Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

VIRTUAL EQUIPMENT FOR BENCHMARKING PREDICTIVE MAINTENANCE ALGORITHMS

Andreas Mattes
Ulrich Schöpka
Martin Schellenberger

Fraunhofer IISB
Schottkystrasse 10
D-91058 Erlangen, GERMANY

Peter Scheibelhofer
Günter Leditzky

ams AG
Tobelbader Strasse 30
A-8141 Unterpremstaetten, AUSTRIA

ABSTRACT

This paper presents a comparison of three algorithm types (Bayesian Networks, Random Forest and Linear Regression) for Predictive Maintenance on an implanter system in semiconductor manufacturing. The comparison studies are executed using a Virtual Equipment which serves as a testing environment for prediction algorithms prior to their implementation in a semiconductor manufacturing plant (fab). The Virtual Equipment uses input data that is based on historical fab data collected during multiple filament failure cycles. In an automated study, the input data is altered systematically, e.g. by adding noise, drift or maintenance effects, and used for predictions utilizing the created Predictive Maintenance models. The resulting predictions are compared to the actual time-to-failure and to each other. Multiple analysis methods are applied, resulting in a performance table.

1 INTRODUCTION

There has been significant research in the last decade about the introduction of predictive models in semiconductor manufacturing. This includes areas such as Advanced Process Control (APC) but also Virtual Metrology (VM) and Predictive Maintenance (PdM). In the case of VM and PdM there have been few actual implementations in fabs up to now. The Virtual Equipment has been developed to make it easier to evaluate the performance of the algorithms by comparing the predictions of different models in an automated and systematic way. This includes the possibility to alter test data by a combination of physical and statistical simulation.

In this paper the focus will be on the comparison of PdM models on the filament failure prediction of an implanter and their sensitivity to noise in the input data.

2 VIRTUAL EQUIPMENT FOR EVALUATION OF PDM MODELS

2.1 Predictive Maintenance: Concept and Definition

The pressure to reduce production costs forces manufacturers to optimize equipment utilization. In semiconductor plants operation is already extended to a 24 hours per day / 7 days per week continuous manufacturing. Classical time based preventive maintenance minimizes unscheduled equipment downtime and scrap production. To further optimize the resource usage the preventive maintenance action should be done at the latest possible time but when the equipment is still performing in its intended way.

While the concept of Predictive Maintenance is already discussed for several years on a broader cross-industrial level (Mobley 2002, Dekker 1996, Liu 2008), tools in semiconductor processing are usu-

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

ally still controlled only by methods of Statistical Process Control (SPC) and Fault Detection and Classification (FDC), but these techniques do not allow prediction of tool failures. Predictive Maintenance is usually not yet implemented in semiconductor processing due to the complexity of the used processes, but is supposed to be crucial to achieve major cost reductions to fulfill Moore's law. Therefore, the ENIAC project IMPROVE was started to find (among other topics) best practice methods for the implementation of Predictive Maintenance in European semiconductor manufacturing sites.

Typical semiconductor manufacturing tools are equipped with hundreds of sensors to control all components of the tool and to achieve stable and reproducible conditions during the processing of a wafer. In many cases interactions of the components are observed but not well understood. In addition the variance of the sensor data over a long time period including maintenance events typically do not follow a normal distribution. In the course of the IMPROVE project, several different methods and algorithms for PdM modeling have been proposed and tested (e.g. Schirru 2010, Scheibelhofer 2011, Susto 2011) utilizing a broad selection of machine learning and data mining methods.

Bayesian Networks are already widely considered as suitable for the prediction of equipment condition (e.g. see Vachtsevanos et al. 2006) while methods based on decision trees are rather new for prediction purposes. Both methods are robust with respect to the underlying statistical distributions and consider possible predictor interactions. For this reason Random Forests and Bayesian Networks with Soft Discretization have been applied to the use case of the prediction of filament breakdown in ion implantation. For reference they are compared to a simple linear regression model.

2.2 Description of Virtual Equipment

The aim of the Virtual Equipment is to test control methods, e.g. for Predictive Maintenance, in regards to different analysis methods before implementing them in the fab. This is done by applying different statistical and physical simulation methods to generate test data based on historical fab data (for details on the simulation part of the Virtual Equipment, e.g. for Virtual Metrology, see Mattes et al. 2011). The test data sets are then split up into learning and test data. The algorithms use the learning data to teach a statistical model which is then used to predict the quality parameters of the test data. The predictions are finally analyzed and compared to both each other and the reference data in the original data set. This paper only uses a subset of the functionality of the Virtual Equipment which does not include physical simulation but only statistical alterations of the historical fab data to evaluate the sensitivity of the VM algorithms to noise.

The Virtual Equipment is implemented in MATLAB/Simulink. The physical/statistical simulation is done in Simulink with some MATLAB callbacks while the GUI and analysis uses MATLAB. The prediction models are considered as black boxes and interfaced via wrapper classes that mainly provide methods for updating the model (learning) and making a prediction.

