Towards Fab Cycle Time Reduction by Machine Learning based Overlay Metrology

Faegheh Hasibi*[a], Leon van Dijk[a], Maialen Larrañaga[a], Anne Pastol[a], Auguste Lam[b], and Richard van Haren[a]

[a]ASML Netherlands B.V., De Run 6501, Veldhoven 5504 DR, Netherlands
[b]STMicroelectronics Crolles, 850 rue Jean Monnet, F-38926 Crolles Cedex, France

ABSTRACT

Overlay is a one of the most critical design specifications in semiconductor device manufacturing. Any state-of-the-art production facility has overlay metrology in place to monitor overlay performance during manufacturing and to use the measurements for overlay control. Especially since the introduction of multi-patterning, with its tight overlay requirements and increased number of process steps, there has been an increased need for additional metrology. Overlay metrology brings cost-added value to semiconductor device manufacturing and it should be reduced to a minimum to keep costs at acceptable levels, which can be a challenge in the multi-patterning era. Replacing some real overlay measurements with predicted values, referred to as *virtual overlay metrology*, could be a viable solution to address this challenge.

In this work, we develop virtual overlay metrology and aim at predicting the overlay for a series of implant layers. To this end, we apply machine learning algorithms, and neural networks in particular, to build a complex non-linear model directly from data. Our model takes a set of features that are designed based on the physical concepts of overlay and outputs the overlay map of a target layer. The features include overlay of another implant layer of the same wafer, exposure tool fingerprints, scanner logging, and process data. We evaluate our model using production data and we show the prediction performance for the raw overlay, as well as for the correctable and non-correctable overlay errors.

Keywords: Machine learning, overlay prediction, neural networks, overlay control, process optimization

1. INTRODUCTION

Overlay is one of the most critical design specifications in semiconductor device manufacturing and it plays a critical role in enabling the extreme miniaturization of integrated circuits (IC). Overlay is a measure for how accurate the layers that comprise an IC are positioned with respect to each other. Without accurate alignment between layers, electrical contacts between structures will be poor or there can be shorts. Maintaining good overlay performance during manufacturing is therefore essential to obtain high yield and to ensure that the performance and reliability of the eventual device are according to the specifications.

Any state-of-the-art semiconductor device production facility has nowadays overlay metrology systems in place to monitor the overlay performance during manufacturing. Overlay metrology can be used to track variations in overlay using a Statistical Process Control (SPC) system. The overlay measurements are also used by a so-called Advanced Process Control (APC) system in order to further minimize overlay variations by applying some kind of feedback control to the lithography systems and/or to other processing equipment.

A common practice in the industry is to employ a sampling scheme for overlay metrology in which only a subset of the wafers going through the production line are measured. However, for SPC, outlier detection, and even overlay (feedback) control, it could still be beneficial to have an accurate and reliable estimate of the overlay performance of the non-measured wafers. This calls for an increased number of overlay measurements, which at the same time involves extra costs and increases fab cycle time. In addition, the introduction of multi-patterning has challenged the industry not only with ever-tightening overlay requirements, but also with

*faegheh.hasibi@asml.com; www.asml.com

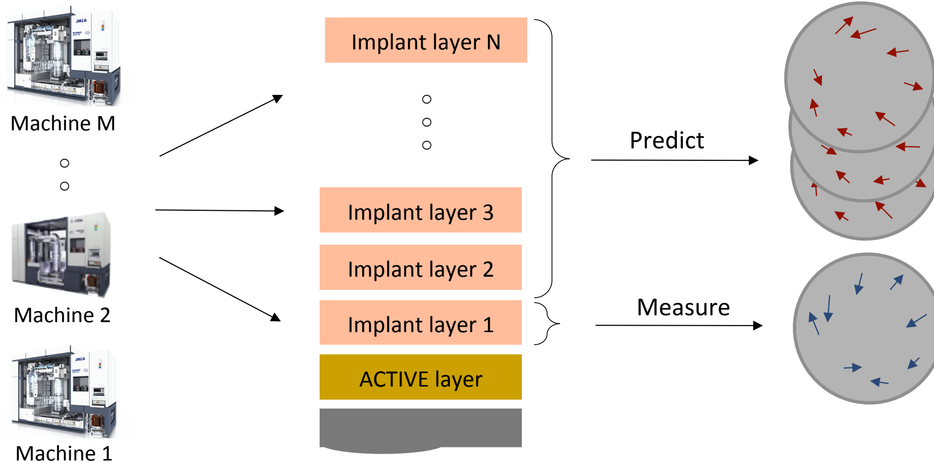


Figure 1. The overlay of a series of nine implant layers is predicted using a machine learning model. The layers are exposed using different scanners and their overlay is measured with respect to the same reference layer (Active layer). Our method takes overlay of one layer as an input and predicts the overlay of $N-1$ other layers, thereby reducing both fab cycle time and capital expenditure.

