2019 20th International Workshop on Microprocessor/SoC Test, Security and Verification (MTV)

Design Space Exploration For Hotspot Detection

Gaurav Rajavendra Reddy and Yiorgos Makris
gaurav.reddy@utdallas.edu, yiorgos.makris@utdallas.edu
Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX 75080, USA

***Abstract*—Advanced technology nodes suffer from high systematic defectivity. A significant part of such defectivity is caused by Lithographic Hotspots. Hotspots are Design Rule Check (DRC) clean areas of a layout which tend to show abnormal variation due to complex design-process interactions. In the past decade, we have witnessed several Pattern Matching (PM) and Machine Learning (ML)-based hotspot detection solutions. However, there is little research towards addressing two pressing needs of the industry: (a) Development of hotspot detection models during early stages of technology development, and (b) Detecting Truly-Never-Seen-Before (TNSB) hotspots in incoming layouts. In this work, we propose solutions to address these issues and quantitatively demonstrate the effectiveness of the proposed methodologies.**

I. Introduction

With extensive scaling, the Integrated Circuit (IC) fabrication process has become extremely complex. Among the unit processes, perfecting lithography is crucial and has become a major challenge. This is partly because, as shown in Figure 1, in the latest technology nodes the wavelength of light used for lithography is much larger than the features being printed. This rise in fabrication complexity is increasing the challenges in the manufacturability space, which in turn, affect the bottom-line of yield. One such manufacturability challenge is what is known as *Lithographic Hotspots*. Hotspots are certain areas of a layout which are DRC clean and, yet, show unexpected variation which causes defects [1]. Though the root causes of hotspots are not very well understood, it is believed that the polygons surrounding the hotspot area are responsible for causing defects.

More than a decade ago, the industry resorted to *Hotspot Detection* in order to prevent the yield loss caused by hotspots. The basic idea was to maintain a database of known hotspots which could be identified through inline inspections, diagnosis, failure analysis etc., and use this database to detect hotspots in incoming designs. Initially, Pattern Matching (PM) tools were used for this purpose [2]. PM tools demonstrated high accuracy and could potentially obtain zero false-alarms while performing an exact match to known hotspots. Such methods, however, could not generalize and identify hotspots which were slight variations of known hotspots. Consequently, fuzzy PM methods [3] were introduced, but they required the user to make a trade-off between accuracy and false-alarm rates. In 2007, for the first time, the use of Machine Learning (ML) for hotspot detection was proposed in [4]. Soon after, ML-based hotspot detection methods gained great popularity and were widely adopted by the community. Almost all ML-based methods proposed thereafter claim that they not only reduce false alarms but can also detect Never-Seen-Before (NSB) hotspots [5], [6], [7], [8], [9], [10], [11]. In [12], however, we demonstrated that the State-Of-The-Art (SOTA) ML-based hotspot detection methods could interpolate between known hotspots but could not detect Truly-Never-Seen-Before (TNSB) hotspots. Moreover, we also suggested potential solutions for TNSB hotspot detection. In this work, we implement one of those methods and quantitatively demonstrate TNSB hotspot detection.

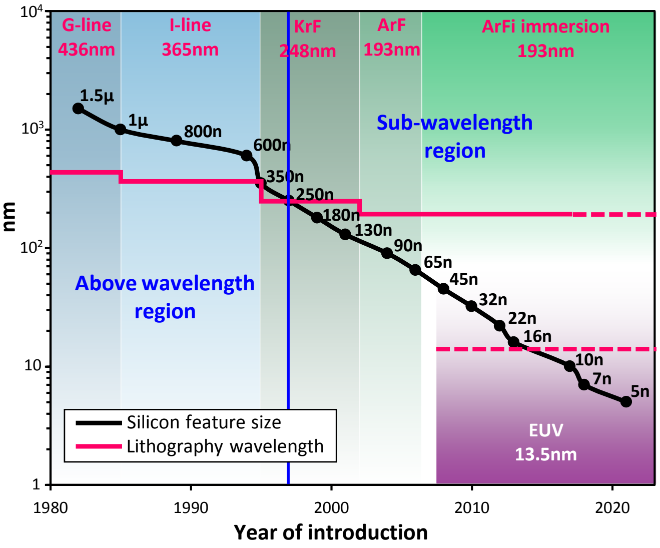

Fig. 1: Changes in lithography with silicon feature size (Adapted from [13])

