Virtual Metrology to Eliminate Test Wafers Measurements on Copper Electroplating Deposition*

Anastasiia Doinycko[1], Umberto Amato[2], Stanislau Raitsyn[3], Stefania Perna[6], Franco Blundo[6], Caterina Genua[6], Daniele Vinciguerra[6], Antonino La Magna[5], Andres Torres[4], Alex Rosenbaum[3], Massih-Reza Amini[1], and Patrizia Vasquez[6]

***Abstract*— In semiconductor manufacturing a virtual metrology (VM) task is a key solution for monitoring critical wafer parameters in-between scheduled measurement check-ups. Its goal is to provide an estimation of metrology values using process state information. This paper presents industrial case study of multiple machine learning (ML) strategies for VM during copper electrochemical deposition (Cu ECD).**

I. INTRODUCTION

There is a major effort in semiconductor manufacturing to develop a methodology that provides necessary in-line control for each produced integrated circuit (IC), also called chip. Fabrication of IC includes a sequence of high complexity chemical processes where the product is gradually created on a wafer made of silicon. The most effective control of the fabrication cycle implies a per-process check of wafers quality indicators, called metrology - physical characteristics like thickness, stress, and so on. Those 2D and 3D features are measured down to a nanometric scale with angstrom levels of precision and accuracy. In practice, collecting metrology for each wafer causes a high cost of production and significantly increases the fabrication cycle time. Therefore, a common method of monitoring consists of using only a few periodically sampled wafers. However, this approach may cause undetected defective items that are produced in-between scheduled measurements.

*The work presented in this paper has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826589, the MADEin4 project. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Netherlands, Belgium, Germany, France, Italy, Austria, Hungary, Romania, Sweden and Israel.

[1]A. Doinycko and M.-R. Amini is with Université Grenoble Alpes, Laboratoire d'Informatique de Grenoble, St Martin d'Hères, 38401 France (e-mail: anastasiia.doinychko@univ-grenoble-alpes.fr, massih-reza.amini@univ-grenoble-alpes.fr)

[2]U. Amato is with the Istituto di Scienze Applicate e Sistemi Intelligenti of the Italian National Research Council, Napoli, 80131 Italy (e-mail: umberto.amato@cnr.it)

[3]S. Raitsyn and A. Rosenbaum are with NVIDIA, Raanana, 4366241 Israel (e-mail: alexr@nvidia.com, stanislau@nvidia.com)

[4]A. Torres is with Siemens, Wilsonville, 97070 United States (e-mail: andres_torres@mentor.com)

[5]A. La Magna is with the Istituto per la Microelettronica e Microsistemi of the Italian National Research Council, Catania, 95121 Italy (e-mail: antonino.lamagna@imm.cnr.it)

[6]S. Perna, F. Blundo, C. Genua, D. Vinciguerra, and P. Vasquez are with STMicroelectronics, Catania, 95100 Italy (e-mail: daniele.vinciguerra@st.com, caterina.genua@st.com, stefania.perna@st.com, franco.blundo@st.com, patrizia.vasquez@st.com)

This problem can be resolved by assuming that metrology values for an individual process can be predicted from a numerical description of the process environment. In contrast to metrology, process parameters for each wafer can be efficiently collected from equipment sensors. Accordingly, Virtual Metrology (VM) objective is to develop robust methods for metrology output in the function of the process features. In state-of-the-art literature, linear and non-linear methods are proposed to solve VM tasks but for only specific tool conditions regarding different case studies such as chemical vapor deposition, factory-wide control, etch depth prediction, and more. However, every new process setting may require different or adapted to the case VM model. This paper proposes and compares several statistical and machine learning (ML) strategies, particularly to copper electroplating deposition (Cu ECD) process VM task.

This paper is organized as follows. A brief review of the relevant to the topic papers is given in Section II. Cu ECD data are presented in Section III. Then, Section IV is devoted to suggested methods for the raw data transformations into relevant input form for proposed different VM approaches, which are described in Section V. Evaluation strategy of corresponding methods and experimental results are discussed in Section VI. A summary of this work is given in Section VII.

II. RELATED WORKS

In recent years modern manufacturing industries hold a particular interest in automation technologies. It facilitates the creation of different scientific directions that are united by the Industry 4.0 concept, where VM is one of the highly demanding topics. Some studies state that the implementation of VM in a fab is estimated to increase the production volume output by nearly 10% [1] that justifies an appropriate research effort.

