Bilinear Lithography Hotspot Detection [*]

Hang Zhang, Fengyuan Zhu, Haocheng Li, Evangeline F. Y. Young, and Bei Yu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin N.T., Hong Kong
{hzhang,fyzhu,hcli,fyyoung,byu}@cse.cuhk.edu.hk

ABSTRACT

Advanced semiconductor process technologies are producing various circuit layout patterns, and it is essential to detect and eliminate problematic ones, which are called lithography hotspots. These hotspots are formed due to light diffraction and interference, which induces complex intrinsic structures within the formation process. Though various machine learning based methods have been proposed for this problem, most of them cannot capture the intrinsic structure of each data. In this paper, we propose a novel feature extraction by representing each data sample in matrix form. We argue that this method can well preserve the intrinsic feature of each sample, leading to better performance. We then further propose a *bilinear lithography hotspot detector*, which can tackle data in matrix form directly to preserve the hidden structural correlations in the lithography process. Experimental results show that the proposed method outperforms state-of-the-art ones with remarkably large margin in both false alarms and runtime, with 98.16% detection accuracy.

1. INTRODUCTIONS

Today, we witness various design for manufacturing (DFM) technologies to tackle problems caused by shrinking feature device size. However, the existence of hotspots after DFM process remains to be a problem, and the issue of hotspot detection is important to ensure high manufacturability. Although full-chip lithography simulation can achieve very satisfactory performance, it is extremely computationally expensive. Thus, it is imperative to derive a fast and accurate hotspot detection method.

Besides full-chip lithography simulation, pattern matching (PM) [1–4] and machine learning (ML) [5–10] based methods are playing an increasingly important role in DFM due to their high performance in detecting hotspots. Particularly, ML based methods show their superiority of detecting unseen layout patterns, which are used more widely than PM based methods. These ML methods are mainly guided by supervised learning algorithms, such as Support Vector Machine (SVM) [8, 11], boosting classifier [9, 10, 12] and Deep Neural Network (DNN) [13, 14]. However, the performance of these methods is not satisfactory empirically.

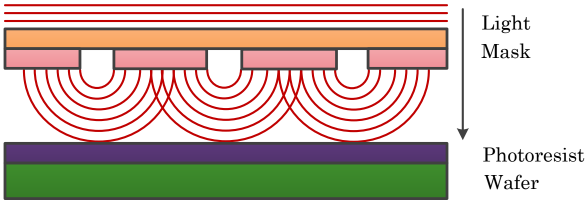

Figure 1: Phenomenon of light propagation, diffraction and interference in lithography process.

One reason is that those endeavors have not fully utilized the hidden information of circuit layout patterns. Conventional PM and ML based hotspot detectors are designed for layout pattern data in vector form. However, circuit layout patterns are intrinsically in matrix form, which can be represented as layout images in nanometer level. When using traditional PM and ML methods to process the layout patterns, we have to reshape data into vectors, resulting in destroying the hidden structural information, such as light propagation and interference between different layout features, and the spatial relationship of nearby pixels within a circuit layout image. Moreover, the dimensionality of each reshaped vector can be rather high and this may cause the problem of over-fitting in ML when the number of available data samples is limited. Therefore, the following two challenges should be well considered to develop more effective hotspot detection approaches: 1) the preservation of intrinsic structures for each layout pattern data in matrix form when training classifiers; 2) the over-fitting issue with limited number of high dimensional data samples.

Several methods [15–17] have been proposed to perform direct matrix classification and preserve the hidden structure of each data. The over-fitting issue can also be handled for these approaches with constrains on model parameters. The paper [15] uses the sum of k rank-one orthogonal matrices to model the classifier matrix, and the paper [16] assumes the rank of classifier matrix to be k. Both methods describe the correlation of the data in different ways, but they require the rank k to be pre-specified. The paper [17] proposes a spectral elastic net penalty into the model to determine the rank automatically, which intends to capture the grouping effect property of the data. However, it assigns same weights to all singular values when using nuclear norm penalty, resulting in the ignorance of important issue that hotspots are formed

[*] The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 14209214).