3 DESCRIPTION OF THE USE CASE

3.1 Problem Description and Motivation

In semiconductor manufacturing ion implantation is the process step where the electrical properties of the silicon substrate are adjusted. The implanted ion species and the dopant level are responsible for the precise definition of the threshold voltage of the integrated n- and p-type transistors and are one of the most critical steps during the manufacturing process.

The implanter tool is highly complex. Its core part as depicted in Figure 1 is operated in high vacuum. It consists of an ion source where the ionization of the implant species takes place, a beam line through which the ions are accelerated and directed towards the end station chamber where the wafer in process is positioned. A more detailed description of the ion implantation process can be found in (Ryssel and Ruge 1978; Wolf 2003, chapter 12). Depending on the process requirements the ions are implanted in a wide range of dose and energies ranging over several orders of magnitude. This leads to an unequal degrada-

tion of the tool components depending on process recipe and is a challenge for state-of-the-art preventive maintenance.

The part which requires most frequent maintenance action is the filament of the ion source – in our case a small tungsten helix which emits electrons. During operation the filament degrades until it breaks. Its lifetime is not only determined by the utilization of the tool but also by the operation condition and the selected recipes.

Motivation for this work is to optimize the usage of the filament, that is to use it as long as possible but to replace it before it breaks. Therefore we model the remaining lifetime of the filament in hours by directly using sensor values as predictors. This is seen as a regression problem with continuous response variable y where the i*th* response value y[i] represents a mapping of the filament condition to the corresponding predictor observation vector x[i].

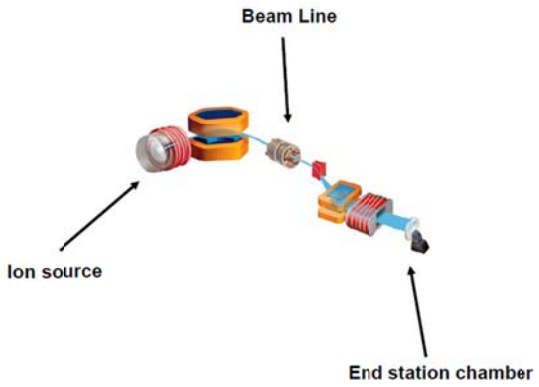

Figure 1: Diagram of the beam line of an medium current implanter tool.

3.2 Description of the Modeling Methods

3.2.1 Variable Selection

The amount of acquired data is limited by the hardware performance of the equipment interface. Hence the first step is to restrict the collected sensor data to a set of potentially influential parameters. This is done by analyzing the technical set up of the investigated component and historical failure modes. For these remaining parameters data collection plans were set up and data were acquired within the existing Fault Detection and Classification (FDC) system. In a second step the parameter set was further reduced by utilizing explorative statistical methods and importance measures based on regression trees (see Breiman et al., 1984; Breiman, 2001 and Hothorn et al. 2006). The results in a ranking of the parameters according to their importance. Table 1 gives an overview of the final parameter selection.

Table 1: Description of the used predictor variables.

Variable Name	**Description**
Fil-I	Filament current, current flow inside the filament
Fil-U	Filament voltage, voltage applied to filament
GAS	Pressure of gas bottle for storing the elements
Ext-I	Extraction current, the current of ions leaving the source
Sup-I	The current at the suppression electrode

3.2.2 Bayesian Networks with Soft Discretization

3.2.2.1 Bayesian Networks Technique

Due to the fact that all variables (predictors, response) are represented as "nodes", and all conditional dependencies between those variables/nodes are represented in (directed) arrows between the nodes, Bayesian Networks are often referred to as a graphical modeling method (Pearl 1988). So models can be set up in a graphical way, defining all nodes and arrows with respect to the real relationships between the predictor and response variables. After model learning, a probabilistic prediction can be made based on new data.

For the Predictive Maintenance system described here, a simple network structure was chosen with only few predictors (the first 4 selected variables from Table 1), and the remaining time to failure (TTF) as response variable (Figure 2). The model structure was determined manually utilizing expert knowledge of process and maintenance personnel. For modeling, MATLAB and a modified version of Kevin Murphy's BNT tool box (Murphy 2001) have been used.