Mathematically VM is defined as a regression problem and a lot of ML regression strategies were adapted and proposed in order to effectively handle different process challenges, like drifts, inconsistent sampling, or preventive maintenance disturbances. Cu ECD VM is hardly described in the literature. Nevertheless, the most frequently used methods in different cases are the following: multiple linear regression (MLR) [5,6,8], support vector regression (SVR) [3], partial least squares (PLS) regression [2], neural networks [8,9,10]

and particularly radial basis function (RBF) neural networks [4,6], Gaussian process regression [7,8].

The purpose of this study is to investigate and discuss the effectiveness of different models, both common for VM tasks (like SVR) and new (like decision trees), applied to particular industrial data of Cu ECD.

III. COPPER ELECTROCHEMICAL DEPOSITION DATA

A Cu ECD electroplating machine generally has several process units called a "process chamber" where the process is performed. Each chamber can be considered a stand-alone machine that has local schedule of maintenance, recalibration, or metrology sampling. Production's wafers are grouped in lots; a lot can be formed of 25 wafers maximum of the same product type and every wafer in the lot is processed in parallel in all available chambers. The same set of process conditions, called recipe, run on each chamber, depending on the product type. Two sequential lots can correspond to different designs and so to different products, as consequence can have different process conditions.

Each chamber is equipped with a set of sensors that record numerical values of critical parameters as a function of time when process is running. If P is the total number of sensors, then Cu ECD is defined as a P-dimensional process

$$\mathbf{v}_t = (v_t^1, v_t^2, ..., v_t^P)^T,$$

where v_t^p is the value of a sensor parameter p at a time t. Then process conditions $\mathbf{x}^i$ of a single $i\text{-}th$ wafer in production line is a multivariate time sequence denoted in the following way:

$$\mathbf{x}^i = \{\mathbf{v}_{t_0^i}, \mathbf{v}_{t_1^i}, ..., \mathbf{v}_{t_{n(i)}^i}\},$$

where $[t_0^i, t_{n(i)}^i]$ is a specific time slot of a running chamber cycle when $i\text{-}th$ wafer of a lot was processed, and $n(i)$ is the total number of sampled points in this window. Generally, the window length $t_n^i - t_0^i$, or $i\text{-}th$ wafer's process duration, depends on recipe, therefore n_i is different for wafers of different product types. Figure 1 shows an example of consecutive time sequences on one of the chambers of a parameter for three wafers, also showing that the length of parameter sequences varies.

The recording system collects sensors' historical data with a fluctuating time step that usually may vary in a tolerance range of few seconds. Moreover, missing records may occur.As a result, specific methods for data preprocessing are required that are discussed further in Section IV.

Thickness – the measure of interest denoted by y in this paper – is a critical metrology characteristic that is monitored after Cu ECD process is finished. It is evaluated by averaging of its measurements at five locations on a wafer.

Current industrial case study includes wafers historical records of 10 different sensor parameters selected by fab experts. Due to the effort and cost, only a few wafers were selected for measuring metrology (two wafers out of the lot on average), and for the rest thickness is supposed to be

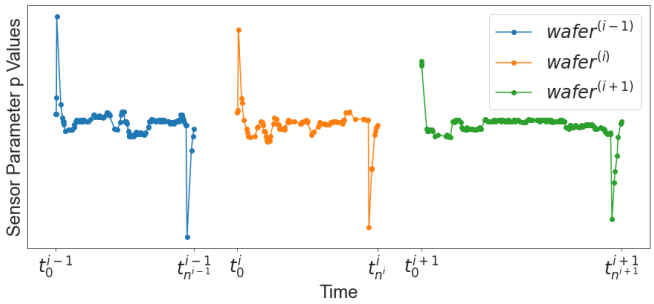

Fig. 1. Example of time sequences of a Cu ECD process parameter.

evaluated by the VM model. Accordingly, the given raw set is the following:

$$\mathcal{S} = \mathcal{S}_l \cup \mathcal{S}_u,$$

where $\mathcal{S}_l = \{(\mathbf{x}^i, y^i), i \in \{1, \ldots, N_l\}\}$ denotes a labeled subset of wafers' process parameters and $\mathcal{S}_u = \{(\mathbf{x}^i), i \in \{1, \ldots, N_u\}\}$ denotes an unlabeled subset of wafers that were not selected for metrology check (i.e. $N_l \ll N_u$). This paper is focused only on supervised learning, which involves development of methods that relies only on $\mathcal{S}_l$ subset to train proposed VM methods. However, semi-supervised learning, that combines labeled and unlabeled data during training, is considered as a promising direction for further study of this problem.