ISPD '17, March 19-22, 2017, Portland, OR, USA
© 2017 ACM. ISBN 978-1-4503-4696-2/17/03 . . . $15.00
DOI: http://dx.doi.org/10.1145/3036669.3036673

by the sum of light influence in the neighboring layout and larger singular values should be shrunk less to preserve the major influential components. Also, lithography process does not have the grouping effects as stated in [17], because layout polygons from different distances to one point may produce various light intensities to that point due to light diffraction. More importantly, the labeling process of layout patterns consists of manually set parameters, which may introduce label noises into the data sample. Therefore, it is important to derive a new model to capture such hidden structural information induced by the lithography process.

To tackle the above problems, we propose a bilinear lithography hotspot detection (BL-HSD) framework, taking the advantages of both hinge loss [18] and weighted nuclear norm [19]. In lithography manufacturing, the wave-length of the current lithography technique (usually $193nm$) is much larger than the feature size of the layout polygons (we use $28nm$ and $32nm$ for evaluation). Therefore, light diffraction and interference, as shown in Fig. 1, will occur under such conditions and cause problematic layout patterns. Since hotspot is formed by light passing through the polygon masks and causing interference with each other, intuitively, there exist structural correlations among the contributions of each layout feature to hotspot formation, and we aim at developing a learning model that can find out these correlations. Based on the above issues of exsiting models and the mechanism of lithography process, we adopt weighted nuclear norm penalty into our model. In addition, as mentioned, the procedure of labeling hotspot and non-hotspot is complicated, and there may exist some noises on labels; thus, we use hinge loss for its robustness and sparseness.

Another important issue for ML based hotspot detection is the feature extraction procedure. Current approaches [4, 9, 10, 20] for hospot detection are all designed to extract vector-form features, which cannot capture the hidden structural correlations induced by the lithography process. Although the paper [10] proposed an maximal circular mutual information (MCMI) scheme for feature optimization with a specifically designed Naive Bayes classifier to capture such correlations, it can only preserve the local correlations instead of global correlations, because it assumes that the sampling points on the same circle are dependent but different circles are independent. To tackle this issue, we propose a simple matrix based concentric circle sampling (MCCS) method. This method extracts features in matrix form, which can preserve the hidden structural information among data and serve for the bilinear machine learning model. More importantly, with the simplicity of MCCS, through appropriate use of parallel programming techniques, we achieve high efficiency in the feature extraction procedure, resulting in an acceleration of the whole framework.

We also conduct extensive empirical experiments. Our proposed BL-HSD framework can outperform current state-of-the-art methods, and achieve satisfactory performance in accuracy, false alarms and runtime. The key contributions of our paper can be summarized as follows.

- A novel matrix based concentric circle sampling method for feature extraction is proposed.
- A novel bilinear machine learning model is constructed to solve hotspot detection problem, which is the first such model.
- Efficient proximal algorithms are derived for model training.
- The excess risk bound of our proposed bilinear model is theoretically analyzed.
- Only $18s$ is needed on average to perform the whole detection process.

The rest of this paper is organized as follows. In Section 2 and 3, we describe the notations for our model, metrics for evaluations, and problem formulations. In Section 4, we derive the bilinear machine learning model and its numerical solver. Section 6 presents the experimental results, followed by a conclusion in Section 7.

2. PRELIMINARIES

We first define two terminologies to quantify the performance of our proposed BL-HSD framework as follows.