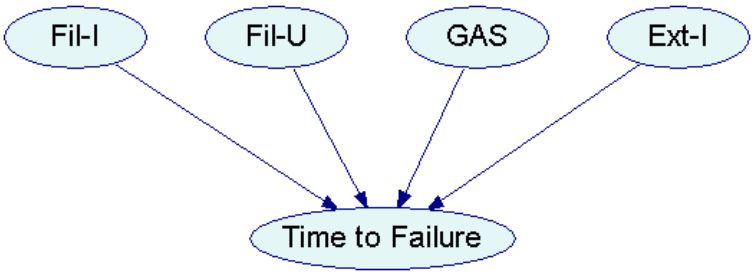

Figure 2: Bayesian Network model for the prediction of the filament breakdown in the Bernas ion source of the implanter

3.2.2.2 Soft Discretization

When using Bayesian Networks, the data usually has to be discretized. This enables the use of faster learning and inference algorithms. For discretization, the data space of a variable is divided into a finite number of states, typically 10 states for the case study presented in this paper. This causes a trade-off between accuracy and required computational power. Due to the fact that the model will not be able to make predictions for combinations of discrete input parameter states that were not explicitly learned, soft discretization was introduced (Ebert-Uphoff 2009).

During standard discretization, all data points are translated into discrete states. The model outcome will be a discrete probability distribution for the defined output node given a certain combination of states in the input nodes. Soft discretization enables to feed the model not only with discrete input data but with a probability distribution over all neighboring states to introduce probability already during learning. This eliminates the problem with combinations of input parameter values that have not been explicitly learned because all possible combinations already appeared, even with low probability. Therefore, soft discretization enables extrapolation of the model outcome and the handling of unknown tool conditions. Due to the same reason, the amount of training data needed for model learning can be greatly reduced by soft discretization.

In addition to the BNT toolbox, the soft discretization toolbox by Imme Ebert-Uphoff was used (Ebert-Uphoff 2009).

3.2.2.3 Reconstruction of Continuous Response from Discrete Probabilistic Output

The output of a Bayesian Network is a distribution over several discrete states. When the nature of the response variable is continuous, then the real signal has to be reconstructed from this. Different methods have been used and compared in this work:

Most probable state reconstruction:
This is the most simple way of reconstruction. For every inference result from every instance in the test data, the state with the highest probability is chosen. These states are then replaced by values that best represent the chosen state in the continuous data space. For calculation of the representative values, all n=10 discretization bins have been divided into q=10 quantiles. As representative value for bin i, the i*th* quantile is chosen. The resulting values are then smoothed using exponential smoothing.

Probability-weighted reconstruction:
For this method, the weighted sum over all representative values is calculated for reconstruction of a continuous value. As weights, the probability for every state from the probabilistic model output are used. The representative values are calculated as described for the most probable state method. Additionally, an exponentially smoothed version is tested in chapter 4.

Linear regression reconstruction
For this method, inference on the learn data is made. The resulting probabilities and the time to failure are used to learn a regression model that is then used for reconstruction utilizing the probabilistic predictions for new observations as input values for the regression model. Again, a smoothed and an unsmoothed version were used for the comparison in chapter 4.

3.2.3 Random Forest

A Random Forest (RF) model (see Breiman 2001) for regression consists of an ensemble of regression tree models (CART models, see Breiman et al. 1984). Out of the initial set of given observations a RF model does not use all of these 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, RF models 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
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). Un-

like 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.

For a detailed description of using RF for predictive failure detection on an implanter tool see Scheibelhofer (2011).

3.2.4 Linear Regression

Linear Regression attempts to fit a linear model on the relation between the predictors and the target value by computing coefficients that minimize the sum of squared residuals on the learning data set. In the case of this paper the sensor values are directly used as predictors. Predictions are calculated as the sum of the multiplication of the coefficients with their associated predictor value and a constant term: $P(X) = CX + C0$. In this paper Linear Regression is mainly used as a simple reference method and for that reason no second-order or interaction terms are used.

3.3 Virtual Equipment Settings and Assessment Goals

The input data used has been collected from the implanter over 8 filament failure cycles. It is split up by the Virtual Equipment into 60% learning data and 40% test data. The original order of the cycles is kept.

The PdM model variants used are:

- Bayesian Network with soft discretization and 3 different reconstruction methods both with and without smoothing of the results
- Random Forest
- Linear Regression without transformation of the predictors

The Linear regression and Bayesian Network models have been implemented in MATLAB while the Random Forest model is implemented in R (see R project website) and interfaced to the Virtual Equipment in MATLAB. The implementation of the linear regression uses the glmfit function of the MATLAB Statistics Toolbox.

Automated Studies are executed with noise on the sensor data ranging from the noise in the input data set in 15 steps up to additional noise with an amplitude of 15% of the input data value. The input variables chosen for the different PdM models are the same five predictors, with the exception of the Bayesian Network PdMs, which does not utilize Sup-I due to calculation time restrictions (the size of the conditional probability table and the time for its calculation increases exponentially with the number of directly connected nodes).