Until now, major focus of hotspot detection work has been on its usage in fairly mature technology nodes. This is, mainly, because a database of known hotspots is not available during early stages of technology development. Therefore, early customers/designs of a technology node are unable to leverage the benefits of hotspot detection. Generating a database of hotspots in early stages of technology development would be very beneficial but it is extremely challenging. Authors of [14] proposed the use of a commercially available tool to generate synthetic layout patterns and explore the design space. Essentially, they suggest generating a large number of synthetic patterns, subjecting them to lithographic simulations / on-silicon experiments [15] to determine whether they are hotspots or non-hotspots and, thereby, building a hotspot database. Several variations of this methodology were proposed in [16], [17], [18]. All these methods presented qualitative results wherein

2332-5674/19/$31.00 ©2019 IEEE
DOI 10.1109/MTV48867.2019.00022

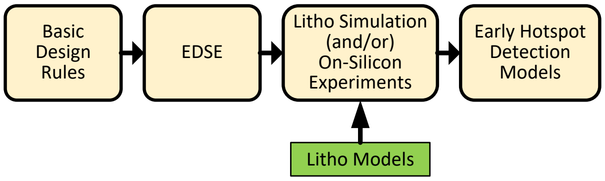
Basic Design Rules
EDSE
Litho Simulation (and/or) On-Silicon Experiments
Early Hotspot Detection Models
Litho Models

Fig. 2: The proposed methodology

examples of a few hotspots identified using their methods were shown, but none of them quantitatively demonstrated ML-based hotspot detection. In this work, we build an early hotspot detection model solely using synthetically generated training patterns and quantitatively demonstrate hotspot detection on test patterns from real designs.

The main contributions of this paper include:

1) The first quantitative demonstration of hotspot detection during early technology development.
2) The first quantitative demonstration of TNSB hotspot detection.

The rest of the paper is organized as follows. The proposed methodology is described in Section II. The experimental results are presented in section III and conclusions are drawn in Section IV.

II. Proposed Methodology

A. Hotspot Detection During Early Technology Development

The proposed methodology is shown in Figure 2. The basic design rules, which are usually available in the early stages of technology development, are fed into an Early Design Space Exploration (EDSE) tool. The EDSE tool generates a wide variety of DRC clean patterns. They are, then, subjected to lithographic simulations in order to identify potential hotspots for the given technology node. The generated database, consisting of hotspots and non-hotspots, is used to build early hotspot detection models.

B. TNSB Hotspot Detection

TNSB hotspots are test hotspot patterns which are very different in appearance in comparison to the hotspots found in the training dataset. When projected to a hyper-dimensional space, such test patterns are usually located farther away from the training patterns [12]. An illustration depicting TNSB patterns is shown in Figure 3a. It is challenging for ML entities to make predictions on such patterns, mainly because they are located in a space which is fairly 'unknown' to the ML entity. To empower an ML entity towards making accurate predictions on TNSB patterns, we use a methodology similar to the one detailed in the previous subsection. Essentially, we use the EDSE tool to explore the design space and find NSB hotspot patterns. We add such synthetically generated hotspot patterns into the training dataset and, thereby, increase its information-theoretic content and potentially transform previously unknown spaces into known spaces. Such transformation is illustrated in Figure 3b.

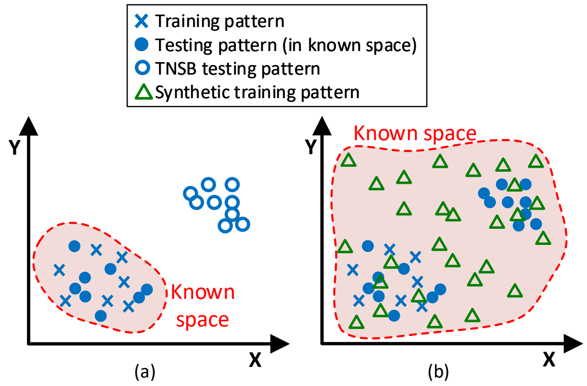

Fig. 3: Illustration of the impact of synthetic training patterns

C. Early Design Space Exploration

The EDSE process consists of a synthetic layout pattern generation tool at its core. An effective pattern generation tool must ensure that it generates a wide variety of random patterns, which accurately reflect various regions of the design space, and also ensures that the resulting patterns obey DRs and resemble real IC layout snippets. Several EDSE tools, including a commercially available tool [14], are available. In [19], we demonstrated that our pattern generation method, Versatile and Intuitive Pattern genERator (VIPER), performs about 3X better in terms of design space exploration in comparison to previously proposed methods. Therefore, in this work, we use VIPER for synthetic pattern generation. Further details of VIPER are out of the scope of this work and the interested reader is referred to [19].