a significant increase in the number of processing steps. This goes hand-in-hand with an increased need for additional metrology. One solution to restrict the use of additional metrology is to predict some measurements from readily available fabrication parameters, metrology, and sensor data using mathematical and statistical models. This is often referred to as *virtual metrology*. Reliable and accurate virtual (overlay) metrology could therefore become essential to keep production costs, both capital expenditure and fab cycle times, at acceptable levels in the multi-patterning era.

The goal of this work is to develop virtual overlay metrology for a series of nine implant layers in production at STMicroelectronics in Crolles, France. For each implant layer, the overlay is typically measured on a subset of wafers. Since the non-lithography processing steps in between the implant layer exposures are not expected to contribute significantly to the overlay, the difference in overlay between the nine implant layers is mainly determined by the scanner baseline performance, matching between scanners, and eventual mask pattern placement errors. By establishing a model that can predict overlay errors, one can limit the overlay metrology to one implant layer and predict the overlay for the remaining eight layers; see Figure 1 for illustration. With overlay metrology reduced with a factor of nine (for implant layers), a significant reduction in fab cycle time could potentially be achieved.

We employ machine learning algorithms for building an overlay prediction model. Machine learning algorithms, and neural networks in particular, have changed the landscape of computational modeling over the past years and are shown to be effective solutions for the semiconductor industry.[1–4] In the following section, we discuss the problem at hand in more details. We provide an understanding of the underlying physical concepts of overlay, which are essential in setting up a machine learning framework with high prediction capability. In the next sections (§3 and §4), we will describe our overlay prediction methodology, the experimental setup, and the prediction performance of our model.

2. PROBLEM STATEMENT

Understanding the underlying physical concepts of the overlay prediction problem is essential for designing a machine learning based solution. Machine learning algorithms learn a model from sample data, represented by a set of features (input variables). In our case, these features are scanner logging, applied APC corrections, and context data (e.g., machine and reticle IDs). Learning a model with high predictive power is achievable only by knowing the concepts behind these features and linking them to the physical properties of the problem. For the problem at hand, the following aspects should be taken into consideration:

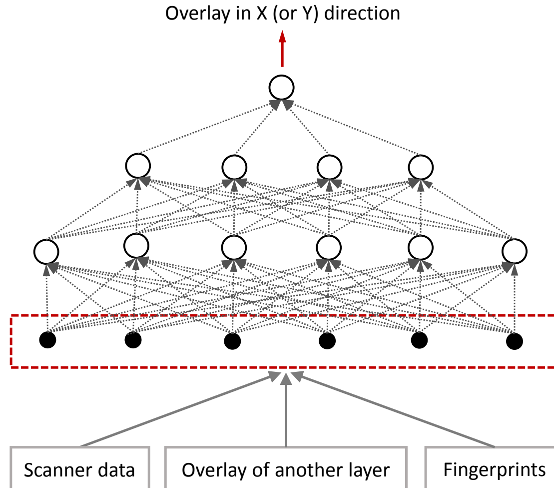


Figure 2. Architecture of the neural network model used in our overlay prediction methodology.

- **Implant layers**: Overlay prediction is performed on implant layers. Each implant layer is exposed by aligning back to the same set of alignment marks in the Active layer, which is also known as the Shallow-Trench-Isolation (STI) layer. Overlay is measured with respect to the Active layer as well. In this work, we will refer to this layer also as the *reference layer*. Wafer processing steps in between the implant layer exposures are not expected to cause alignment mark deformation, overlay target degradation, and wafer grid deformation. This means that the impact of non-lithography contributors on the overlay performance is minimal and the difference between the overlay of various implant layers is mainly determined by the scanner performance, matching of scanners, and eventual mask effects.

- **Same overlay target location**: Overlay is measured on a specific target, consisting of a pattern created in the Active layer and another pattern created in the overlaying photoresist of an implant layer. For each implant layer, overlay is measured with respect to the same pattern in the Active layer, and therefore overlay targets are at the same location for all implant layers. This implies that intra-field effects do not contribute to the difference in overlay between the various implant layers.