IV. DATA PREPROCESSING

To assure the quality and effectiveness of the VM application it is necessary to perform preliminary process data transformation into a suitable form of input variables for predictive models. The core of the data set is given by the time sequences of 10 different parameters that consist of hundreds of recorded values. This means that one set of process sequences for one wafer contains thousands of features in the average. Accordingly, it is a problem where features space dimension is of the same order as the number of samples. Therefore, measures must be taken to face the high dimensionality of the problem, both for accuracy of the solutions and to avoid oversmoothing. Another challenge, as mentioned above, is that the time sequences are not equispaced in time. A paths that is pursued in this work to solve the problems of different size of the sequences and of high dimensionality relies on feature extraction methods.

A. Time Series Feature Extraction

One of the most popular methods for reducing the number of variates is to deal with features extracted from the time series instead of original ones. In this work, the following set of basic statistical and descriptive features is defined to be computed for every single $i\text{-}th$ wafer's parameter p sequence:

- *AverageValue*: the average of the parameter's values along the entire time sequence
- *AverageLeft*: the average of the parameter's values along the first half of the time sequence

- *AverageMiddle*: the average of the parameter's values along the central 50% part of the time sequence (discarding the 25% left and right parts)
- *AverageRight*: the average of the parameter's values along the second half of the time sequence
- *AverageDelta*: the difference *AverageRight – AverageLeft*
- *FirstValue*: the first recorded value in the parameter's time sequence
- *LastValue*: the last recorded value in the parameter's time sequence
- *MedianValue*: the median of the parameter's values along the entire time sequence
- *SD*: the standard deviation of the parameter values along the entire time sequence
- *SDLeft*: the standard deviation of the parameter values along the first half of the time sequence
- *SDMiddle*: the standard deviation of the parameter values along the central 50% part of the time sequence (discarding the 25% left and right parts)
- *SDRight*: the standard deviation of the parameter values along the second half of the time sequence
- *ValueMin*: the minimum of the parameter values along the entire time sequence
- *ValueMax*: the maximum of the parameter values along the entire time sequence
- *Time1*: the first available time of the parameter's time sequence
- *Time2*: the last available time of the parameter's time sequence
- *Duration*: the difference *Time2 – Time1*
- *Area*: the total area under the curve of the parameter's time sequence

Such features are intended to catch the main characteristics of the signal, also attempting to get basic local information on the sequence.

Some of the features can measure similar quantities, therefore can show high correlations. Then, there is a high chance that performance of a predictive model can be impacted by a problem called multicollinearity. How to proceed with correlated features is a matter of choice and depends on many factors, starting from the regression framework. LASSO-type methods or boosted trees algorithms, for example, are immune to multicollinearity by nature, while linear regression can be possibly numerically unstable due to highly correlated predictors.

In all cases removing highly correlated variables reduces the features set size, therefore saving computational time for running predictive models and possibly giving more accurate solutions. In this work, a threshold of the maximum admissible correlation to 0.9 (in absolute value) was set in order to identify strong positive or negative relationships between features and discard corresponding ones.

B. Categorical Process Features

Every wafer corresponds to the specific product, or design in other words, that implies specific process settings – recipes. Generally, it is not necessary that every new product needs a new recipe. Indeed it is usual that different products share similar recipe, but never in reverse. Accordingly, both *Recipe* and *Product* are considered to be included in the feature set as categorical ones. Nevertheless, *Recipe* shows high correlation with some already introduced numerical process features, like *Duration*, therefore in many models Recipe is excluded.

Moreover, the data is collected from several chambers that are independent and each can bring its own bias due to different reasons, like different active time since the last recalibration, and so on. Therefore, *Chamber* is considered as another categorical feature.

