PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Machine learning for mask/wafer hotspot detection and mask synthesis

Yibo Lin, Xiaoqing Xu, Jiaojiao Ou, David Z. Pan

SPIE.

Invited Paper

Machine Learning for Mask/Wafer Hotspot Detection and Mask Synthesis

Yibo Lin, Xiaoqing Xu, Jiaojiao Ou, David Z. Pan,

ECE Department, University of Texas at Austin, Austin, TX USA
Email: {yibolin, xiaoqingxu, jiaojiaoou}@cerc.utexas.edu; dpan@ece.utexas.edu

ABSTRACT

Machine learning is a powerful computer science technique that can derive knowledge from big data and make predictions/decisions. Since nanometer integrated circuits (IC) and manufacturing have extremely high complexity and gigantic data, there is great opportunity to apply and adapt various machine learning techniques in IC physical design and verification. This paper will first give an introduction to machine learning, and then discuss several applications, including mask/wafer hotspot detection, and machine learning-based optical proximity correction (OPC) and sub-resolution assist feature (SRAF) insertion. We will further discuss some challenges and research directions.

1. INTRODUCTION

The recent success of machine learning in various fields such as pattern recognition for images and speeches, data mining, and artificial intelligence (AI)[1–3] raises significant interests in its research and applications. Machine learning can be briefly explained as the procedure of learning/training from data and making predictions. It has a substantial impact on the ubiquitous applications from devices to systems and software.

However, the expansion of machine learning from scientific and engineering communities to public mainly comes from the landmark victory by AlphaGo from Google DeepMind in 2016,[4] which uses the Monte Carlo tree search algorithm based on deep neural network (DNN), a branch of machine learning techniques.[5] The Go game is known to have considerably large real-time searching space, e.g., 2.08×10^{170} legal positions for a 19×19 board, which is believed to be extremely difficult for computing. Thus the victory proves the potential of machine learning to applications with large searching space and imperfect training data.

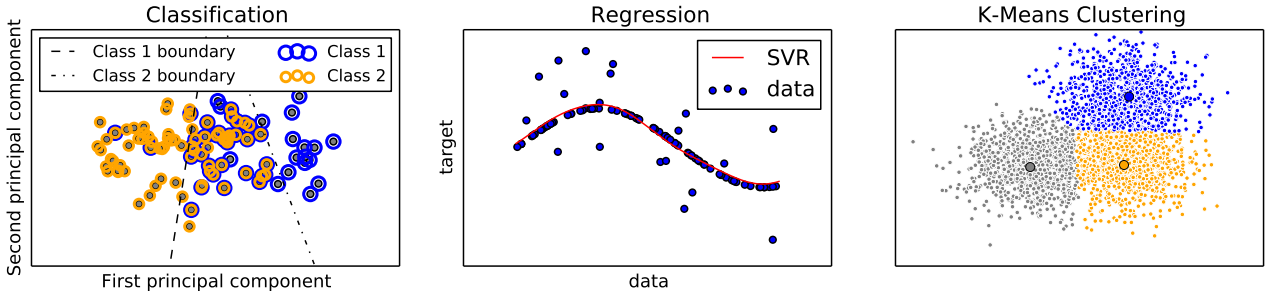
Another reason for the rise of machine learning lies in the booming of available data and demands for training and prediction. The whitepapers from Cisco in 2017 show that the annual global IP traffic will reach to 3.3 Zettabytes per year by 2021 and will increase nearly threefold in the next 5 years, indicating the explosion of data amounts on the Internet.[6] At the same time, the human communities are more connected with the Internet of Things (IoT), where the remote access of edge devices such as local refrigerators and ovens becomes possible. The communications between devices will generate huge amount of data that can be used for the modeling of the environmental variations, instruction patterns, and any other application to improve the performance of devices as well as human experience.

1.1 Machine Learning Tasks

Typical machine learning problems are categorized as supervised learning, unsupervised learning, reinforcement learning, etc. In supervised learning, each data sample consists of a feature and a label. The model is trained to produce desired labels from input features. In unsupervised learning, the target is to learn the hidden structures in the features, like that in clustering, rather than to make predictions, since no labels are given to the learning algorithm. Reinforcement learning refers to the interaction scheme with the external environment such as rewards or punishments according to the decisions the algorithm makes, which is widely adopted in robotics.

Besides various types of machine learning problems, machine learning tasks are often divided into classification, regression, and clustering, where the main differences lie in the characteristics of labels. Classification divides inputs into two or more classes where each class corresponds to a label, as shown in Fig. 1(a) using

Photomask Technology, edited by Peter D. Buck, Emily E. Gallagher, Proc. of SPIE Vol. 10451,
104510A · © 2017 SPIE · CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2282943



(a) (b) (c)

Figure 1: (a) Support vector machine for classification and (b) support vector machine for regression and (c) k-means clustering.

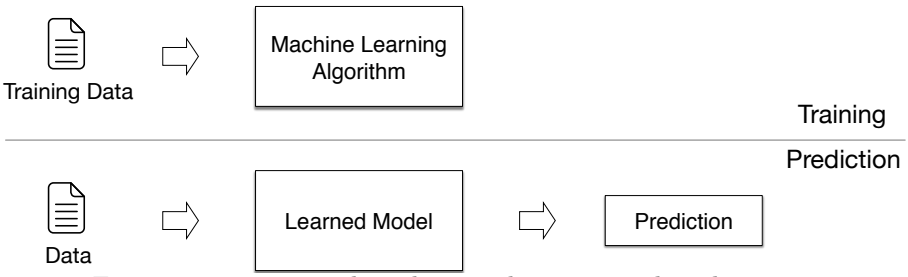


Figure 2: Training and prediction phases in machine learning.

support vector machine (SVM). For example, in the classification of positive and negative reviewers' comments, "positive" comments are labeled as "class 1" and "negative" ones are labeled as "class 2". Regression requires the algorithm to produce a continuous value for each input rather than a discrete one, as shown in Fig. 1(b) where SVM is applied to fit the data. It can be applied to model continuous labels like the optical simulation for aerial images. Both classification and regression are typically problems of supervised learning since labels are needed, while clustering is an unsupervised learning task that divides data into groups, as shown in Fig. 1(c) where data is clustered into three groups.