- **Multiple scanners**: The exposure fingerprint of a single lithography tool is ideally a perfect grid. However, in practice the scanner is not perfect and small systematic residual errors with respect to the grid remain. Since the stages of the scanner are extremely repeatable, these errors will cancel out in the overlay between subsequent layers that are exposed on the same scanner when operating in dedicated chuck overlay (DCO) mode. In case the subsequent layers are exposed on different chucks or on a different scanner, the machine and chuck specific exposure fingerprints will become visible in the overlay between layers. In the problem at hand different scanners have been used to expose the Active layer and the implant layers. All these machines, as well as both their chucks, have a specific exposure fingerprint, which need to be incorporated in our prediction model in order to achieve accurate virtual overlay metrology.

- **Short time line between implant exposures**: The machine and chuck specific exposure fingerprints may drift over time. For this reason, ASML has introduced the BaseLiner^{TM} products in order to maintain long-term machine overlay stability. The implant layers, however, are exposed and processed consecutively within a few days. Therefore, the amount of scanner drift is low and the scanner drift can be neglected in the overlay prediction model.

To summarize, the overlay of each implant layer is measured with respect to the Active layer on targets at exactly the same location. This means that the overlay of one particular implant layer can reveal information about the overlay of the other implant layers. For example, a certain process-induced distortion pattern in the Active layer of a particular wafer will also appear in the measured overlay of that same wafer for all implant

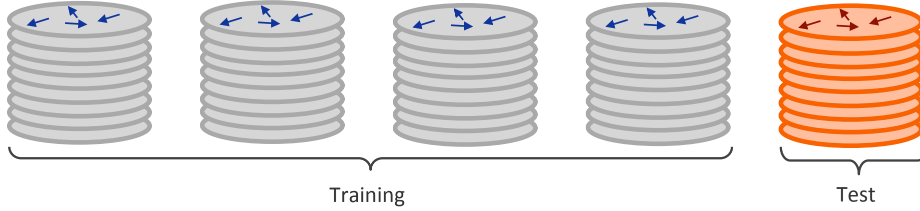


Figure 3. The overlay measurement points are randomly assigned to the training and test sets such that points belonging to the same lot are kept in either of the sets. This strategy of train-test splitting is used to perform 5-fold cross validation. Other methods of train-test splitting result in substantial information leakage from the training set.

layers. We will exploit this characteristic and use the measured overlay of one implant layer as input to the machine learning algorithm to predict the overlay for the other implant layers. This further enables prediction of a target layer overlay using the measured overlay of a layer, exposed before or after the target layer. In the next section, we will detail out the overlay prediction methodology and show how the above aspects are incorporated in the machine learning framework.

3. OVERLAY PREDICTION METHODOLOGY

We employ machine learning algorithms to build a model for overlay prediction. Machine learning algorithms are able to learn a data-driven model from sample inputs and make a prediction for unseen inputs. In our setting, we represent each layer with a set of measurement points and assign a set of features (predictors) to each point. These features provide signals about the overlay of a layer and are extracted from various sources. we discuss these features in more details in Section 3.1, followed by a description of our machine learning algorithm in Section 3.2.

3.1 Overlay predictors

In order to build an overlay prediction model, we develop a set of features for each measurement point of an exposed layer and use them in a supervised learning framework. The features are of three main categories: (i) overlay of another layer, (ii) machine-chuck fingerprints, and (iii) scanner and context data. Below we describe each of these features.

3.1.1 Overlay of another layer

When the overlay of several layers is measured with respect to the same reference layer, the overlay of one particular layer can reveal information about the overlay of other layers; especially information related to the reference layer. This is the core idea of our approach to perform through-stack overlay prediction: we utilize one implant layer of each wafer to predict overlay of other implant layers of the same wafer. For the sake of consistency, we choose to use the first measured layer of a wafer and we refer to it as *first layer overlay* (l_1). In order to achieve accurate overlay prediction using the first layer overlay, we need to account for different conditions that are used for exposing each wafer (e.g., machine, and chuck) and isolate the effect of each condition separately. We define the overlay of a particular wafer-layer (w, l_x) as a general function of:

$$OVL(w, l_x) = (\mathcal{M}^w_{l_x} - \mathcal{M}^w_{l_{ref}}) + (\mathcal{C}^w_{l_x} - \mathcal{C}^w_{l_{ref}}) + \mathcal{E}^w_{l_x}, \quad x \in \{1, 2, .., N\} \tag{1}$$