4 RESULTS AND DISCUSSION

Linear Regression, Random Forest and 5 Bayesian Network PdMs with different reconstruction methods are run in a study in the Virtual Equipment with added noise ranging from 0% to 15%. The input fab data is altered with the added noise and then split into 60% learning data and 40% test data. All algorithms use the same data set for predictions and the result is then analyzed and compared to the actual time to fail in the input data (see Figure 3 for a comparison of the actual time to fail and two of the predictions). Figure 4 shows the root-mean-square error (RMSE) of the predictions on the learning data and the test data. Random Forest is shown to adapt the most to the learning data and does the best prediction on that but the performance on the test data set is considerably worse. Linear Regression actually performs worse on the

learning data than the test data which is mostly because of a part in the learning data where it predicts negative time to fail which could be cut off to improve performance. The Bayesian Network variants use the same soft discretization and therefore the predicted probabilities are the same. The difference is in the way the probabilistic values are reconstructed into a non-probabilistic time to fail prediction and if the results are smoothed over time.

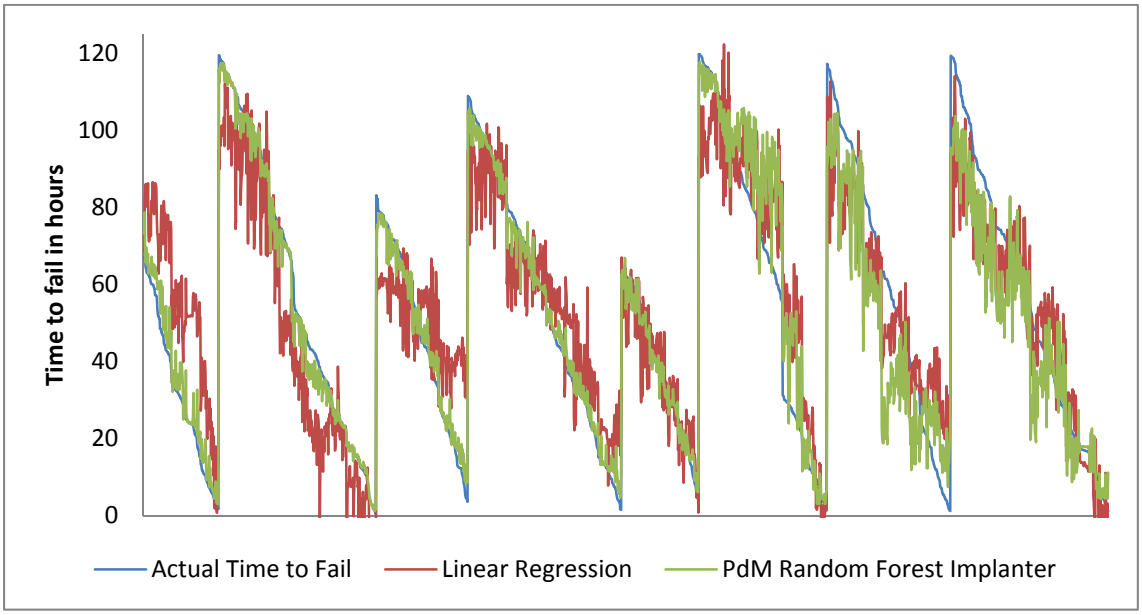

Figure 3: The predictions of two models over the filament failure cycles

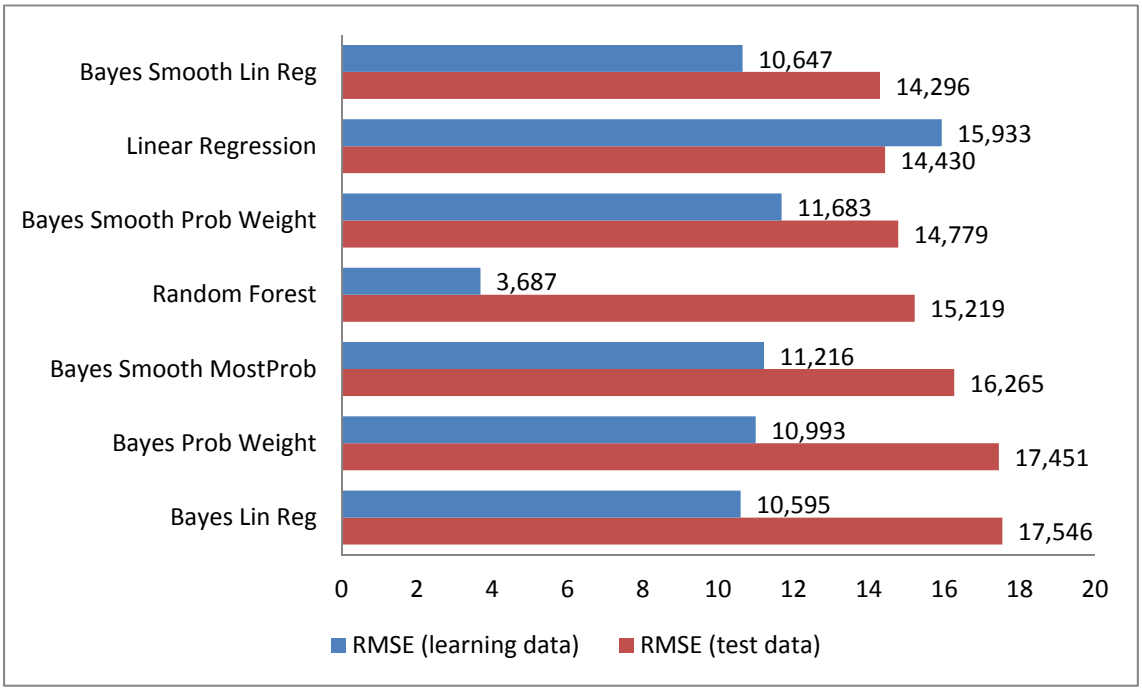

Figure 4: Root-mean-square error of predictions on test and learning data