1.2 Machine Learning Flow

Typical procedure of machine learning consists of training and prediction phases, as shown in Fig. 2. In training phase, training data is required by a specific machine learning algorithm to learn/calibrate the model. In the prediction phase, the learned model is then used to make the prediction for new data. Generally, in supervised learning, the training data also contains labels for training, while the data for prediction phase needs the learned model for the prediction of labels. Most studies on machine learning focus on the training phase since it is very critical in the model accuracy and often time-consuming to fit the model.

1.3 Machine Learning Algorithms

Popular machine learning algorithms include logistic regression,[7] AdaBoost,[8] SVM,[9] various neural networks,[10,11] etc. These algorithms provide different formats of models that can be trained to fit the data for a wide range of applications.

Logistic regression adopts the logistic function as the probabilistic estimation and the model is calibrated with maximum likelihood method.[12] Its mathematical formulation for training is shown as follows,[13]

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_i \log(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}), \tag{1}$$

where $\boldsymbol{w}$ is the vector of weight parameters determined during training, $\boldsymbol{x_i}$ and y_i are features and label (-1 or 1 for two-class classification) for i^{th} data sample, respectively. The first term $\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$ denotes the $L2$ regularization to avoid overfitting. The second term denotes the overall error cost. Parameter C sets the importance of the regularization term. Thus the objective for training is to minimize the overall error cost with $L2$ regularization.

Support vector machine defines a hyperplane that maximizes the margin between the decision boundaries, as shown in Fig. 4(a), where blue and orange circles represent data points in two classes. The optimal hyperplane is shown as the solid line and the data points encompassed by gray squares correspond to support vectors that decide the decision boundaries in dashed lines. The objective for training is to minimize the margin between two dashed lines. The detailed mathematical formulation for SVM with linear kernel is defined as follows,[14]

$$\min_{\boldsymbol{w},b,\xi} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_i \xi_i, \tag{2a}$$

$$\text{s.t.} \quad y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) \geq 1 - \xi_i, \tag{2b}$$

$$\xi_i \geq 0, \quad \forall i, \tag{2c}$$

where $\boldsymbol{w}, b, \xi$ are parameters that need to be determined during training. Parameter b is the bias for the hyperplane and ξ denotes the error for the i^{th} data sample. The objective function consists of the term for error minimization and that for $L2$ regularization like that in logistic regression.

The accuracy of learning algorithms like logistic regression and SVM is highly correlated to the performance of feature extraction, shown as the "traditional machine learning flow" in Fig. 3. Features with a good representation of the input data but usually in lower dimensions are extracted for the machine learning algorithm, while it still needs manual efforts to search for the suitable feature representations according to the learning tasks and distributions of input data. However, deep learning algorithm is able to extract features automatically with high accuracy and generality using raw input data. In other words, the feature extraction may no longer be a necessary step.

Deep learning is a class of learning algorithms that use a cascade of layers with linear or nonlinear transformations for feature extraction and prediction. The output of each layer is feed to the successive layer as input. Due to the flexibility of the transformation functions, deep learning can be applied to various fields. Most algorithms for deep learning are based on neural networks, including deep artificial neural networks (ANN), convolutional neural networks (CNN) and recurrent neural networks (RNN) which have demonstrated the power in image classification and speech recognition.[15–17] The depth of neural networks refers to the number of layers which may vary from several to even 1000 layers.[18]

Here we give an example of a simple neural network with 2 hidden layers in Fig. 4(b) where the input dimension is 6 and output dimension is 1. The name of "hidden layer" is simply introduced to differentiate from input and output layers. With the data propagating through each layer, a specific transformation is applied as $f_i(x)$ to the ith layer. The training of the neural network tries to minimize an error function between the predicted values $f_3(f_2(f_1(x)))$ at the output layer and golden labels. Different from logistic regression or SVM which guarantees global optimality due to the convexity of the problem formulations, training a neural network usually results in solving a non-linear non-convex problem owing to the flexible function $f_i(x)$ at each layer that is likely to be non-linear. Therefore it is difficult to find optimal solutions. Nevertheless, despite the lack of theoretical insights, various empirical results have demonstrated its capability of convergence to local optimum solutions with high quality.[11,15–17]

While deep learning is powerful for difficult learning problems, it is computationally expensive and often requires hardware acceleration like graphics processing unit (GPU) or even tensor processing unit (TPU).[19] It is likely for the training process to take hours or even weeks according to the data volumes even with GPU. Therefore, learning algorithms with simpler models such as logistic regression and SVM are a good start if the data is not difficult to fit. Reducing the layers in neural networks also helps to simplify the models.

1.4 Feature Representations

Feature representations play a significant role in the performance of machine learning algorithms. While neural networks are able to extract features automatically, preprocessing to the raw input data sometimes still helps

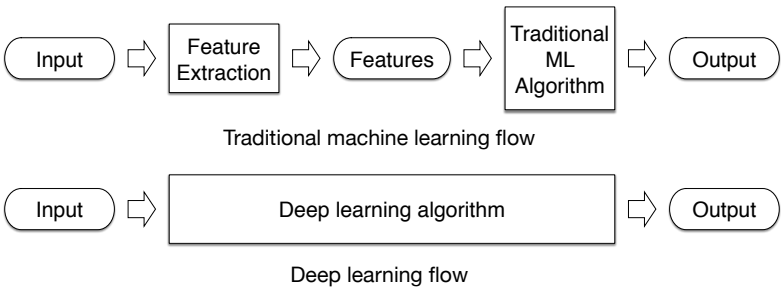
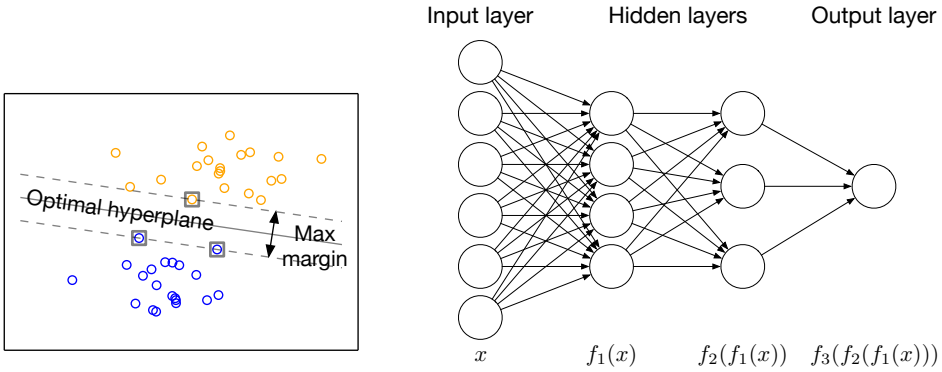


Figure 3: Traditional machine learning flow v.s. deep learning flow.