where the subscript l_{ref} refers to the reference layer, and $\mathcal{M}_{\{.\}}$ and $\mathcal{C}_{\{.\}}$ represent the fingerprints of the machine and chuck used for the exposure of a particular wafer and layer. Here, $\mathcal{E}^w_{l_x}$ denotes other contributors of overlay, such as the reticle and metrology tool. By subtracting the overlay of the first layer l_1 from a target layer l_x, we are left with:

$$OVL(w, l_x) = OVL(w, l_1) + (\mathcal{M}^w_{l_x} + \mathcal{C}^w_{l_x}) - (\mathcal{M}^w_{l_1} + \mathcal{C}^w_{l_1}) + \mathcal{E}^w_{l_x} - \mathcal{E}^w_{l_1} \tag{2}$$

$$\approx OVL(w, l_1) + (\mathcal{M}^w_{l_x} + \mathcal{C}^w_{l_x}) - (\mathcal{M}^w_{l_1} + \mathcal{C}^w_{l_1}). \tag{3}$$

Here, we assume $\mathcal{E}^w_{l_x}$ and $\mathcal{E}^w_{l_1}$ to be negligible, and estimate the overlay function as a combination of the first layer overlay and the difference between machine-chuck fingerprints of the two layers.

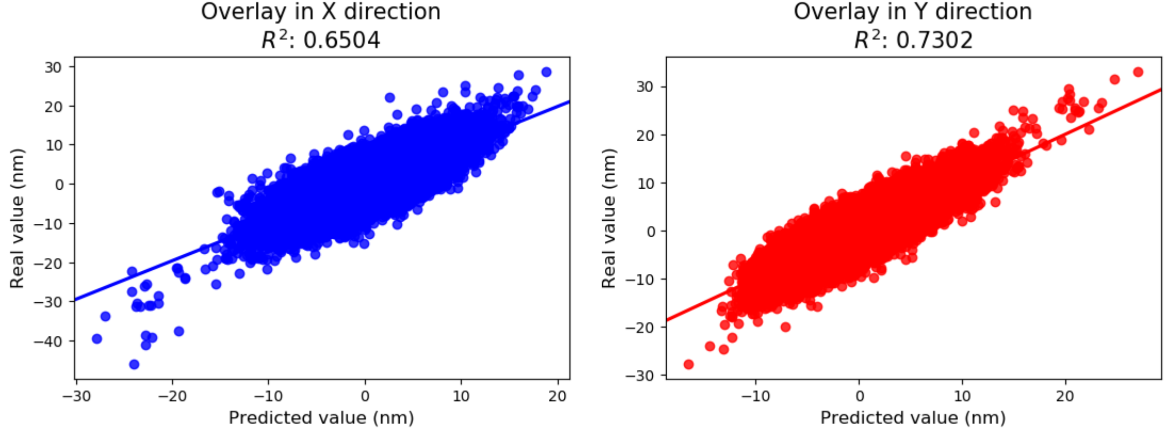


Figure 4. Measured versus predicted values of overlay per measurement point. The left and right scatter plot correspond to the overlay in X and Y directions, respectively, and the R^2 statistic is used as a measure of prediction performance.

3.1.2 Machine-Chuck fingerprints

It is well recognized that each exposure tool leaves certain fingerprints on overlay, and these fingerprints vary for each chuck of the machine.[5] To capture these fingerprints, we choose a layer l_k and take the average overlay of this layer under certain conditions. Broadly speaking, the fingerprint function $\mathcal{F}$ is parameterized by a layer l_k and computed for a condition c as:

$$\mathcal{F}(c; l_k) = \frac{1}{n_c} \sum_{i:C(w_i,l_k)=c} OVL(w_i, l_k), \tag{4}$$

where n_c is the number of wafer-layers (w, l_k) that are exposed under condition c and $C(w_i, l_k)$ is a function that returns the condition under which layer l_k of wafer w_i is exposed. For our experiments, we define the function $C(.)$ as:

$$C(w, l_x) = \langle (w, l_x).\text{machine}, (w, l_x).\text{chuck}, (w, l_{ref}).\text{machine}, (w, l_{ref}).\text{chuck} \rangle, \tag{5}$$

which returns the machine and chuck identifier of the wafer-layer (w, l_x), as well as its reference layer l_{ref}. We use these fingerprints as individual features and also for computing $(\mathcal{M}^w_{l_x} + \mathcal{C}^w_{l_x})$ and $(\mathcal{M}^w_{l_1} + \mathcal{C}^w_{l_1})$ terms in Eq. 3.