D. Lithographic Simulations

Synthetic patterns generated using the EDSE process are subjected to lithographic simulations in order to determine their ground truth (hotspot/non-hotspot). Generally, a pattern is considered a hotspot when a defect(s) occurs in the central region of the pattern, as shown in Figure 4a [11]. We refer to such region under consideration as the *hotspot region*. A small hotspot region is preferred in order to ensure that the context (neighborhood) is captured effectively. In this work, however, the primary objective of this step is to identify as

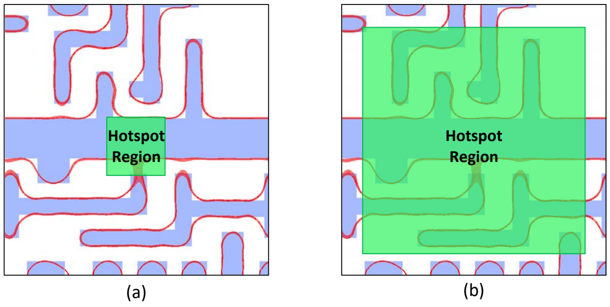

Fig. 4: Patterns with (a) a small hotspot region, (b) a large hotspot region

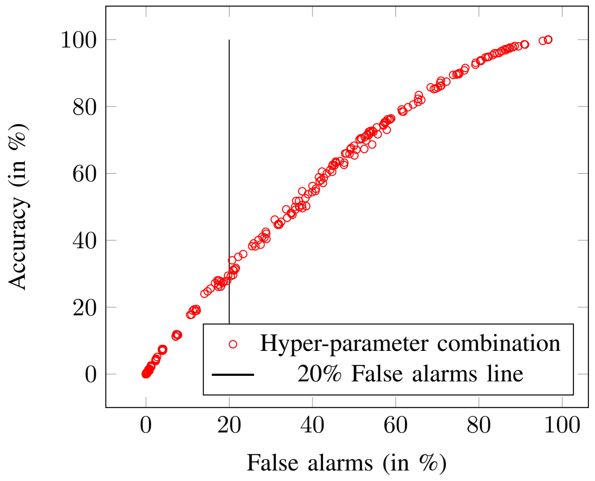

Fig. 5: Hyper-parameter tuning analysis

many hotspots as possible. Therefore, we use a large hotspot region which spans almost the entire area of the pattern, except the periphery, as shown in Figure 4b.

III. Experimental Results

We perform our experiments on a 45nm technology node [20]. An open-source circuit is placed and routed using the Nangate open cell library [21] to obtain baseline design patterns. The basic design rules from the same PDK are provided as inputs to VIPER to generate synthetic layout patterns. In the rest of the paper, we refer to patterns captured from the real design as 'Design patterns' and patterns obtained from VIPER as 'VIPER patterns'. The Design patterns were captured from the baseline design using a moving window scheme [11]. All patterns correspond to $Metal1$ and have their side dimensions equal to 8.5*$layer\_pitch$, which translates to 1105nm. All patterns used in this analysis are DRC clean, and their true labels (hotspots/non-hotspots) are determined through lithographic simulations. Co-ordinate transformation [11] is used for feature extraction. Principal Component Analysis (PCA) is used for dimensionality reduction and only the first 250 principal components are used in our experiments. Formulas used in this work are:

$$accuracy = \frac{hotspot\_hits}{total\_hotspots}$$

$$false\_alarms = \frac{false\_positives}{total\_patterns\_tested}$$

The main objectives of this work are: (a) to demonstrate hotspot detection early during technology development, and (b) to demonstrate TNSB hotspot detection. Corresponding results are presented in the following subsections.

