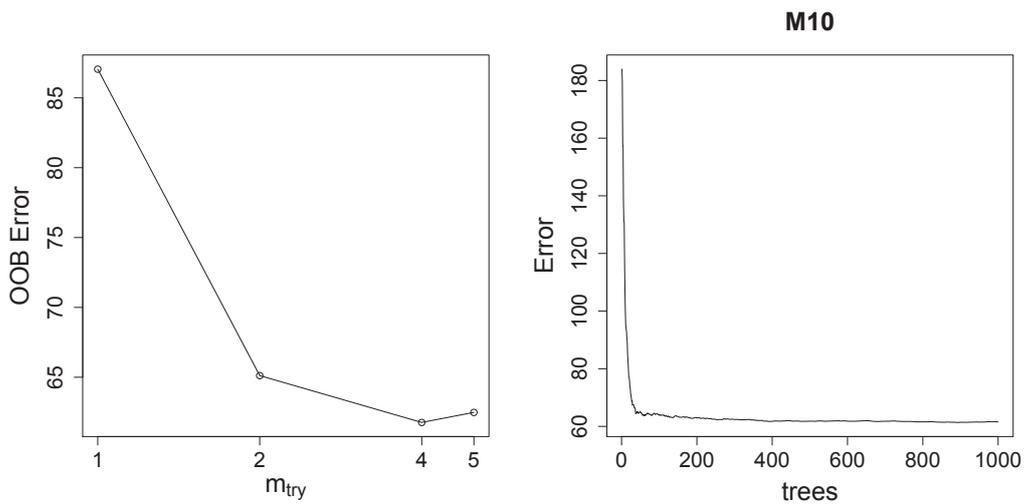


Figure 6: Values of  $m_{try}$  (left) and the number of trees against the OOB error rate.

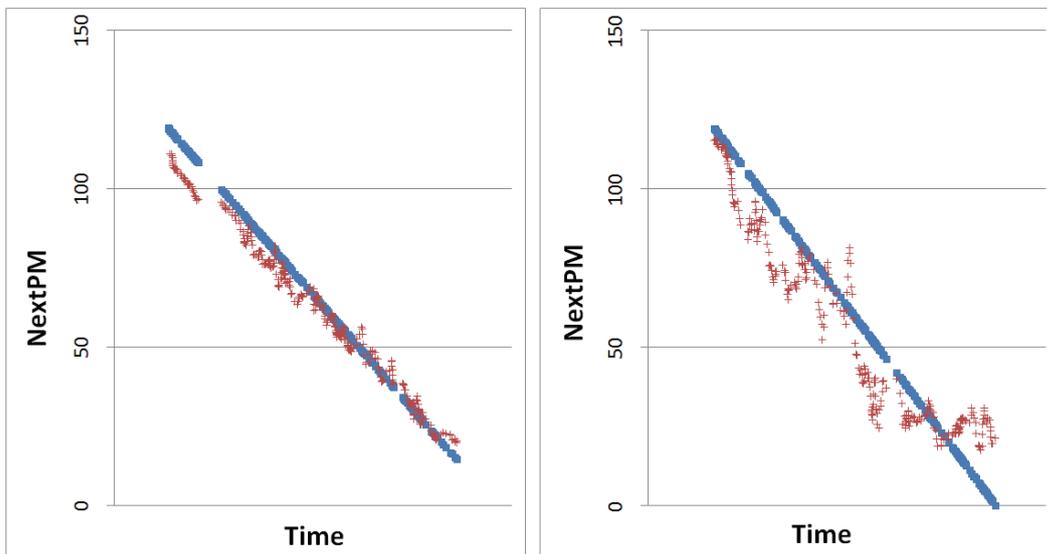


Figure 7: Graphical comparison of the observed values of *NextPM* from the test set (represented as quadrangles) and the MA over the last five predicted values for two filament lifetime cycles (cross symbols)

quantile regression Forests as introduced in Meinshausen (2006). This method allows for a more specific prediction in that it is able to estimate its reliability with prediction intervals.