(a) (b)

Figure 4: Example of (a) SVM for classification and (b) a neural network with 2 hidden layers.

to improve the performance. Typical feature representations include pixel maps, density based sampling with preprocessing,[20] concentric square sampling (CSS),[21] concentric circle area sampling (CCAS),[22] etc. Details will be explained later with specific applications.

The historic success of machine learning in various fields raises the interests of applying the learning approaches to modeling problems in VLSI design and manufacturing. This paper will focus on the applications of machine learning to mask/wafer hotspot detection and mask synthesis including optical proximity correction and sub-resolution assist feature insertion. We will explain the motivations of tasks and benefits of machine learning based approaches to conventional modeling techniques.

The rest of the paper is organized as follows. Section 2 introduces the machine learning applications to mask/wafer hotspot detection. Section 3 explains the optimization challenges in manufacturing and how machine learning can help. Section 4 comes up with the conclusion.

2. MACHINE LEARNING FOR MASK/WAFER HOTSPOT DETECTION

With the continuous scaling of technology nodes, the printability of masks has been seriously affected by the limitation of light wavelength. To address the challenges and improve layout pattern printability, various resolution enhancement techniques (RETs), such as optical proximity correction (OPC), source mask co-optimization, and sub-resolution assist features (SRAFs) have been proposed. However, due to the complexity of lithography system and process variation, failure to print specific patterns still happen even with RETs, which is known as lithography hotspot. The early detection of lithography hotspot remains to be a critical step to enhance manufacturability and reduce costs. Although lithography simulation is often accurate enough for hotspot detection, it is also extremely time-consuming. Therefore, it is imperative to develop efficient hotspot detection approaches with high accuracy for the reduction of the overall turn-around time.

Generally, if two sets of hotspots and non-hotspots layout clips are given, the task of hotspot detection is to construct a model based on the given data and classify hotspots on testing layouts. Examples of two hotspot layouts are shown in Fig. 5. The evaluation metrics of hotspot detection include detection accuracy and false

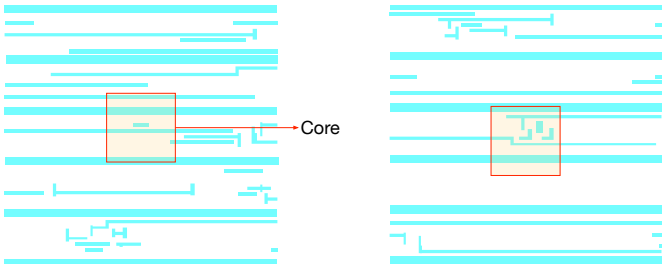


Figure 5: Example of hotspot patterns marked in red.[23]

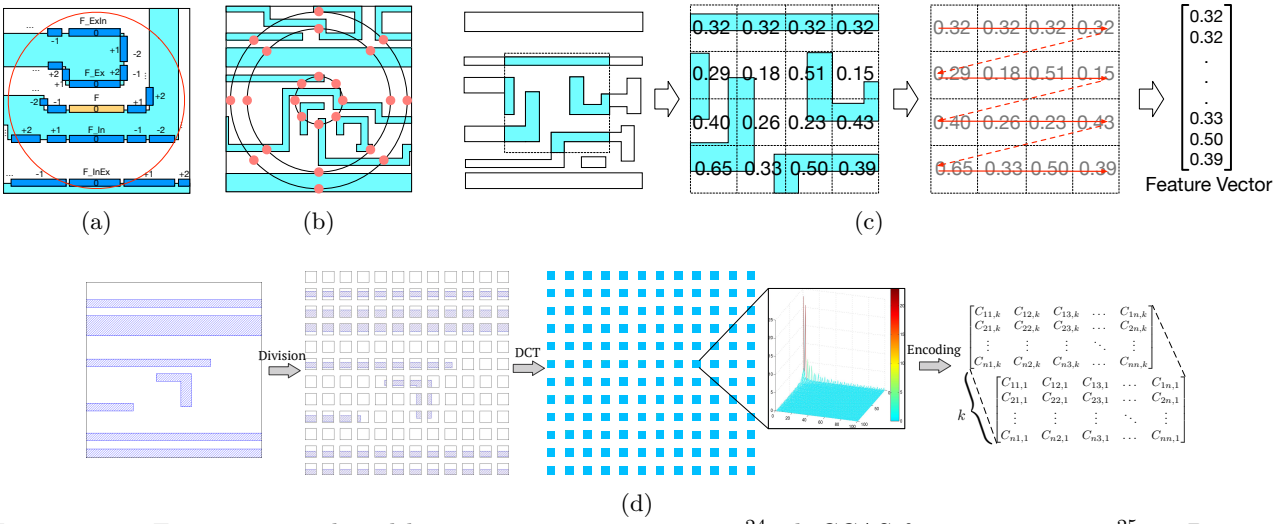


Figure 6: 6a Fragmentation based hotspot signature extraction.[24] 6b CCAS feature extraction.[25] 6c Density-based pattern representation.[26] 6d Feature tensor generation.[20]

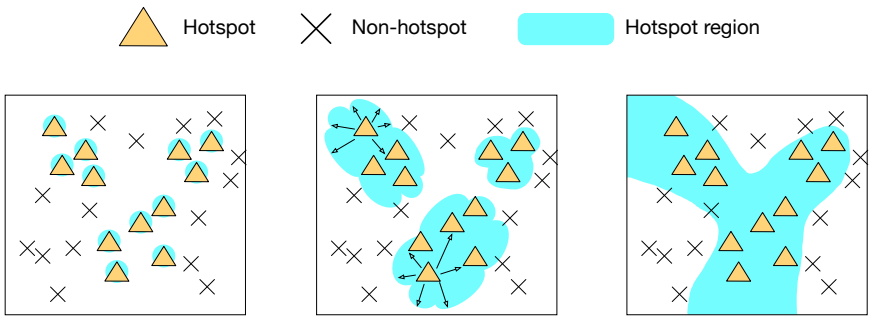
alarm. The *detection accuracy* is the ratio between the number of correctly detected hotspots and the number of real hotspots, while the *false alarm* is defined as the number of non-hotspots that are recognized as hotspots. This section presents different hotspot detection techniques for mask and wafer.