Considering the difference between predictions and actual time to fail as a controlled statistical process, we can compute a process capability index that represents the capability of the models to make predictions that are within given limits. Figure 5 shows the Cpk of the predictions considering a lower limit of 10 below the actual time to fail and an upper limit of 10 above the time to fail. Only Random Forest on the learning data keeps well within the limit and reaches a significant Cpk value. Considering the prediction as a statistical process there is a high chance that it predicts with an accuracy within the limit. Linear Regression scores best as it has a similar performance over the entire filament failure cycle.

Figure 6 evaluates that by splitting up the cycles in four parts. The important predictions are not those near the start of the cycle when the next maintenance is far away but instead the predictions that are made shortly before fail. The Bayesian Networks without smoothing after reconstruction and the Random Forest algorithm make far more accurate predictions near the end of the cycle than at the start.

The reason for this behavior is that Random Forests and Bayesian Networks are classification methods, while linear regression models allow extrapolation and therefore use all learning data to extrapolate the outcome near the breakdown. Classification methods rely much more on the learned failure conditions. Due to the fact that the behavior near the breakdown seems to change physically (the root cause for this is yet undiscovered), both classification methods show better results near the breakdown (Figure 6).

Another reason for linear regression showing relatively good performance in this case study is the strong linear relationship between the remaining filament lifetime hours and the most important predictor FIL-I.

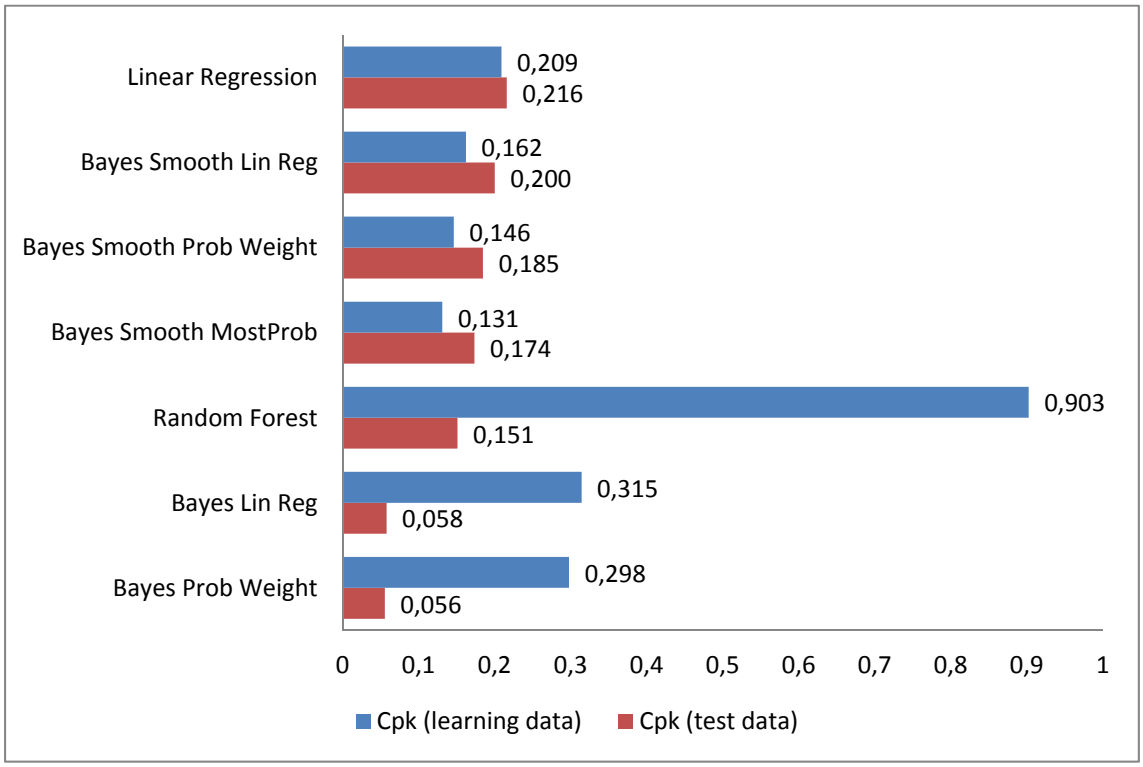

Figure 5: Cpk value (limits: -10/+10) of the predictions on test and learning data