This case study operates with around hundred product categories and 6 different chambers that are denoted in the dataset by unique names. Then, categorical feature embedding methods are used to transform the names into numerical labels, for example, One-Hot-Encoding approach or encoding with a method that associate each category with a unique discrete number (randomly or using some defined process, for example, bigger numbers are assigned to more populated categories).

V. METHODOLOGY

VM task is defined as a regression problem that aims to evaluate the thickness numerical values for processed wafers by a function of their process condition features. This work pursued different prediction methods based on process data representation described in the previous section. The models are described in the following subsections.

A. Gradient Boosted Decision Trees

Gradient Boosted Decision Trees (GBDT) is an efficient predictive approach from the family of ensemble learning algorithm [14]. As all methods in this family, GBDT relies in making predictions on a combination of decisions made by many "weak" learners, particularly decision trees. Specifically, all the "weak" models in GBDT are trained sequentially one at a time. When a new decision tree f_{k+1} is added, the k existing "weak" learners are fixed and left unchanged:

$$f(\mathbf{x}^i) = f_1(\mathbf{x}^i) + f_2(\mathbf{x}^i) + ... + f_k(\mathbf{x}^i),$$

while the new one is trained to reduce the error of the updated ensemble, therefore trained on the following set:

$$\mathcal{S}_{k+1} = \{(\mathbf{x}^i, y^i - \sum_{j=1}^{k} f_k(\mathbf{x}^i))\}.$$

Accordingly, boosting is an optimization approach that aims to minimize a loss of the model by adding weak learners using a gradient descent like procedure. Trees, in turn, are constructed in a greedy manner, choosing the best split points based on purity scores like Gini or to minimize the loss. In this case study GBDT is applied to predict the thickness from the all features extracted from parameters' time series. As it is already mentioned above, one of the advantages of

this method is that it can manage highly correlated features. Moreover, it also can manage missing values, which means that observations that has one or several parameters entirely missing are kept for training.

B. Categorical Boosted Regressor

As mentioned above, the gradient boosting is a powerful machine-learning technique that achieves state-of-the-art results in a variety of practical tasks with heterogeneous features, noisy data, and complex dependencies. Decision trees well fit numerical features, but work worse with categorical features, which are also important for prediction.

Categorical Boosted Regressor (CBR) is a new gradient boosting algorithm that successfully handles categorical features and takes advantage of dealing with them during training as opposed to preprocessing time. Another advantage of the algorithm is that it uses a new schema for calculating leaf values when selecting the tree structure, which helps to reduce overfitting [11].

CBR is applied to predict the thickness from the extracted features that consist of both numerical and categorical types and to prevent overfitting.

C. Random Forest

Another widely used ensemble learning approach is the Random Forest (RF) [15]. It combines multitude of decision trees at learning time. Each tree is trained independently in parallel on the randomly subsampled small set of the data. This type of ensemble learning is called bootstrap and it allows to improve the variance and provide unbiased estimate of the generalization error. Generally, RF is considered as highly accurate estimator in many tasks, it can successfully handle high-dimension feature input. However, it has risk to overfit for especially noisy data sets and in case categorical data with many different levels is included RF has "absent levels" problem in some random subsets.

RF in this work is applied to estimate thickness from both numerical and categorical extracted features. The numerical highly correlated features are effectively processed with RF. However, for more efficient use of categorical variables CBR method is considered.

D. Support Vector Regression

Support Vector Regression (SVR) [16], unlike most linear regression models, operates to minimize the L_2 norm of the coefficient vector under additional constrain for error margin that allows to define an "acceptable" level of error to fit the data. Moreover, the penalization term for the values outside of ε is introduced, which is controlled by hyperparameter C. As a result, the objective function is the following:

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i^N |\xi^i|.$$

under constrain : $|y^i - \mathbf{w}^i\mathbf{x}^i| \leq \varepsilon + |\xi^i|$. Additionally, kernel function can be applied to transform the data to make it possible to fit with linear model. The kernel functions exist of different types, but in this work the Radial Basis Function (RBF) is used: $K(\mathbf{x}^i, \mathbf{x}^j) = \exp\left(-\|\mathbf{x}^i - \mathbf{x}^j\|^2/2\sigma^2\right)$.