Definition 1 (Accuracy). *The rate of correctly predicted hotspots among the set of actual hotspots.*

Definition 2 (False Alarm). *Non-hotspot that is incorrectly predicted.*

We then give notations for our optimization framework. We present the scalar values with lower case letters (e.g., x); vectors by bold lower case letters (e.g., $\mathbf{x}$); and matrix by bold upper case letters (e.g., $\mathbf{X}$). For a matrix $\mathbf{X} \in \mathbb{R}^{p\times q}$ of rank r where $r \leq \min(p, q)$, its (i, j)-entity is represented as $\mathbf{X}_{i,j}$. $\mathrm{tr}(\cdot)$ denotes the trace of a matrix, $(a)_+ = \max(0, a)$ and $\langle A, B\rangle = \sum_{i,j} A_{i,j} \cdot B_{i,j}$ is the element-wise multiplication for matrices . We further set $||\mathbf{X}||_F$ and $||\mathbf{X}||_*$ as the Frobenius norm and nuclear norm of a matrix $\mathbf{X}$, respectively, where $||\mathbf{X}||_F = \sqrt{\sum_{i,j} X_{i,j}^2}$ and $||\mathbf{X}||_* = \sum_{i=1}^n \sigma_i$ (σ_i is the i^{th} singular value for matrix $\mathbf{X}$). Weighted nuclear norm is defined as $||\mathbf{X}||_{\mathcal{W},*} = \sum_i^n w_i\sigma_i$, where $\mathcal{W} = [w_1, w_2, ..., w_n]$ and w_i is a non-negative weight for σ_i. For a given norm $||\cdot||$ on $\mathbb{R}^n$, the dual norm, denoted $||\cdot||^*$ is the function from $\mathbb{R}^n$ to $\mathbb{R}$ with values $||\mathbf{y}|| = \sup_{\mathbf{x}} \mathbf{x}^\top\mathbf{y}$, s.t. $||\mathbf{x}|| \leq 1$. The dual norm of the nuclear norm and the weighted nuclear norm are denoted as $||\mathbf{X}||_*^*$ and $||\mathbf{X}||_{\mathcal{W},*}^*$.

3. LAYOUT FEATURE EXTRACTION

In this section, we will tackle the issue of layout feature extraction, which plays an important role in keeping layout pattern information. Only with well preserved pattern information can machine learning model get a good performance. Current feature extraction methods [4, 9, 10, 20] encode layout clips into feature vectors that contains geometrical information, such as density, shape and polygon topology. However, none of the existing methods takes the physical essentials, such as light propagation, diffraction, and interference of the lithography process into consideration, resulting in loss of pattern information. Empirically, as shown in Fig. 1, problematic layout patterns are caused by light passing through the photo masks, and these diffracted lights interfere with each other. Therefore, it is imperative to derive a feature extraction method that can efficiently capture the physical phenomenon of light propagation and interference.

Recently, besides the hotspot detection problem, machine learning methods have also been used in optical proximity correction (OPC) works [21, 22] and show reasonably good performance. The paper [21] proposes a concentric square sampling (CSS) method to model OPC, but this method only samples squares on layout clips, which ignores the information of light propagation. The paper [22] addresses this issue by

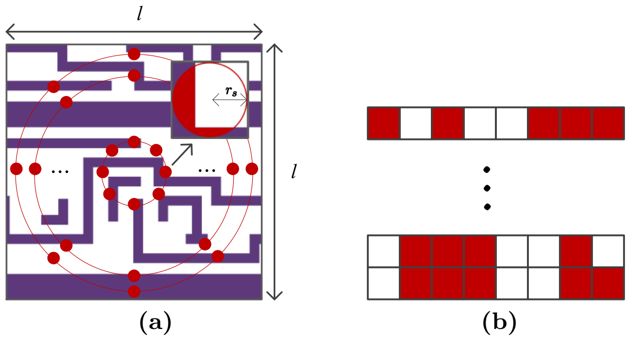

Figure 2: Examples of CCAS and MCCS feature. (a) Illustration of concentric circle area sampling. (b) The feature matrix of our proposed matrix based concentric circle sampling.

introducing the concentric circle area sampling (CCAS) methods. Although CCAS considers light propagation and shows its superiority in OPC regression work compared to CSS, it still ignores the important information of light interference. Inspired from the fact that OPC work involves a lithography process similar to hotspot detection, the paper [10] proposes an MCMI scheme to perform circle selection with reasonable good result. However, this feature extraction method [10] can only preserve light interference in a local manner, and ignores important global information, and it also required expensive time cost to perform the circle selection procedure. To tackle the above issue, we propose a novel layout feature extraction method, matrix based concentric circle sampling (MCCS), to preserve the structural correlations and serve for the bilinear machine learning model. In the following two sub-sections, we first review the process of CCAS and then we present our MCCS.