There are two aspects that directly affect the performance of hotspot detection: layout feature extraction and model selection, which will be covered in the following sections.

2.1 Layout Feature Extraction and Encoding

The layout feature extraction is a fundamental step for hotspot detection. It should represent the layout attributes of hotspots and non-hotspots. Different layout feature representations have been proposed to improve the accuracy and reduce false alarms, such as density based feature,[27,28] fragmentation based feature, and concentric circle area sampling (CCAS).[26,29,30]

The fragmentation based feature extraction is illustrated in Fig. 6(a). An effective radius r is defined to cover neighboring fragments of each fragment F. The representation of F includes geometric characteristics of fragments covered by the circle, such as pattern shapes, distances between layout and corner information. The density-based feature extraction is illustrated in Fig. 6(c). The layout is represented as a vector of pattern densities which is calculated as the ratio of the layout and the area of each grid. The concentric circles with area sampling is proposed to capture the layout information that matches the diffraction of lights, as shown in Fig. 6(b). Since all the features extracted from the layouts are stored in a feature vector, Yang et al.[20] argue that such kind of representations lose the spatial information. They propose a feature tensor representation to keep the spatial information of the layout. As shown in Fig. 6(d), the original clip is converted to a hyper-image after feature tensor extraction. In this example, the original clip is divided into 12×12 blocks and each block



(a) (b) (c)

Figure 7: A 2D-space example of hotspot region decision. (a) Pattern matching. (b) Fuzzy Pattern Matching. (c) Machine learning.[25]

is converted to 100×100 images. The feature tensor is obtained after applying discrete cosine transformation (DCT) to each block.

2.2 Pattern Matching

As the hotspot patterns often follow some characteristics, pattern matching is adopted for the detection problem by keeping a set of pre-characterized hotspot patterns stored in a library. Pattern matching based methods discover the problematic regions by comparing the topology of the input patterns with that of the patterns in the hotspot library. The performance of pattern matching based hotspot detection highly relies on the generality of the hotspot library. Thus these methods suffer from poor performance to unknown topologies in advanced technology nodes.[31,32] In order to improve the hotspot detection result, a fuzzy matching model is proposed to dynamically tune the regions around the known hotspot.[25] As shown in Fig. 7, the pattern matching approach can detect each known hotspot, while the fuzzy region can iteratively grow to provide better accuracy.

2.3 Conventional Machine Learning

Different from pattern matching, machine learning models are not limited to patterns in the hotspot library. On the contrast, it is able to predict hotspots for any input pattern. Recent studies on machine learning based hotspot detection have demonstrated its detection accuracy and efficiency with advanced technology nodes. There are two major aspects to consider when applying machine learning to hotspot detection: feature extraction and model selection/training.

Feature extraction has critical impacts to the accuracy and generality of the machine learning models. Although simple and low-dimensional layout features may reduce the training time, it can be too rough to achieve high accuracy. Complicated and high dimensional layout features may result in over-fitting and long runtime.

Besides feature extraction, it is also challenging to design machine learning algorithms that can simultaneously achieve high accuracy and low false alarms with small training data set. Various machine learning models have been used as hotspot detection kernels including support vector machine (SVM),[33,34] artificial neural network (ANN)[33] and boosting methods.[28,30] Zhang et al.[30] also propose an online learning scheme to verify newly detected hotspots and incrementally update the model.

2.4 Deep Learning

To tackle the feature extraction issue and improve the detection accuracy, deep neural network (DNN) classifier has been adopted for hotspot detection.[35,36] DNN is able to take the high-dimensional layout and perform automatic feature extraction during training, which avoids the manual efforts to reduce select feature extraction methods. Promising empirical results have been observed with DNN in several papers.[35–38] Fig. 8 gives a typical configuration of DNN structure.

In spite of the convenience in automatic feature extraction, it takes manual efforts to configure the DNN with high performance, such as the number and types of layers, which is still done through trial and error process.

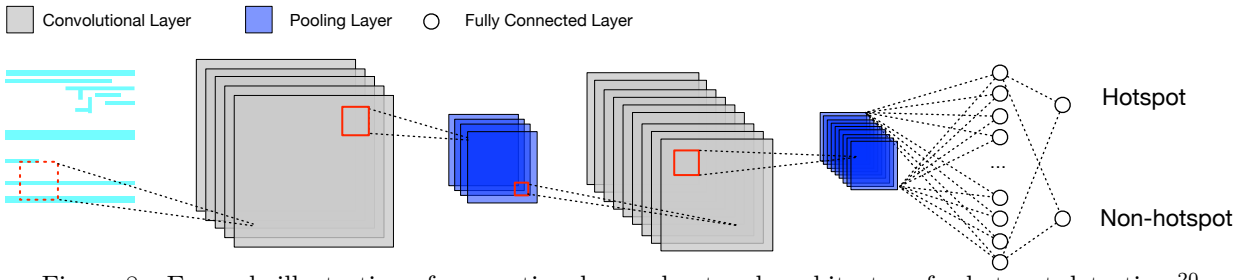


Figure 8: Example illustration of conventional neural network architecture for hotspot detection.[20]

Table 1: Comparison between the state-of-the-art hotspot detectors[20,37]