3.1.3 Scanner and context data

The other overlay predictors used in our model are taken from the exposure logging and the contextual information around it. We take into account this information, such as APC corrections, reticle information, wafer and stage alignment data.

3.2 Machine learning algorithm

We now detail the machine learning framework employed to predict the overlay map. In this framework, each measurement point is represented by its position in Cartesian coordinates $p = (x, y)$. The overlay map, consisting of m measurement points, is then presented as $\{(d_{x_1}, d_{y_1}), (d_{x_2}, d_{y_2}), ..., (d_{x_m}, d_{y_m})\}$, where d_{x_i} and d_{y_i} are overlay errors of point p_i in X and Y directions. Using a machine learning algorithm, we build a model that can predict the overlay error for a single measurement point in one of X or Y directions. Each instance in our model is a measurement point of an exposed layer, represented by a set of features explained in the previous section (§ 3.1).

We experimented with a variety of machine learning algorithms: (i) Lasso,[6] which is a linear regression algorithm, (ii) Random Forests,[7] which is a non-linear model based on ensemble of decision trees, and (iii) Feed-forward neural networks.[8] In the remainder of this paper, we focus on the neural networks model and its results, as it achieves the best performance compared to the other models.

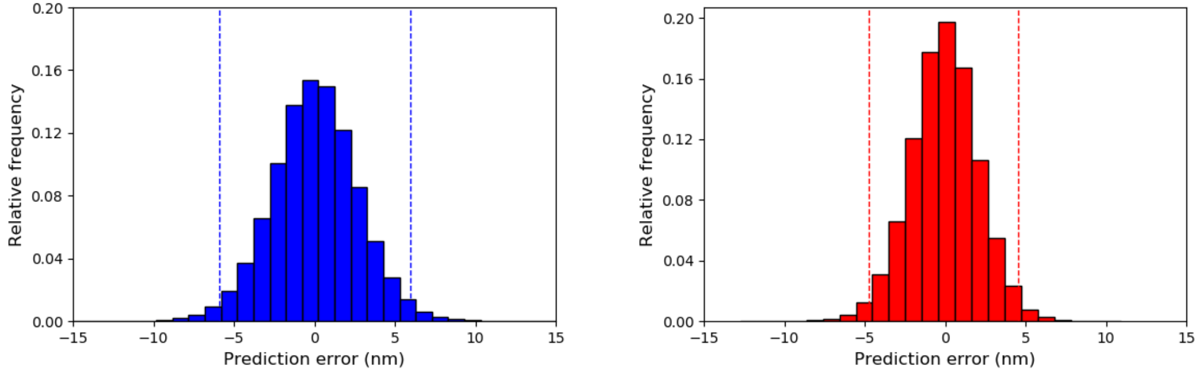


Figure 5. Histogram of the prediction error for overlay in X (left) and Y (right). As indicated by the dashed lines, 97% of the prediction errors are within 6-nm and 5-nm accuracy for overlay in X and Y, respectively.

3.2.1 Neural network architecture

Neural networks are a class of machine learning algorithms that can approximate a function directly from data. Feed-forward neural networks consist of a set of neurons (nodes) arranged in multiple layers, such that the connections between the neurons do not form a cycle. Based on the Universal Approximation Theorem,[9] a Feed-forward neural network can in theory represent any continuous function, given with appropriate parameters.

Figure 2 presents the architecture of our Feed-forward neural network, consisting of an input layer z_0, $n-1$ hidden layers, and an output layer z_n. The input layer z_0 is the mapping from a layer-point pair to a fixed-size feature set (cf. §3.1). The output layer z_n is a single continuous output, which holds the overlay errors for the given layer-point in one of the X or Y axes. Each hidden layer z_i is fully connected to the next layer, and is computed as:

$$z_i = f(W_i . z_{i-1} + b_i), \tag{6}$$

where the weight matrix W_i and the bias term b_i are fitted during training. The function f is the activation function, which is a non-linear transformation that is applied to each node.[10] For the output layer z_n, the identity activation function is employed.