A. Hotspot Detection During Early Technology Development

To demonstrate the effectiveness of the proposed methodology, we train a hotspot detection model solely using 40,000 VIPER patterns and test them using 40,000 design patterns. A Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel is used as a two-class classifier. To find the optimal hyper-parameters of the SVM, we used the grid search method [22] along with k-fold cross validation [23]. We varied the hyper-parameters across a wide range (546 combinations) and performed 3-fold cross validation for every hyper-parameter combination. The result of this analysis is shown in Figure 5. Each data point in this plot indicates the accuracy and false alarm rates obtained for a certain hyper-parameter combination. The data points along the pareto-front represent optimal hyper-parameter combinations and the user can choose any one of them depending on the desired region of operation.

TABLE I: Early hotspot detection results

Training dataset	Testing dataset	Accuracy	False alarms
VIPER patterns	Design patterns	54.83%	19.16%

In case of early hotspot detection models, our priority is to identify as many hotspots as possible. Therefore, during training, we choose the hyper-parameters which provide higher accuracy rates and accept a slight increase in false alarms. In this work, we train the hotspot detection model using the hyper-parameter combination which shows about 20% false-alarm rate, and observe the model performance on the actual testing dataset. The results are shown in Table I. We observe that the early hotspot detection model detects about 55% of the hotspots found in real designs. This is a significant result considering that a very small-sized dataset, solely consisting of synthetic patterns, was used for training.

B. TNSB Hotspot Detection

In this analysis, we consider an initial dataset of 5,000 design patterns for training. This dataset acts as a proxy to a known database built by a foundry over time. We train a model using this dataset, test it on a testing dataset containing 40,000 design patterns, and use the results as our baseline. Prior to such analysis, however, we need to ensure that our testing dataset contains TNSB test hotspots. Therefore, to visualize the distribution of testing patterns w.r.t. the training patterns, we performed Principal Component Analysis (PCA) on the training dataset and projected the testing dataset onto the same space. We plot the first three principal components in Figure 6. For brevity, the test non-hotspots are not shown in the plot. From Figure 6, we observe that all the test hotspots align well with the training hotspots, thereby, indicating that this testing dataset does not contain TNSB hotspots.

In order to create a test case which consists of TNSB hotspots (as illustrated in Figure 3a), we trim down the original training dataset by removing the training hotspots whose first principal component value is greater than or equal to -2. Then, we consider all the test hotspots which are located far away from the training hotspots (specifically, test hotspots whose first principal component value is greater than +2) as TNSB hotspots. The PCA plot of the new dataset is shown in Figure