Bench	Train		Test		SPIE'15 AdaBoost[28]		ICCAD'16 Online[30]		DAC'17 Deep[20]		SOCC'17 Deep[37]	
	HS#	NHS#	HS#	NHS#	FA#	Accu (%)	FA#	Accu (%)	FA#	Accu (%)	FA#	Accu (%)
ICCAD	1204	17096	2524	13503	2919	84.2	4497	97.7	3413	98.2	1776	97.36
Industry1	34281	15635	17157	7801	557	93.2	1136	89.9	680	98.9	307	98.41
Industry2	15197	48758	7520	24457	1320	44.8	7402	88.4	2165	93.6	793	90.56
Industry3	24776	49315	12228	24817	3144	44.0	8609	82.3	4196	91.3	1723	83.63
Avg.	-	-	-	-	2397	66.6	5411	89.6	2613	95.5	1150	92.49
Ratio	-	-	-	-	0.92	0.70	2.07	0.94	1.0	1.0	0.44	0.97

Matsunawa et al.[36] propose a DNN structure that can achieve low false alarms. Yang et al.[20] propose a feature representation for DNN to speed up the feed-forward and back-propagation. They also propose a biased learning technique to improve the accuracy and decrease false alarms.

Table 1 shows the comparison between various state-of-the-art hotspot detectors on both ICCAD 2012 contest benchmarks and industrial designs.[20,37] Column "HS#" denotes the number clips with hotspots and column "NHS#" denotes the number of clips without hotspots. Column "Accu" denotes the accuracy and column "FA" denotes the false alarm. Although there might be other objectives in the problem formulations of difference detectors, the table reports the accuracy and false alarm for reference. Generally, deep learning achieves high accuracy with relatively low false alarm.[20,37] While the online boosting algorithm[30] mainly tries to reduce the overall detection and simulation time (ODST) using online learning, it can still achieve reasonable accuracy.

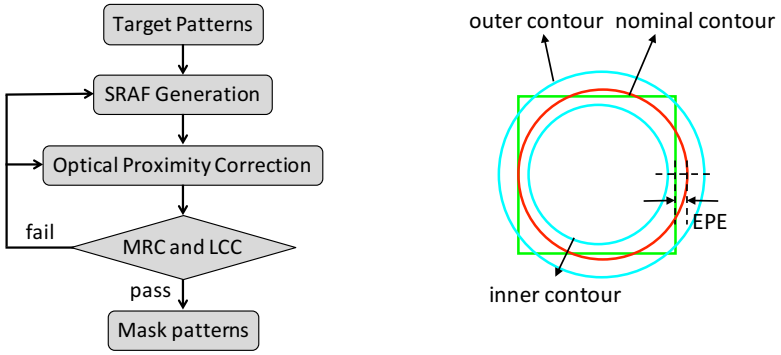
3. MACHINE LEARNING FOR MASK SYNTHESIS

In this section, we show the application of machine learning in mask synthesis problems.

3.1 Mask Synthesis Flow

A standard mask synthesis flow is shown in Fig. 9(a), which takes target patterns (layout) as input and generates mask patterns for robust lithography printing. The entire flow runs iteratively for better lithography printing, where each iteration involves SRAF generation, OPC, mask rule check (MRC) and lithography compliance check (LCC). SRAF generation means sub-resolution assist features are inserted around target patterns to benefit the printing of original target patterns. OPC means the edge segments of target layout are shifted to contribute to robust lithography printing. The MRC further checks whether mask patterns are manufacturing friendly by following a set of mask manufacturing rules. The LCC conducts the lithography simulations under a set of process windows to check whether robust lithography printing is achieved or not.

The evaluation of process windows is shown in Fig. 9(b), where the lithography simulations are performed under a set of {focus, dose} conditions to generate a set of printing contours, i.e. nominal, inner and outer contour. To quantify the process windows of mask patterns, the edge placement error (EPE) is defined as the distance between the target pattern contour and the nominal contour. The process variation (PV) band is defined as the area between the inner and outer contour. EPE and PV band shall both be minimized to obtain robust lithography printing.



(a) (b)

Figure 9: Mask synthesis: (a) a standard mask synthesis flow, (b) lithography simulation contours under a set of {focus, dose} conditions.[39]

3.2 SRAF Generation

SRAF generation is one of the most important RETs for robust lithography printing in advanced technology nodes. The SRAFs are within the sub-resolution domain and assist the printing of target patterns without printing themselves. Fig. 10 demonstrates the benefit from SRAFs for an isolated contact. The lithography contours of the isolated contact without and with SRAFs are shown in Fig. 10(b) and Fig. 10(c), respectively. With SRAFs inserted, much smaller PV band can be achieved than the case in Fig. 10(b). Since SRAFs deliver light to target-pattern positions in a proper phase, the target patterns can be printed more robustly. It is becoming increasingly important to develop fast yet high-quality SRAF generations to improve the yield of lithography printing.[39,40]

Conventional SRAF generation includes model-based and rule-based approaches, which have been widely adopted in semiconductor manufacturing industry. Model-based approaches[41–46] lead to high-quality and robust lithography printing with expensive computational cost. In other words, model-based approaches are not scalable to large layout designs. Rule-based approaches[47–49] are based on complicated look-up-tables, which leads to super fast turnaround time. However, the performance of rule-based approaches highly depends on the size of the look-up-tables which require significant amounts of engineering efforts to enumerate different layout configurations.[39]