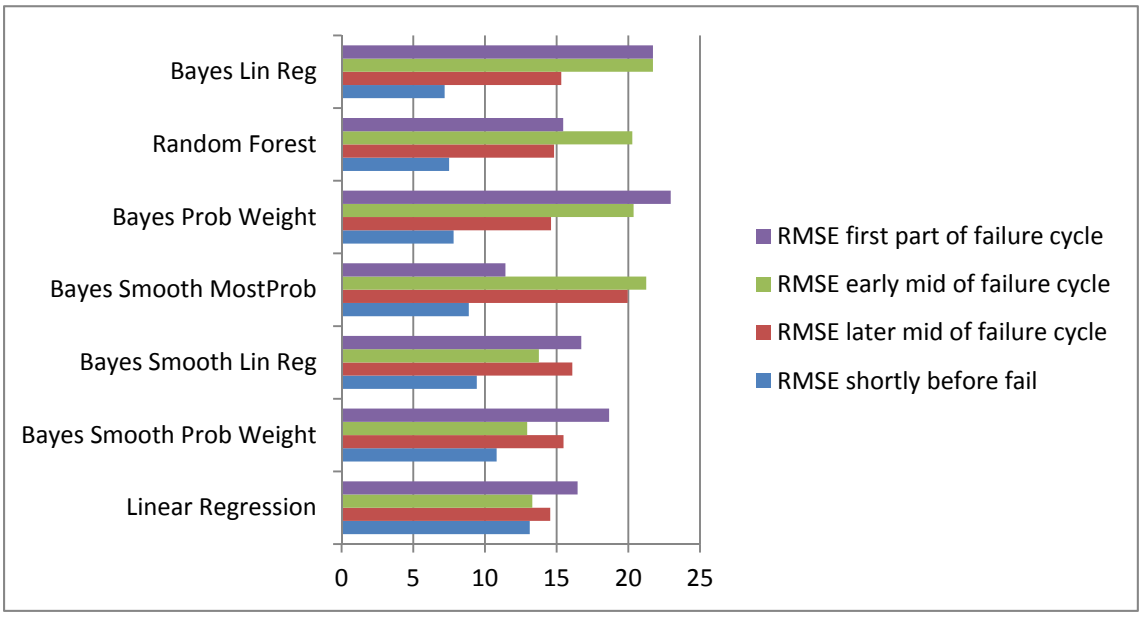

Figure 6: RMSE of the predictions in the different parts of the filament failure cycle

Figure 7 and 8 show the effect of noise on the performance of the models. Note that there is only one run per noise level so a certain amount of variance in the prediction quality is to be expected. The Bayesian Network models with smoothing tend to perform better at higher noise levels as the probabilistic approach is not as susceptible to noise. Through soft discretization during model learning, small deviations are also considered already during model learning, which makes their better performance under noisy conditions reasonable.