3.2.2 Training

Let $T = \{(l_1, p_1, d_{l_1,p_1}), (l_1, p_2, d_{l_1,p_2}), ..., (l_n, p_m, d_{l_n,p_m})\}$ be the set of training instances. Each instance corresponds to the layer l_i and point p_j of a wafer, as well as its overlay error d_{l_i,p_i} in one axis. Our goal is to learn a function $\mathcal{D}(l, p; W, b)$ that estimates the overlay of layer l at position p in X (or Y) direction, given model parameters W and b. We consider the mean square error loss function to train the neural network model:

$$\mathcal{L}(T, W, b) = \frac{1}{|T|} \sum_{i=1}^{n} \sum_{j=1}^{m} (\mathcal{D}(l_i, p_j; W, b) - d_{l_i,p_j})^2 + \frac{\lambda}{2|T|} \sum_{k} w_k^2, \tag{7}$$

where the model parameters W and b are updated by computing the gradient of the loss function for a given mini batch of training instances. The second part of Eq. 7 is a regularization term, which demonstrates one among many other regularization techniques for neural networks.[8]

4. EXPERIMENTAL SETUP AND RESULTS

4.1 Experimental setup

For our experiments, we used on-product-overlay data of 2553 exposed wafer layers, provided by STMicroelectronics. These measurements correspond to 9 implant layers and belong to 264 lots. The layer that is used to estimate the machine-chuck fingerprints and the first measured overlay of each wafer are removed from our data.

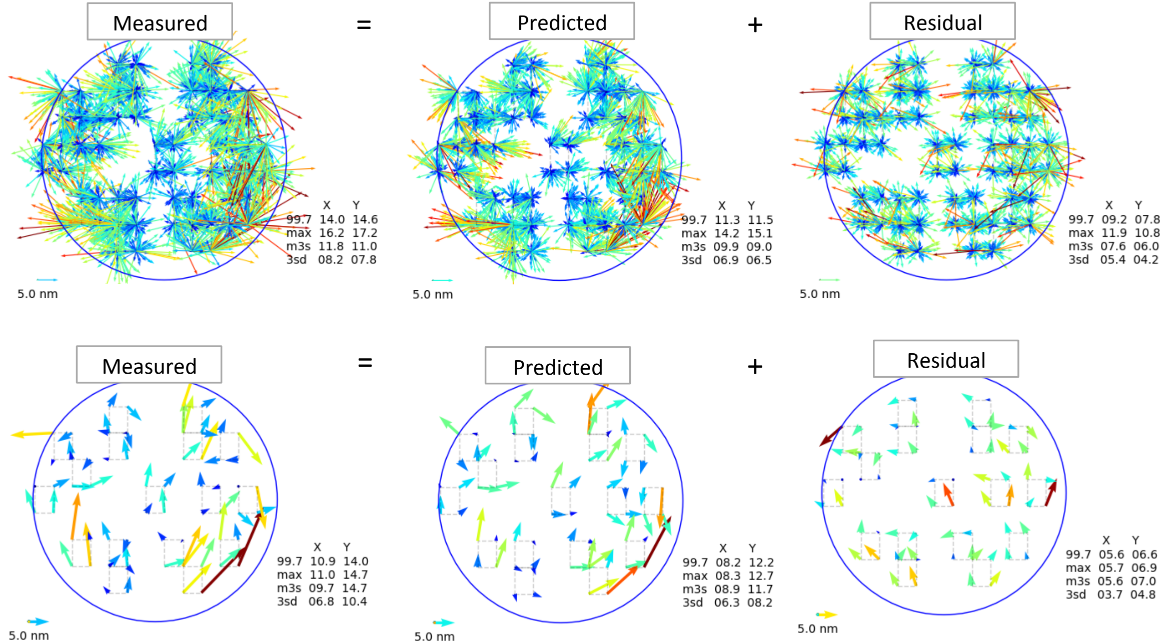


Figure 6. Wafer map of the measured, predicted and residual overlay for fifty stacked wafers are shown in the top row. The bottom row exhibits similar wafer maps for a single wafer.

At the end, we were left with around 122 thousand measured overlay values, which are used as our training and evaluation instances.