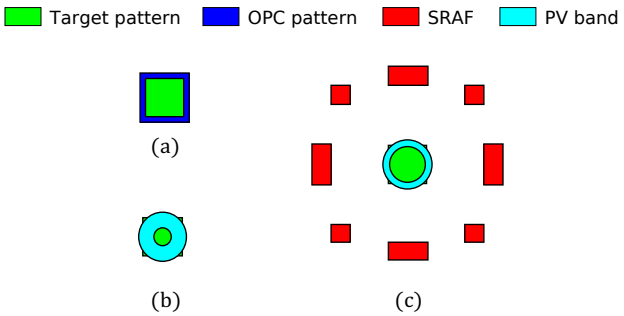
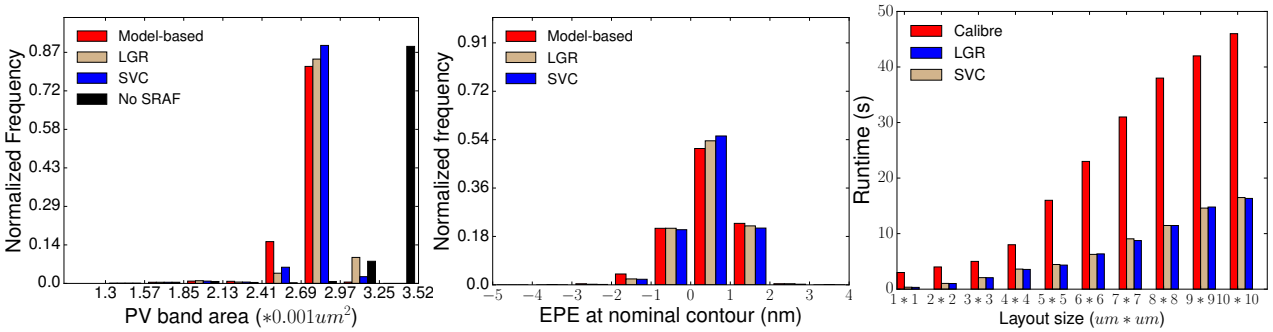


Figure 10: (a) An isolated contact, (b) printing with OPC only, (c) printing with SRAF generation and OPC.[39]

Supervised learning is introduced to improve the turnaround time from model-based approaches with the high quality of SRAFs.[39] For the SRAF generation problem, in the training phase, the mathematical models are calibrated with the high-quality SRAFs generated by the model-based approaches. In the testing phase, the calibrated model is applied to the target patterns to obtain fast yet high-quality SRAFs. The SRAF generation is formulated into a classification problem, where CCAS is adopted as the feature extraction technique and both logistic regression and SVM are used as the kernels for the learning models.



(a) (b) (c)

Figure 11: Comparison among different schemes in terms of, (a) PV band distribution, (b) EPE distribution at nominal conditions, (c) runtime.[39]

Fig. 11 compares the lithography performance of various SRAF generation approaches, in terms of EPE, PV band and runtime. The model-based SRAF generation is implemented with Mentor Calibre using industry-strength setup. Fig. 11(a) shows that SRAF generations ("Model-based", "LGR" for logistic regression and "SVC" for SVM based classification) significantly reduce the PV band area comparing with the case of "no SRAF". SVM based classification generates slightly smaller PV band area than logistic regression. Both learning-based approaches lead to a slightly larger average PV band area than that of the "Model-based" approach, while the EPE values are marginally better as shown in Fig. 11(b). The significant advantage of the learning-based approaches comes from the runtime as shown in Fig. 11(c), where the learning-based approaches can obtain >3X speed up for a layout clip with $10\mu m \times 10\mu m$ size due to the efficient prediction of the learned models.

3.3 Optical Proximity Correction

OPC is another important step to improve the manufacturing yield for advanced lithography. Fig. 12 illustrates the widely-adopted OPC technique, where the edges of design target layout are fragmented and each segment is shifted according to the optical environment such that the final wafer image is robustly printed, i.e., the EPE values are minimized. Traditional model-based OPC approaches[21,50] can generate high-quality results but they are known to be time-consuming. To overcome the runtime overhead, linear regression based[21,51] and nonlinear regression-based[52,53] approaches have been proposed to achieve fast full-chip OPC results with an acceptable performance loss.

Regression model for OPC is divided into training phase and testing phase, as shown in Fig. 13. Edge fragmentation is a standard step for both model-based OPC and machine learning based approaches. In the training phase, both model-based OPC and feature extraction are required to calibrate the model, while in the testing phase, only feature extraction is needed to validate the model. However, previous regression-based techniques suffer from the overfitting issues, which introduce non-negligible accuracy loss for OPC results in the testing phase. Moreover, the problem complexity of the OPC is becoming increasingly high due to the complicated optical proximity effects toward the sub-resolution domain. Therefore, it is very hard to achieve a highly complex yet accurate regression model.

To overcome the aforementioned issues, a hierarchical Bayes model (HBM) is proposed for the OPC problem with CCAS feature extraction.[22] The HBM trains a generalized linear mixed model (GLMM) to explicitly consider various edge types, such as normal, convex, concave and line-end edge. Each specific input edge type is treated as a random effect with a random variance in GLMM. The HBM assumes a non-informative prior distribution for unknown variables which helps avoid the lack of prior information. Due to the unique properties mentioned above, HBM can generate much better OPC results compared with previous regression models.

Fig. 14 shows the comparison between HBM-based approach and model-based (MB) approach, where MB_ik denote OPC results from the k^{th} iteration of the MB-approach. The HBM-based approach can generate very competitive or even better EPE results than the 10^{th} iteration of the MB-approach. However, the zoom-in region for large EPE values demonstrates that better results from MB-approach (MB_i10) than that from the

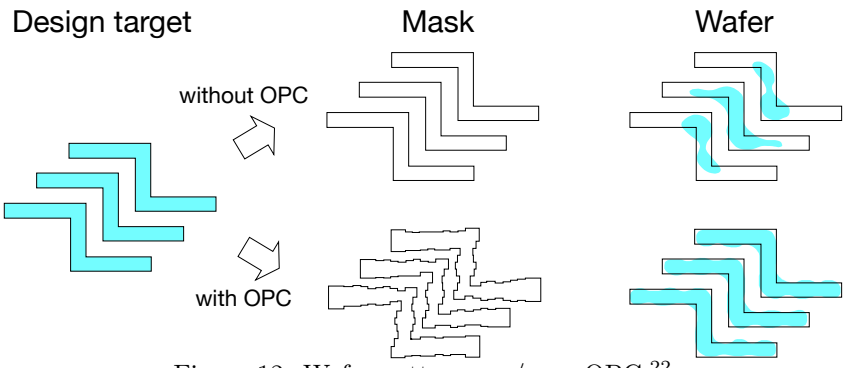


Figure 12: Wafer patterns w./w.o. OPC.[22]