1) Stepwise Regression: Linear and Multiple regression are the most consolidated tools for explaining a variable (regressor) from one or more numerical and/or categorical variables, respectively (predictors). They have excellent and well studied theoretical properties, besides being very intuitive in explaining a resulting model. Of course, such models are limited in all problems where a nonlinear relationship between the regressor and the predictors is expected. Furthermore, they can be used only in a context where the number of predictors is much less than the size of the available samples, otherwise, due to intrinsic ill-conditioning arguments, variance of the estimated coefficients grows up to make them meaningless. For problems as the one of the present paper the number of predictors can be as high as over 700. However one expects that the number of really significant predictors (i.e., explaining Thickness) is a small fraction of them. A way to face ill-conditioning is to select a small number of predictors. Stepwise regression is a multiple regression method where one predictor (i.e., Feature of a Parameter) at time is added until R^2 predicted on a test data does not reached a fixed criterion, e.g., when no further improvement of R^2 is observed. The estimate of coefficients of predictors and of R^2 have to be performed on separate data sets, a training and testing one, respectively, because otherwise R^2 always decreases with the number of Parameter/Features. Inclusion of further Parameter/Features as predictors occurs semi-exhaustively by a trial mechanism at different steps, where at a certain step all remaining variables (not already included in the model at previous steps) are tried and the one with the best R^2 is selected.

E. Penalization/Elastic net

In the same direction as Multiple Regression, Penalization relies on functional arguments to select variables. Rationale comes from the observation that in high dimensional setting the solution can be controlled introducing ancillary constraints under the form of a penalizing functional and a corresponding penalization coefficient. In mathematical terms the quadratic optimization problem that is at the basis of Regression is replaced by the following constrained problem:

$$\min_{\beta} \|Y - X\beta\|_2^2 + \lambda p(\beta),$$

with Y being the regressor (Thickness), X the design matrix of the predictors, $\beta = (\beta_i)_{i=1,\dots,n}$ the vector of n unknown coefficients, p a penalization function and λ the corresponding penalization coefficient.

Prototype of such regression methods is LASSO [12], where $p(\beta) = \|\beta\|_1$. This penalization term yields sparse solutions, in the sense that some (possibly most) coefficients are set to 0, and therefore corresponding predictors do not enter the model and are not selected.

Some generalizations of LASSO have been proposed aimed at fixing the bias inherent in LASSO and at improving accuracy, through different choices of the penalization function and even replacing the L_2 norm of the LS term with different loss functions. We defer to [13] for a comprehensive

treatment. In the present work we shall also considered LASSO, MCP and SCAD penalty functions.

In all methods we also included a second penalization term as the L_2 norm with its penalization coefficient (elastic net):

$$\min_{\beta} \|Y - X\beta\|_2^2 + \lambda p(\beta) + \alpha\|\beta\|_2^2.$$

This fixes drawbacks of penalization methods with high number of predictors and highly correlated predictors. Both penalization coefficients λ and α were estimated by Cross Validation.

Finally we stress that in order to increase accuracy, penalization was resorted only to select predictors, their estimate being obtained by a multiple regression with the selected predictors.

VI. RESULTS

This section shows comparison of the methods considered in the paper.

A. Evaluation Level

Comparison of different VM methods in this work is accomplished by estimating the coefficient of determination R^2. It is a statistical measure in the range [0,1], which shows a percentage of the dependent variable variation that a linear model explains:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_{real}^i - y_{predicted}^i)^2}{\sum_{i=1}^{N}(y_{real}^i - \frac{1}{N}\sum_{i=1}^{N} y_{real}^i)^2}.$$

Higher R^2 values represent smaller differences between the observed data and the fitted values.

It is mandatory that the error indicator is evaluated on data that are never seen from the model while training itself and choosing hyperparameters, otherwise a heavy bias would be generated that would substantially decrease the error indicator. This phenomenon is called "overfitting". For some methods it could lead to a perfect reconstruction of the test data, but running the final model on truly new data would possibly result in large errors. This effect is more evident when the number of predictors is not a fraction of the numerosity of the samples or for those methods that involve a high number of parameters to estimate (e.g., Neural Networks, Random Forest based).

Accordingly, the data for this case study are divided into two subsets: Development data and Running data. The first subset is used for parameters analysis, preprocessing strategy and feature selection, training and validating the VM models; while the second subset is used to test the performance of the VM models. The separation into the two parts is performed by time split, meaning that Running data are the most recent observations of the historical data. In this study the Development data are formed of observations collected for about a year and the Running data consists of observations collected during the next month after the Running data set.