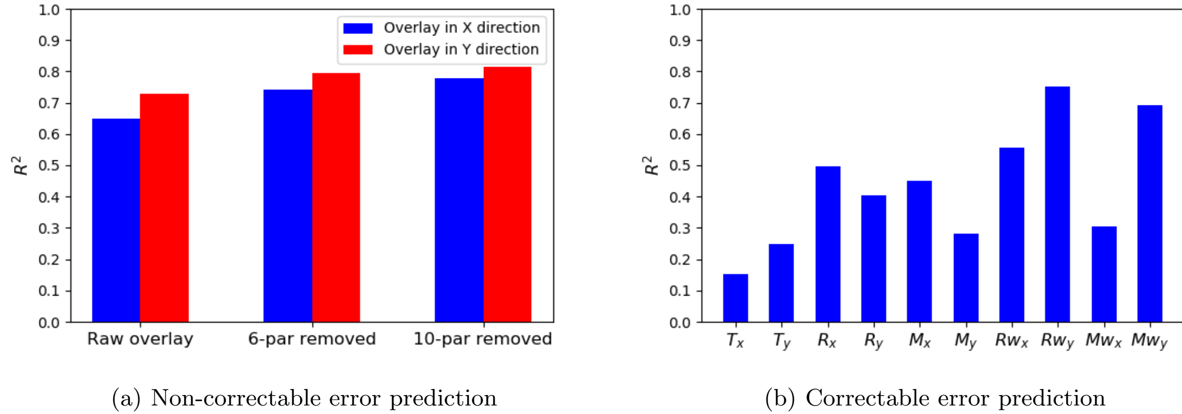
All the results are obtained by performing 5-fold cross validation. Following this approach, the measurement points are randomly divided into five groups, or *folds*, of approximately equal size. The first fold is treated as an evaluation set and the other four folds are used for training the prediction model. This procedure is repeated five times and each time a different fold is treated as evaluation set. Tuning the hyper parameters is performed using a separate train-validation splitting.

When performing the 5-fold cross validation, we ensured that all measurements from the same lot are assigned to the same fold, see Figure 3 for illustration. This is done to avoid information being leaked from the training to the test sets. Note that when train-test splitting is performed based on other strategies (i.e., points from the same lot/wafer/layer appear in both training and test sets), information leakage is substantial and prediction performance is significantly higher than what a model can achieve in the real world scenario in the production fabs (up to 24% difference in our setting). To report the results, we use the R^2 coefficient of determination to measure the prediction performance of our model. The R^2 statistic is based on the proportion of the variance in the real values explained by the prediction model.

4.2 Results

Figure 4 shows the results of our overlay prediction model. In the two scatter plots, the predicted overlay (X-axis) is plotted against the corresponding measured overlay (Y-axis). The left scatter plot corresponds to the overlay in X direction with an R^2 of 0.6504 as a measure of prediction accuracy. The right scatter plot shows R^2 of 0.7302 for overlay in Y direction, which is higher than prediction performance for overlay in X direction

The better prediction accuracy for overlay in Y direction as compared to X direction is also reflected in Figure 5. This graph exhibits histograms of the prediction errors for both X (blue) and Y direction (red). We observe that for both directions, the prediction error is Gaussian distributed and centered around zero. However, the distribution of the X-errors is significantly wider. As indicated by the dashed vertical lines, 97% of the prediction error for overlay in X and Y is within 6-nm and 5-nm accuracy, respectively.



(a) Non-correctable error prediction

(b) Correctable error prediction

Figure 7. Prediction performance for non-correctable and correctable errors. The left bar plot shows the prediction performance expressed in R^2 for the raw overlay, overlay with 6-parameters removed and overlay with 10-parameters removed. The right bar plot exhibits the prediction capability for the correctable errors by reporting the R^2 values for each parameter.

If we look in more details to the prediction errors (Figure 4), then we see that the outliers (the extreme values of overlay such as $d_x < -30$ and $d_y < -20$) are often estimated with lower overlay values than measured. Similar observations can be obtained from Figure 6, which exhibits six wafer plots. The top left wafer plot represents the measured overlay for a subset of 50 wafers. Each vector on the map represents an overlay measurement and there are 50 vectors per location, one for each wafer. The center wafer plot in the top row represents the corresponding maps for the predicted overlay values. By subtracting the measured and predicted overlay point-by-point one obtains the residual maps, which are represented by the upper right wafer plot. From the statistics it is obvious that the predicted overlay is generally lower than the measured overlay.

A significant part of the residual maps consist of correctable errors. With this we mean that the ten parameters, translation T_x and T_y, field rotation R_x and R_y, field magnification M_x and M_y, wafer rotation Rw_x and Rw_y, and wafer scaling Mw_x and Mw_y contribute significantly to these residuals maps. For example, the bottom row in Figure 6 shows the wafer plots for one particular wafer and the residual plot clearly exhibits translation errors in Y direction. To further evaluate the prediction capability of our model, we have compared the R^2 values of predicting the raw overlay, overlay with 6-parameters (T_x, T_y, Mw_x, Mw_y, Rw_x, and Rw_y) removed, and the overlay with all 10-parameters removed. This comparison is shown in the left bar plot of Figure 7 and it shows that our model can already achieve high accuracy in predicting the non-correctable part of the overlay. After removing 10-par correctable errors from the overlay data, an R^2 value of approximately 0.8 is achieved for both directions.