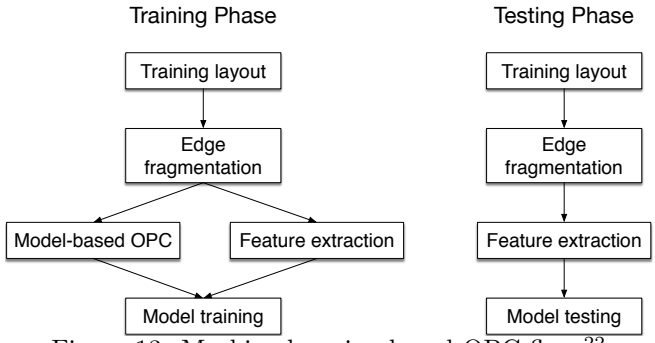


Figure 13: Machine learning based OPC flow.[22]

HBM-based approach because the MB_i10 can effectively reduce the large EPE values. While the overall results of the MB-based approach is still better than that of the HBM-based approach, it is suggested the proposed approach can provide valuable initial OPC conditions for model-based iterations to the overall runtime from model-based OPC.[22]

4. CONCLUSION

With the advances in machine learning, various techniques can be applied to semiconductor manufacturing such as mask/wafer hotspot detection and mask synthesis for better manufacturability and less turnaround time. Basic machine learning algorithms and the state-of-the-art applications to hotspot detection, OPC and SRAF insertion are reviewed. Promising results are reported which demonstrate the effectiveness of learning-based approaches.

There are still various open problems in applying learning-based approaches to VLSI manufacturing. For example, the general feature representation for layouts and masks is desired as most mask/wafer related applications are similar tasks in nature. In terms of learning techniques, how to reduce the amount of data required for training is quite important since it is expensive to obtain a large amount of manufacturing data. The configuration of DNN still requires manual trial and error and whether there exist optimal structures remains to be explored. Moreover, training time also becomes an issue due to the fact that DNN goes deeper for better performance. All these problems remain to be explored, which will ultimately push the advancement of the semiconductor industry for next generation VLSI designs and products.

5. ACKNOWLEDGE

This work is supported in part by NSF and Toshiba Memory Corporation. The authors would like to thank Dr. Tetsuaki Matsunawa and Dr. Shigeki Nojima from Toshiba Memory Corporation for helpful discussions.

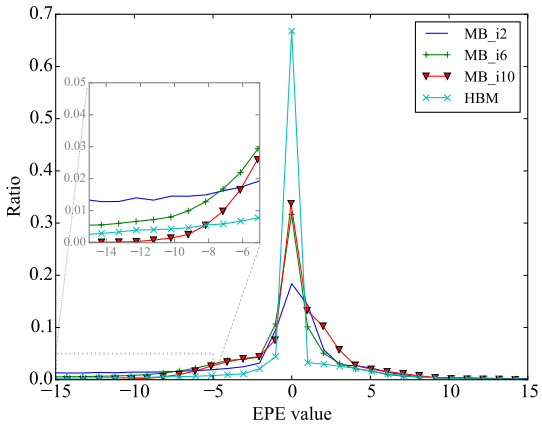


Figure 14: Compare HBM-based and model-based OPC in terms of EPE distributions.[22]

REFERENCES

[1] Alpaydin, E., [*Introduction to machine learning*], MIT press (2014).

[2] Chen, C.-h., [*Handbook of pattern recognition and computer vision*], World Scientific (2015).

[3] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J., [*Data Mining: Practical machine learning tools and techniques*], Morgan Kaufmann (2016).

[4] "Google DeepMind." `https://deepmind.com/`.

[5] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *Nature* **521**(7553), 436–444 (2015).

[6] "Cisco visual networking index: Forecast and methodology, 2016-2021." `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf`.

[7] Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X., [*Applied logistic regression*], vol. 398, John Wiley & Sons (2013).

[8] Freund, Y., Schapire, R. E., et al., "Experiments with a new boosting algorithm," in [*Icml*], **96**, 148–156 (1996). AdaBoost.

[9] Cortes, C. and Vapnik, V., "Support-vector networks," *Machine learning* **20**(3), 273–297 (1995). SVM.

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

[11] Goodfellow, I., Bengio, Y., and Courville, A., [*Deep learning*], MIT press (2016).

[12] Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J., "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer* **27**(2), 83–85 (2005).

[13] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J., "LIBLINEAR: A library for large linear classification," *Journal of machine learning research* **9**(Aug), 1871–1874 (2008).

[14] Chang, C.-C. and Lin, C.-J., "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology* **2**, 27:1–27:27 (2011). Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[15] Ciregan, D., Meier, U., and Schmidhuber, J., "Multi-column deep neural networks for image classification," in [*Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*], 3642–3649, IEEE (2012).

[16] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in [*Advances in neural information processing systems*], 1097–1105 (2012).

[17] Graves, A., Mohamed, A.-r., and Hinton, G., "Speech recognition with deep recurrent neural networks," in [*Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*], 6645–6649, IEEE (2013).

[18] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 770–778 (2016).

[19] Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al., "In-datacenter performance analysis of a tensor processing unit," *arXiv preprint arXiv:1704.04760* (2017). TPU.

[20] Yang, H., Su, J., Zou, Y., Yu, B., and Young, F. E., "Layout hotspot detection with feature tensor generation and deep biased learning," in [*ACM/IEEE Design Automation Conference (DAC)*], (2017).

[21] Gu, A. and Zakhor, A., "Optical proximity correction with linear regression," *IEEE Transactions on Semiconductor Manufacturing (TSM)* **21**(2), 263–271 (2008).

[22] Matsunawa, T., Yu, B., and Pan, D. Z., "Optical proximity correction with hierarchical bayes model," *Journal of Micro/Nanolithography, MEMS, and MOEMS* **15**(2), 021009–021009 (2016).

[23] Gao, J.-R., Yu, B., and Pan, D. Z., "Accurate lithography hotspot detection based on pca-svm classifier with hierarchical data clustering," in [*Proc. SPIE*], **9053**, 90530E (2014).

[24] Ding, D., Torres, J. A., and Pan, D. Z., "High performance lithography hotspot detection with successively refined pattern identifications and machine learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* **30**(11), 1621–1634 (2011).

[25] Lin, S.-Y., Chen, J.-Y., Li, J.-C., Wen, W.-y., and Chang, S.-C., "A novel fuzzy matching model for lithography hotspot detection," in [*ACM/IEEE Design Automation Conference (DAC)*], (2013).