One of the most consolidated methods for splitting a data set into a Training and Validation one is K-fold Cross Validation (CV), in particular a 10-fold CV (K=10). Essentially the entire Development data set is split into K=10 disjoint groups approximately of the same size. They generate 10 different sets of Validation and Training data; in each one the Validation set is given by one of the 10 groups with 10% of data and the Training set is made of the remaining 90% of data. In this way Training and Validation data are disjoint for each group. To estimate an error indicator, the model is run 10 times, each time on a couple Training-Validation, where the Training data set is used to train the model (and hyperparameters, eventually), and error is estimated on the Running data (test set). The final estimate of the error indicator is obtained averaging the error indicators of the 10 runs.

Despite of its simplicity, random generation of K-fold groups needs a deeper understanding in some circumstances. A plain random selection can give troubles on the representativeness of other variables both in the Training and Validation groups. This is exactly the case of the process data set, namely with the presence of the different recipes or product types. It can happen that in a Validation data set of a group there exist samples belonging to a certain recipe or product that do not exist in the corresponding Training data set. In this case estimate for such a recipe or product on the Validation data set is not possible, and the resulting error indicator is biased. Another problem arises with products or recipes that are less populated than K samples. In this case it is sure that for some CV groups the corresponding recipe or product will be missing, again producing a bias in the error indicator. To overcome these problems, the following actions were made: randomly select CV groups stratifying by recipe or product; in the case of less populated recipes (or products; less than K), artificially increase the sample up to K by random selection of K samples with repetitions. As a result, some bias is still introduced in the error indicator, however experiments show that it is much better controlled.

B. Development Level Results

First, a comparison of the evaluated R2-scores during the development level for all of the introduced methods is considered and summarized in Table I.

TABLE I
R^2-SCORE AND ITS STANDARD DEVIATION FOR THE METHODS CONSIDERED BASED ON A 10-FOLD CV SETTING

VM model	**Input**	**R^2-score**
GBDT	Full set of extracted features	0.886±0.012
CBR	Subset [*FirstValue*, *LastValue*, *ValueMin*, *ValueMax*, *SD*, *AverageValue*] + categorical features	0.87±0.011
Stepwise regression	200 Uncorrelated features	0.683±0.001
Penalization (MCP penalty)	200 Uncorrelated features	0.714±0.040
RF	Full set of extracted features	0.855±0.016
Neural networks	200 Uncorrelated features	0.671±0.081
SVR	200 Uncorrelated features	0.699±0.017

Evaluation of each model follows the same scenario described above, however, the input set is different and the choice of it explained by predictive algorithm type. Trees based methods, like GBDT or RF, receive all extracted features, since they effectively can discard unimportant ones without loss in performance. On the other hand, sensitive to highly correlated features methods, like Stepwise Regression or Penalization, requires cleaning with the 0.9 threshold of correlation. Moreover, GBDT as far as the method can handle missing data, includes a bigger set of observations for training (20% of observations that have around 40% of missing feature values). It is straightforward to observe that ensemble methods show the best prediction accuracy in case when the observations for error evaluation belongs to the familiar feature distribution and classes during training. Moreover, it can be noticed that decision trees methods that by nature have their own mechanism for feature selection shows generally better results.

C. Running Level Results

Next, the error of predictions on the Running level is investigated. Generally, the process equipment has changing behavior over time and undergoes maintenance events, which cause abrupt recalibration (equipment reset). Additionally, new products can appear in production that may require new process conditions. All above certainly affects the accuracy of the developed VM model in time. Therefore, the Running level is an experiment to show how the proposed methods are affected by different changes. Thereby Table II shows R2-score rate on different horizons in the future test set. Evidently, all methods generally decrease at a quite fast pace, nevertheless, GBDT and Stepwise Regression methods have the best generalization on new data.