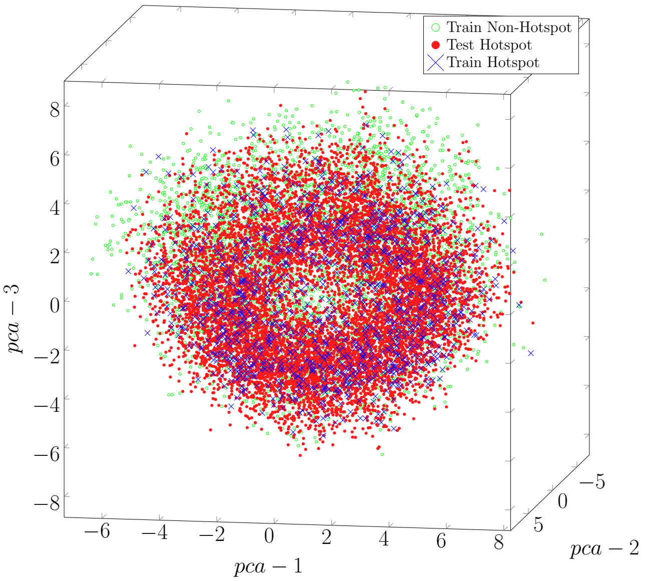

Fig. 6: Distribution of testing hotspots w.r.t. training dataset

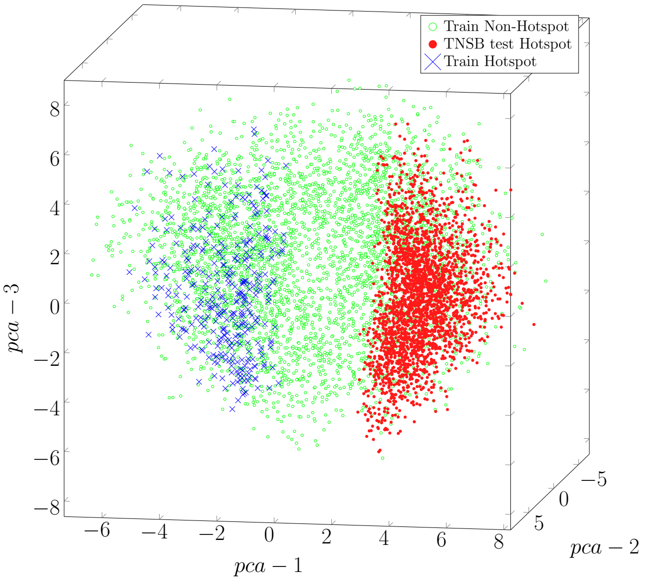

Fig. 7: Distribution of TNSB hotspots w.r.t. trimmed down training dataset

7. This plot shows that, similar to the illustration in Figure 3a, the test hotspots identified as TNSB hotspots are located far away from all the training patterns. Therefore, in the rest of the paper, we use this training dataset as our baseline training dataset and we refer to the model trained using this dataset as the *baseline model*. The new baseline training dataset consists of 4,131 training patterns, of which 340 are hotspots. The complete testing dataset consists of 30,413 non-hotspots and 9,587 hotspots, of which 2,477 are TNSB hotspots.

We train a classifier using the baseline dataset and test it on the complete testing dataset which includes TNSB hotspots. The results are shown in the Table II. We observe that the baseline model provides about 40% accuracy. When tested on only the TNSB test hotspots, however, we find that the model accuracy drops to 0%.

TABLE II: TNSB hotspot detection results

Training dataset	**Testing dataset**	**Accuracy**	**False alarms**
Baseline dataset	Complete testing dataset (including TNSB hotspots)	39.86%	16.30%
	TNSB hotspots only	**0%**	Not applicable
Enhanced dataset	Complete testing dataset (including TNSB hotspots)	56.25%	15.48%
	TNSB hotspots only	**21.52%**	Not applicable

According to the proposed methodology, described in Section II-B, we enhance the baseline training dataset by adding synthetic patterns. The baseline training dataset is highly imbalanced due to its large number of non-hotspots and few hotspots. Therefore, of the 40,000 VIPER patterns, we only use the hotspots (1,346) for enhancement. We train a hotspot detection model using the enhanced dataset. Hereafter, we refer to this model as the *enhanced model*. Results from the enhanced model are also shown in Table II. We observe that the enhanced model has an accuracy rate of about 56% while maintaining a similar false alarm rate as that of the baseline model. The enhanced model shows a modest improvement in accuracy when tested on the complete testing dataset. When tested on only TNSB hotspots, however, we find that the enhanced model detects about 22% of TNSB hotspots (533 out of 2477 TNSB hotspots), which is a significant improvement over the baseline results.

For both enhanced and the baseline models, we used an SVM with an RBF kernel for classification. Hyper-parameter selection was performed through grid-search and 3-fold cross validation on their respective training datasets, using the same procedure as detailed in the previous subsection.

IV. Conclusion

In this work, we have proposed methodologies to perform early hotspot detection and TNSB hotspot detection. We have demonstrated about 55% accuracy in the case of early hotspot detection and about 22% accuracy in TNSB hotspot detection, which is a significant improvement over the baseline. While plenty of room for improvement remains, this is the first quantitative demonstration of both early hotspot detection and TNSB hotspot detection, which we hope that will motivate the research community to look more diligently into this problem.

Acknowledgment

This research has been partially supported by the Semiconductor Research Corporation (SRC) through task 2709.001.

References

[1] V. Dai, L. Capodieci, J. Yang and N. Rodriguez, "Developing DRC plus rules through 2D pattern extraction and clustering techniques," *SPIE Advanced Lithography*, pp. 727517 – 727517–10, 2009.

[2] H. Yao, S. Sinha, C. Chiang, X. Hong and Y. Cai, "Efficient process-hotspot detection using range pattern matching," in *IEEE/ACM International Conference on Computer-aided Design (ICCAD)*, 2006, pp. 625–632.

[3] W. Y. Wen, J. C. Li, S. Y. Lin, J. Y. Chen and S. C. Chang, "A fuzzy-matching model with grid reduction for lithography hotspot detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 11, pp. 1671–1680, 2014.

[4] N. Nagase, K. Suzuki, K. Takahashi, M. Minemura, S. Yamauchi, and T. Okada, "Study of hot spot detection using neural networks judgment," in *Photomask and Next-Generation Lithography Mask Technology XIV*, 2007, vol. 6607.