[26] Zhang, H., Zhu, F., Li, H., Young, F. E., and Yu, B., "Bilinear lithography hotspot detection," in [*ACM International Symposium on Physical Design (ISPD)*], (2017).

[27] Wen, W.-Y., Li, J.-C., Lin, S.-Y., Chen, J.-Y., and Chang, S.-C., "A fuzzy-matching model with grid reduction for lithography hotspot detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* **33**(11), 1671–1680 (2014).

[28] Matsunawa, T., Gao, J.-R., Yu, B., and Pan, D. Z., "A new lithography hotspot detection framework based on adaboost classifier and simplified feature extraction," in [*Proceedings of SPIE*], **9427** (2015).

[29] Matsunawa, T., Yu, B., and Pan, D. Z., "Optical proximity correction with hierarchical bayes model," in [*Proceedings of SPIE*], **9426** (2015).

[30] Zhang, H., Yu, B., and Evangeline, Y. F., "Enabling online learning in lithography hotspot detection with information-theoretic feature optimization," in [*IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*], (2016).

[31] Andrew B. Kahng, Chul-Hong Park, X. X., "Fast dual graph based hotspot detection," in [*Proceedings of SPIE*], (2006).

[32] Yu, Y.-T., Chan, Y.-C., Sinha, S., Jiang, I. H.-R., and Chiang, C., "Accurate process-hotspot detection using critical design rule extraction," in [*ACM/IEEE Design Automation Conference (DAC)*], (2012).

[33] Ding, D., Yu, B., Ghosh, J., and Pan, D. Z., "Epic: Efficient predition of ic manufacturing hotspots with a unified meta-classfification formulation," in [*IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*], (2012).

[34] Yu, Y.-T., Lin, G.-H., Jiang, I. H.-R., and Chiang, C., "Machine learning based hotspot detection using topological classification and critical feature extraction," in [*ACM/IEEE Design Automation Conference (DAC)*], (2013).

[35] Shin, M. and Lee, J.-H., "Accurate lithography hotspot detection using deep convolutional neural networks," in [*Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*], (2016).

[36] Matsunawa, T., Nojima, S., and Kotani, T., "Automatic layout feature extraction for lithography hotspot detection based on deep neural network," in [*Proceedings of SPIE*], (2016).

[37] Yang, H., Lin, Y., Yu, B., and Young, F. E., "Lithography hotspot detection: From shallow to deep learning," in [*IEEE International System-on-Chip Conference (SOCC)*], (2017).

[38] Yang, H., Luo, L., Su, J., Lin, C., and Yu, B., "Imbalance aware lithography hotspot detection: A deep learning approach," in [*Proceedings of SPIE*], (2017).

[39] Xu, X., Lin, Y., Li, M., Matsunawa, T., Nojima, S., Kodama, C., Kotani, T., and Pan, D. Z., "Sub-Resolution Assist Feature Generation with Supervised Data Learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* **PP**(99) (2017).

[40] Xu, X., Matsunawa, T., Nojima, S., Kodama, C., Kotani, T., and Pan, D. Z., "A Machine Learning Based Framework for Sub-Resolution Assist Feature Generation," in [*ACM International Symposium on Physical Design (ISPD)*], 161–168 (2016).

[41] Sakajiri, K., Tritchkov, A., and Granik, Y., "Model-based sraf insertion through pixel-based mask optimization at 32nm and beyond," in [*Proceedings of SPIE*], 702811–702811 (2008).

[42] Viswanathan, R., Azpiroz, J. T., and Selvam, P., "Process optimization through model based sraf printing prediction," in [*Proceedings of SPIE*], 83261A–83261A (2012).

[43] Ye, J., Cao, Y., and Feng, H., "System and method for model-based sub-resolution assist feature generation," (Feb. 1 2011). US Patent 7,882,480.

[44] Shang, S. D., Swallow, L., and Granik, Y., "Model-based sraf insertion," (Oct. 11 2011). US Patent 8,037,429.

[45] Pang, L., Liu, Y., and Abrams, D., "Inverse lithography technology (ilt): a natural solution for model-based sraf at 45nm and 32nm," in [*Proceedings of SPIE*], 660739–660739 (2007).

[46] Kim, B.-S., Kim, Y.-H., Lee, S.-H., Kim, S.-I., Ha, S.-R., Kim, J., and Tritchkov, A., "Pixel-based sraf implementation for 32nm lithography process," in [*Proceedings of SPIE*], 71220T–71220T (2008).

[47] Ping, Y., McGowan, S., Gong, Y., Foong, Y. M., Liu, J., Qiu, J., Shu, V., Yan, B., Ye, J., Li, P., et al., "Process window enhancement using advanced ret techniques for 20nm contact layer," in [*Proceedings of SPIE*], 90521N–90521N (2014).

[48] Jun, J.-H., Park, M., Park, C., Yang, H., Yim, D., Do, M., Lee, D., Kim, T., Choi, J., Luk-Pat, G., et al., "Layout optimization with assist features placement by model based rule tables for 2x node random contact," in [*Proceedings of SPIE*], 94270D–94270D (2015).

[49] Kodama, C., Kotani, T., Nojima, S., and Mimotogi, S., "Sub-resolution assist feature arranging method and computer program product and manufacturing method of semiconductor device," (Aug. 19 2014). US Patent 8,809,072.

[50] Miyama, S., Yamamoto, K., and Koyama, K., "Large-area optical proximity correction with a combination of rule-based and simulation-based methods," *Japanese Journal of Applied Physics* **35**(12S), 6370 (1996).

[51] Jia, N. and Lam, E. Y., "Machine learning for inverse lithography: using stochastic gradient descent for robust photomask synthesis," *Journal of Optics* **12**(4), 045601:1–045601:9 (2010).

[52] Luo, R., "Optical proximity correction using a multilayer perceptron neural network," *Journal of Optics* **15**(7), 075708 (2013).

[53] Luo, K.-S., Shi, Z., Yan, X.-L., and Geng, Z., "SVM based layout retargeting for fast and regularized inverse lithography," *Journal of Zhejiang University SCIENCE C* **15**(5), 390–400 (2014).