TABLE II

R^2-SCORE ESTIMATED ON DIFFERENT TIME SPANS OF THE RUNNING TEST SET WITH SEVERAL MODELS TRAINED ON THE DEVELOPMENT DATA SET

VM model	1 day	1 week	2 weeks	3 weeks	4 weeks
GBDT	0.71	0.79	0.73	0.71	0.57
CBR	0.45	0.42	0.52	0.47	0.44
Stepwise regression	0.87	0.65	0.57	0.57	0.53
Penalization (MCP penalty)	0.69	0.70	0.58	0.53	0.53
RF	0.69	0.73	0.68	0.62	0.49
Neural networks	0.86	0.59	0.53	0.40	0.27
SVR	0.30	0.31	0.33	0.38	0.33

VII. CONCLUSIONS

This paper provides an overview of different ML algorithms for VM task in an application with Cu ECD process data. It proposes a method for feature extraction that allows reducing the size of the feature vector required for analysis but still keeping similarity in representation with the original data set. It suggests adapted to particular problem GBDT and Stepwise Regression models and shows their better performance and generalization compared with widely used in literature SVR and Neural Networks. However, the reliability of the proposed methods is shown to decrease with time, which gives a perspective to work on development of a maintenance algorithm in future.

REFERENCES

[1] F. Cheng, J. Y. Chang, H. Huang, C. Kao, Y. Chen and J. Peng, "Benefit Model of Virtual Metrology and Integrating AVM Into MES," in IEEE Transactions on Semiconductor Manufacturing, vol. 24, no. 2, pp. 261–272, May 2011

[2] S. Lynn, J. V. Ringwood and N. MacGearailt, "Weighted windowed PLS models for virtual metrology of an industrial plasma etch process," 2010 IEEE International Conference on Industrial Technology, Via del Mar, Chile, 2010, pp. 309–314

[3] H. Purwins et al., "Regression Methods for Virtual Metrology of Layer Thickness in Chemical Vapor Deposition," in IEEE/ASME Transactions on Mechatronics, vol. 19, no. 1, pp. 1–8, Feb. 2014

[4] M. Hung, T. Lin, F. Cheng and R. Lin, "A Novel Virtual Metrology Scheme for Predicting CVD Thickness in Semiconductor Manufacturing," in IEEE/ASME Transactions on Mechatronics, vol. 12, no. 3, pp. 308–316, June 2007

[5] A. A. Khan, J. R. Moyne and D. M. Tilbury, "An Approach for Factory-Wide Control Utilizing Virtual Metrology," in IEEE Transactions on Semiconductor Manufacturing, vol. 20, no. 4, pp. 364–375, Nov. 2007

[6] A. Ferreira, A. Roussy and L. Conde, "Virtual Metrology models for predicting physical measurement in semiconductor manufacturing," 2009 IEEE/SEMI Advanced Semiconductor Manufacturing Conference, Berlin, Germany, 2009, pp. 149–154

[7] S. Lynn, J. Ringwood and N. MacGearailt, "Gaussian process regression for virtual metrology of plasma etch," IET Irish Signals and Systems Conference (ISSC 2010), Cork, 2010, pp. 42–47

[8] J. Wan, S. Pampuri, P. G. O'Hara, A. B. Johnston and S. McLoone, "On regression methods for virtual metrology in semiconductor manufacturing," 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), Limerick, 2014, pp. 380–385

[9] Y.-J. Chang, Y. Kang, C.-L. Hsu, C.-T. Chang and T. Y. Chan, "Virtual Metrology Technique for Semiconductor Manufacturing," The 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 2006, pp. 5289–5293

[10] F. Cheng, H. Huang and C. Kao, "Dual-Phase Virtual Metrology Scheme," in IEEE Transactions on Semiconductor Manufacturing, vol. 20, no. 4, pp. 566–571, Nov. 2007

[11] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush and A. Gulin, "CatBoost: unbiased boosting with categorical features", NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, Canada, 2018, pp. 6639–6649

[12] R. Tibshirani, "Regression shrinkage and selection via the Lasso," in J. Royal Statistical Society Ser B, vol. 58, no. 1, pp. 267–288, 1996

[13] U. Amato, A. Antoniadis, I. De Feis and I. Gijbels, "Penalised robust estimators for sparse and high-dimensional linear models," Statistical Methods & Applications, vol. 30, pp. 1–48, 2021

[14] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: a highly efficient gradient boosting decision tree," In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 3149–3157

[15] L. Breiman, "Random Forests," Mach. Learn. 45, 1 (October 1 2001), 5–32

[16] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines" In Proceedings of the 9th International Conference on Neural Information Processing Systems (NIPS'96), MIT Press, Cambridge, MA, USA, 155–161