[5] D. Ding, J. A. Torres and D. Z. Pan, "High performance lithography hotspot detection with successively refined pattern identifications and machine learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 11, pp. 1621–1634, 2011.

[6] D. Ding, X. Wu, J. Ghosh and D. Z. Pan, "Machine learning based lithographic hotspot detection with critical-feature extraction and classification," in *IEEE International Conference on IC Design and Technology (ICICDT)*, 2009, pp. 219–222.

[7] D. Ding, B. Yu, J. Ghosh and D. Z. Pan, "EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation," in *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2012, pp. 263–270.

[8] H. Zhang, B. Yu and E. F. Y. Young, "Enabling online learning in lithography hotspot detection with information-theoretic feature optimization," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, pp. 1–8.

[9] K. Madkour, S. Mohamed, D. Tantawy and M. Anis, "Hotspot detection using machine learning," in *IEEE International Symposium on Quality Electronic Design (ISQED)*, 2016, pp. 405–409.

[10] H. Yang, J. Su, Y. Zou, B. Yu, and E. F. Y. Young, "Layout hotspot detection with feature tensor generation and deep biased learning," in *Design Automation Conference (DAC)*, 2017, p. 62.

[11] G. R. Reddy, C. Xanthopoulos, and Y. Makris, "Enhanced hotspot detection through synthetic pattern generation and design of experiments," in *IEEE VLSI Test Symposium (VTS)*, 2018, pp. 1–6.

[12] G. R. Reddy, K. Madkour, and Y. Makris, "Machine learning-based hotspot detection: Fallacies, pitfalls and marching orders," in *IEEE/ACM International Conference on Computer-aided Design (ICCAD)*, 2019.

[13] V. Luong, "EUV lithography coming to your local IC manufacturer! soon$^{\mathrm{TM}}$," Presentation at Arenberg Youngster Seminar, Leuven, 2018.

[14] H. Li, E. Zou, R. Lee, S. Hong, S. Liu, J. Wang, C. Du, R. Zhang, K. Madkour, H. Ali, D. Hsu, A. Kabeel, W. El-Manhawy, and J. Kwan, "Design space exploration for early identification of yield limiting patterns," in *Design-Process-Technology Co-optimization for Manufacturability X*, 2016, vol. 9781.

[15] G. R. Reddy, J. Wallner, K. Babich, and Y. Makris, "Pattern matching rule ranking through design of experiments and silicon validation," in *ASM International Symposium for Test and Failure Analysis*, 2018, pp. 443–448.

[16] J.-W. Jeon, J. Song, J.-L. Kim, S. Park, S.-H. Yang, S. Lee, H. Kang, K. Madkour, W. ElManhawy, S. Lee, and J. Kwan, "Early stage hot spot analysis through standard cell base random pattern generation," in *Design-Process-Technology Co-optimization for Manufacturability XI*, 2017, vol. 10148.

[17] Y. Chen, T. Gai, X. Su, Y. Wei, Y. Su, and T. Ye, "Hybrid hotspot library building based on optical and geometry analysis at early stage for new node development," in *Design-Process-Technology Co-optimization for Manufacturability XII*, 2018, vol. 10588.

[18] A. Hamouda, M. Bahnas, D. Schumacher, I. Graur, A. Chen, K. Madkour, H. Ali, J. Meiring, N. Lafferty, and C. McGinty, "Enhanced OPC recipe coverage and early hotspot detection through automated layout generation and analysis," in *Optical Microlithography XXX*, 2017, vol. 10147.

[19] G. R. Reddy, M.-M. Bidmeshki, and Y. Makris, "VIPER: Versatile and intuitive pattern generator for early design space exploration," in *International Test Conference (ITC)*, 2019.

[20] FreePDK45, ," https://www.eda.ncsu.edu/wiki/FreePDK, 2018, [Online; accessed 1-Nov-2018].

[21] Nangate OCL, ," https://www.nangate.com/?page_id=22, 2017, [Online; accessed 1-May-2017].

[22] I. Syarif, A. Prugel-Bennett, and G. Wills, "SVM parameter optimization using grid search and genetic algorithm to improve classification performance," *Telkomnika*, vol. 14, no. 4, pp. 1502, 2016.

[23] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, 1995, vol. 14, pp. 1137–1145.