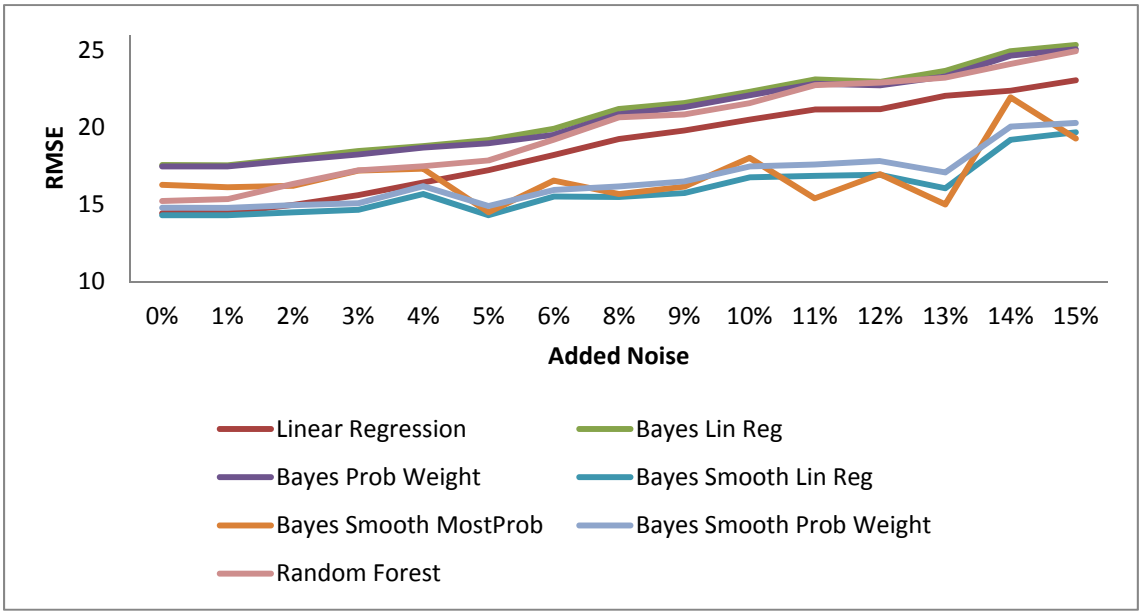

Figure 7: RMSE of the prediction depending on added input noise

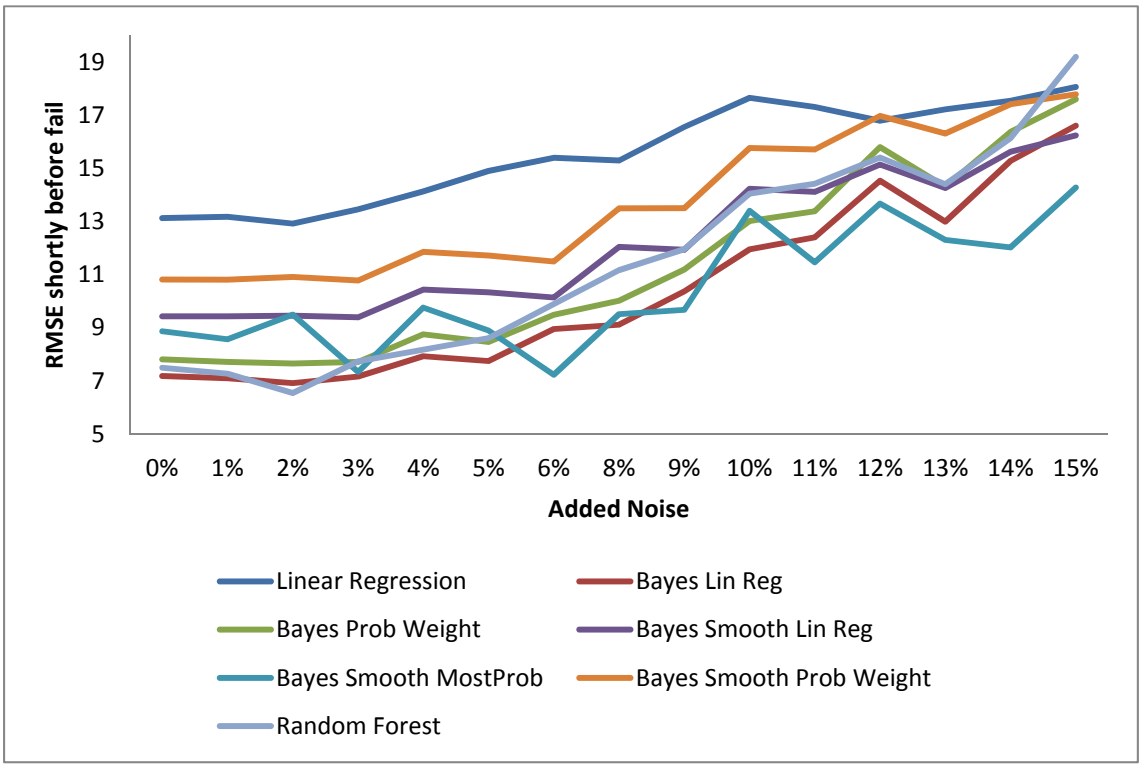

Figure 8: RMSE shortly before fail depending on added input noise

5 SUMMARY AND CONCLUSION

This paper demonstrates the viability of the Virtual Equipment for comparing different PdM models before their integration in the fab. The automated studies evaluate both the accuracy of the predictions in different parts of the failure cycle as well as their sensitivity to noise.

Main results of the application of the Virtual Equipment on three different PdM model types are:

- while Random Forest fits very well on the learning data, its performance on the test data is average
- the Linear Regression reconstruction variant of Bayesian Network PdM showed the best performance shortly before filament failure on the test data
- relying on the most probable state for the Bayesian Network PdM reconstruction is resistant to noise
- Linear Regression works well seen over the whole failure cycle but it is not that good near the end of the cycle

Future work will include:

- Interfacing of more prediction algorithms to the Virtual Equipment
- Extension of the automated study possibilities, e.g. repeatability studies (adding different noise of the same amplitude or assembling the learning/test data in different ways and evaluating the variance of the predictions)
- Application of the Virtual Equipment to further test cases
- Evaluation of the Virtual Equipment on other control applications

ACKNOWLEDGMENTS

This work was done within the IMPROVE project (ENIAC ID 12005). Funding by the German „Bundesministerium für Bildung und Forschung“, the „Österreichische Forschungsförderungsgesellschaft mbH“ and the EU is gratefully acknowledged.

REFERENCES

Berk R.A. 2008. *Statistical Learning from a Regression Perspective*. Springer, New York.