In order to apply the model in a production environment its corresponding R output is integrated in an automated framework as it is described in Schellenberger et al. (2011).

## 4 Conclusion

In this work we presented an exemplary implementation of a methodology for advanced process control based on regression trees and Random Forests. The methodology allows a transition from a time-based to a condition-based maintenance, a reduction of problem complexity and it offers high predictive performance. As the Random Forest approach is free of parametric or distributional assumptions, the method can be applied to a wide range of predictive maintenance problems. We exemplarily implemented the approach on an ion implantation tool where a standard maintenance operation, namely the breakdown of the filament part, can be predicted with a satisfying accuracy for production needs. This proves to be highly useful for applying predictive maintenance in wafer production and equipment control. The implementation of the model in the production environment offers the possibility to specifically schedule maintenance operations. This leads to a reduction of tool downtime, maintenance and manpower costs and improves competitiveness in the semiconductor industry.

## Acknowledgement

Part of this work was supported by ENIAC project IMPROVE (Implementing manufacturing science solutions to increase equipment pROductiVity and fab pERformance). Funding by the EU and the FFG is gratefully acknowledged.

We would also like to thank an anonymous referee for all his valuable comments and suggestions that resulted in an improved version of our paper.

## References

- Berk, R. A. (2008). *Statistical Learning from a Regression Perspective*. Springer, New York.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123-140.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5-32.
- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, California.
- Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7, 1-26.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer, New York.
- Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: a conditional inference framework. *Journal of Computational and Graphical Statistics*, 15, 651-674.
- Hothorn, T., and Zeileis, A. (2012). partykit; a toolkit for recursive partitioning [Computer software manual]. Available from <http://CRAN.R-project.org/package=partykit> (R package version 0.1-3)
- Liaw, A., and Wiener, M. (2002). Classification and regression by randomForest. *R News*, 2, 18-22.

- Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, 7, 983-999.
- R Development Core Team. (2011). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Available from <http://www.R-project.org/> (ISBN 3-900051-07-0)
- Sandri, M., and Zuccolotto, P. (2006). Variable selection using random forests. In S. Zani, A. Cerioli, M. Riani, and M. Vichi (Eds.), *Data Analysis, Classification and the Forward Search. Proceedings of the Meeting of the Classification and Data Analysis Group (CLADAG) of the Italian Statistical Society, University of Parma, June 6-8, 2005*. Springer, Berlin.
- Scheibelhofer, P. (2011). *Tree-based Methods for Predictive Failure Detection in Semiconductor Fabrication*. Unpublished master's thesis, Institute of Statistics, Graz University of Technology.
- Schellenberger, M., Roeder, G., Mattes, A., Pfeffer, M., Pfitzner, L., Knapp, A., et al. (2011). Developing a framework for virtual metrology and predictive maintenance. *Future Fab International*, 39, 32-37.
- Therneau, T. M., and J., A. E. (1997). *An introduction to recursive partitioning using the rpart routines* (Tech. Rep. No. 61). Department of Health Science Research, Mayo Clinic, Rochester.
- Wolf, S. (2003). *Microchip Manufacturing*. Lattice Press, Sunset Beach.

Authors' addresses:

Peter Scheibelhofer, Dietmar Gleispach, Günter Hayderer  
ams AG

Tobelbaderstraße 30  
A-8141 Unterpremstätten  
Austria

E-Mails: [peter.scheibelhofer@ams.com](mailto:peter.scheibelhofer@ams.com), [dietmar.gleispach@ams.com](mailto:dietmar.gleispach@ams.com),  
[guenter.hayderer@ams.com](mailto:guenter.hayderer@ams.com)

Homepage: <http://www.ams.com>

Ernst Stadlober  
Institute of Statistics  
Graz University of Technology  
Kopernikusgasse 24/III  
A-8010 Graz

E-Mail: [E.Stadlober@TUGraz.at](mailto:E.Stadlober@TUGraz.at)

Homepage: <http://www.statistics.tugraz.at>