The right bar plot in Figure 7 exhibits the prediction capability with respect to the correctable errors by reporting the R^2 values for each individual parameter. These R^2 values are obtained for each parameter by comparing the measured and predicted overlay. We observe that the prediction capability is significantly better for Rw_y and Mw_y (with R^2 values of approximately 0.7), compared to the other parameters. Improvements are definitely required for predicting T_x, T_y, M_y, and Mw_x parameters in overlay, for which the R^2 values are below 0.3.

Virtual overlay metrology will be established in a production environment only when its prediction capability is accurate and robust. This requires good prediction capability for both the non-correctable and correctable overlay errors. Good prediction capability is already achieved for the non-correctable errors and some correctable errors (parameters). The prediction performance for other correctable errors needs further improvements. We have already identified several improvement areas, for example by designing specifically engineered features related to wafer and stage alignment. In the future, we focus on further improving the prediction accuracy for the correctable overlay errors.

5. CONCLUSIONS

To conclude, we have presented a machine learning based approach for predicting overlay for a series of implant layers. The prediction model has been built using a neural network model, fed with a set of specifically engineered features based on the physical concepts of overlay. Three categories of features have been employed: (i) measured overlay of one particular implant layer, which captures information about other implant layers of the same wafer, (ii) exposure tool fingerprints, which captures the contribution of the scanner and its chucks on overlay, and (iii) scanner logging, applied APC corrections, and context information.

To evaluate our overlay prediction model, we have used production data, consisting of approximately 122 thousand overlay measurements. Measured in terms of R^2 statistic, we have achieved a prediction capability of 0.6504 and 0.7302 for overlay in X and Y direction, respectively. Considering only the prediction of non-correctable overlay errors, we have obtained an even higher prediction performance of $R^2 \approx 0.8$. The correctable overlay errors, with the translation errors in particular, are predicted with a relatively lower accuracy as compared to the non-correctable errors, and requires further improvements.

Future work is aimed towards improving the prediction capability of our model, both for non-correctable and correctable overlay errors. The improvement areas include further development of the neural network architecture and designing engineered features related to the underlying physical concepts of some correctable errors (e.g. translation). Implementation of these improvements is required to obtain reliable and accurate virtual overlay metrology. Only then virtual metrology can be considered to replace real metrology in a production environment and further reduce the fab cycle time.

REFERENCES

[1] Gkorou, D., Ypma, A., Tsirogiannis, G., Giollo, M., Sonntag, D., Vinken, G., van Haren, R., van Wijk, R. J., Nije, J., and Hoogenboom, T., "Towards big data visualization for monitoring and diagnostics of high volume semiconductor manufacturing," *Proc. of the Computing Frontiers Conference* , 338–342, ACM (2017).

[2] Lam, A., Ypma, A., Gatefait, M., Deckers, D., Koopman, A., van Haren, R., and Beltman, J., "Pattern recognition and data mining techniques to identify factors in wafer processing and control determining overlay error," *Proc.SPIE* **9424**, 9424 – 9424 – 10 (2015).

[3] Lam, A., Pasqualini, F., de Caunes, J., and Gatefait, M., "Overlay breakdown methodology on immersion scanner," *Proc. of SPIE* **7638**, 7638 – 7638 – 12 (2010).

[4] Giollo, M., Lam, A., Gkorou, D., Liu, X. L., and van Haren, R., "Machine learning for fab automated diagnostics," *Proc. of SPIE* **10446**, 10446 – 10446 – 8 (2017).

[5] Laidler, D., Leray, P., Dhave, K., and Cheng, S., "Sources of overlay error in double patterning integration schemes," *Proc. of SPIE* **6922**, 6922 – 6922 – 11 (2008).

[6] Tibshirani, R., "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)* **58**(1), 267–288 (1996).

[7] Breiman, L., "Random forests," *Machine Learning* **45**, 5–32 (Oct 2001).

[8] Goodfellow, I., Bengio, Y., and Courville, A., [*Deep Learning*], MIT Press (2016).

[9] Hornik, K., Stinchcombe, M., and White, H., "Multilayer feedforward networks are universal approximators," *Neural Networks* **2**(5), 359 – 366 (1989).

[10] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *nature* **521**(7553), 436 (2015).

View publication stats