Breiman L., J.H. Friedman, R.A. Olshen, C.J. Stone. 1984. *Classification and Regression Trees*. Wadsworth, California.

Breiman, L. 1996. “Bagging predictors”. *Machine Learning* 24:123-140.

Breiman L. 2001. “Random forests”. *Machine Learning* 45(1):5-32.

Ebert-Uphoff, I. 2009. “A Probability-Based Approach to Soft Discretization for Bayesian Networks.” Research Report GT-ME-2009-002.

Efron B. 1979. “Bootstrap methods: another look at the jackknife.” *The Annals of Statistics* 7(1):1-26.

Hothorn T., K. Hornik, A. Zeileis. 2006. “Unbiased recursive partitioning: a conditional inference framework.” *Journal of Computational and Graphical Statistics*, 15(3), 651-674.

Liaw A., M. Wiener. 2002. “Classification and Regression by randomForest.” Technical report, R News.

Liu, Y..2008. “Predictive modeling for intelligent maintenance in complex semiconductor manufacturing processes.” PhD thesis, University of Michigan,USA.

Mattes A., M. Koitzsch, D. Lewke, M. Müller-Zell, M. Schellenberger. 2011. “A virtual equipment as a test bench for evaluating virtual metrology algorithms.” In *Proceedings of the 2011 Winter Simulation Conference*, Edited by Sain S., Creasey R.R., Himmelspach J., White, K.P., Fu M.: 1888-1897. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Murphy K.P. 2001. “The Bayes Net Toolbox for Matlab.” In *Computing Science and Statistics*.

Pearl J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (2nd Ed.)*. Morgan Kaufmann.

R Development Core Team. 2011. “R: A language and environment for statistical computing. R Foundation for Statistical Computing”. Vienna, Austria. ISBN 3-900051-07-0. http://www.R-project.org/.

Ryssel H., I. Ruge. 1978. *Ionenimplantation*. Teubner, Stuttgart.

Schirru H. A., S. Pampuri, G. De Nicolao. 2010. “Particle filtering of hidden gamma processes for robust Predictive Maintenance in Semiconductor Manufacturing.” *6th IEEE Conference on Automation Science and Engineering*. 57-62, Toronto.

Scheibelhofer P. 2011. “Tree-based methods for predictive failure detection in semiconductor Fabrication.” Master thesis, Institute of Statistics, Graz University of Technology.

G.A. Susto, A. Beghi, C. De Luca. 2011. “A Predictive Maintenance System for Silicon Epitaxial Deposition.” *7th IEEE Conference on Automation Science and Engineering*. Trieste.

Vachtsevanos G., F.L. Lewis, M. Roemer, A. Hess, B. Wu. 2006. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Wiley, New Jersey.

Wolf S. 2003. *Microchip Manufacturing*. Lattice Press.

AUTHOR BIOGRAPHIES

ANDREAS MATTES is a research scientist at Fraunhofer IISB. He holds a M.S. (Dipl.-Inf.) in computer science from the University of Stuttgart and joined Fraunhofer IISB in 2007. Andreas Mattes is working on the topic of equipment and process simulations for semiconductor manufacturing equipment. His email address is andreas.mattes@iisb.fraunhofer.de.

ULRICH SCHOEPKA received his M.S. (Dipl.-Ing.) degree in the field of electrical engineering at the Technische Universität München in 2006. He joined Fraunhofer in 2007. Since then, he gained experience in several assessment projects (SEA-NET, SEAL), mainly in the field of data analysis and statistical modeling and semiconductor equipment assessment. His email address is ulrich.schoepka@iisb.fraunhofer.de.

PETER SCHEIBELHOFER received the M.S. degree in mathematics from the University of Technology, Graz, Austria in 2011. Currently he is pursuing the Ph.D. degree in statistics in cooperation with ams. His current research interests are statistical data mining and multivariate statistical process control and its applications in semiconductor manufacturing. His email address is peter.scheibelhofer@ams.com.

GÜNTER LEDITZKY is project manager in the production at ams. He holds a PhD in technical physics from Graz University of Technology. Currently he is supervising the implementation of Fault Detection and Classification (FDC) and predictive maintenance in the 200mm wafer fabrication at ams. His email address is guenter.leditzky@ams.com.

MARTIN SCHELLENBERGER joined the department of Semiconductor Manufacturing Equipment and Methods at the Fraunhofer IISB in 1998. As Group Manager for Equipment and Advanced Process Control he is responsible for the development of solutions for quality control and their link to manufacturing systems. He is active in the fields of software architectures for the integration of metrology and sensors, interfaces and protocols for equipment communication in semiconductor manufacturing as well as cluster processing for new materials. He is leader of the European SEMI Process Control Systems task force. He received his diploma in Electrical Engineering and a Ph.D. in Electrical Engineering from the University of Erlangen-Nuremberg, Germany. His email address is martin.schellenberger@iisb.fraunhofer.de.