3.1 Concentric Circles with Area Sampling

CCAS [22] is proposed to train a Hierarchical Bayesian Model (HBM) for OPC regression. It extracts sub-sampled pixel values on concentric circles and forms a feature vector $\mathbf{x}$. The basic concept of CCAS is shown in Fig. 2(a), where the clip sample is taken from the ICCAD benchmark suite [23]. Parameters of the CCAS feature extraction method consist of the total size of the clip l (indicated in Fig. 2(a)), r_{in} and r_s. Each circle of CCAS has 8 sampling points, and each point stands for a circular sampling area (see Fig. 2(a)) with radius r_s (we would get the sum of the red area). r_{in} is the the sampling density controlling parameter, where we sample circles up to radius r_{in} in increments of 10 and further up to radius $\frac{l}{2}$ using increments of 20. Under the conditions that the clip size $l = 1200nm$, and $r_{in} = 60nm$, the dimension of each feature vector $\mathbf{x}$ should be 265 ($265 = 1+(6+27)\times 8$). Although CCAS can correctly preserve layout pattern information that affects propagation of diffracted light from a mask pattern, it is not able to preserve the most important information in the lithography process, light interference, as CCAS destroys the structural correlation by forming the feature in vector form.

3.2 Matrix based Concentric Circle Sampling

We then describe our proposed MCCS feature extraction method. Intuitively, the light intensity induced by each sampling point on a certain circle should have proportional influence to the hotspot formation. Therefore, we concatenate sampling point values within one circle, putting them into a vector forming one row of our feature matrix, as shown in Fig. 2(b). The center point of a clip is the place where hotspot forms, and statistically the layout feature at this point does not contribute to the hotspot formation. Hence, we will ignore the value of the center point in MCCS. From inner to outer circles, values on one circle form one row of the feature matrix from bottom to top and one example is shown in Fig. 2(a) and Fig. 2(b). We also denote the vector form of MCCS feature as vector based concentric circle sampling (VCCS), where each instance is represented as a vector instead of a matrix. Besides the parameters in CCAS, l, r_{in}, r_s, we add a new parameter n_p, which is the number of sampling points on a circle. For some complicated layout designs, it would be more accurate to use more sampling points to represent the layout patterns.

Under the conditions that $l = 1200nm$, $r_{in} = 60nm$ and $n_p = 16$, the dimension of each feature matrix $\mathbf{X}$ is 33×16 ($33 = 6+27$). Each row of the feature matrix consists of sampling point values on a circle, which affects the phenomenon of light propagation, and each entry in the matrix is a continuous number ranging from zero to the maximal value of the sampling area (e.g, when $r_s = 2$, the maximal value is 4π). With the data in matrix form, our bilinear learning model (described in Section 4) can capture the correlations among these rows and columns, which affects the phenomenon of light interference. Although MCCS is still an approximation of the original layout pattern, we achieve satisfactory performance by utilizing the information of hidden structural correlations.

4. BILINEAR CLASSIFIER

In this section, we introduce our proposed bilinear classifier to address hotspot detection problem. Then an efficient Alternating Directional Method of Multipliers (ADMM) algorithm will be proposed for model training.

4.1 Model

In lithography hotspot detection with MCCS feature extraction, it is essential to efficiently handle feature matrices while preserving the hidden topological structures of the layout patterns. However, using traditional linear or non-linear classification methods directly requires to reorganize matrices into vectors, which may destroy the hidden structures. To tackle this issue, we propose a bilinear classifier that can process the data matrices directly with their hidden structures preserved. The procedure of labeling layout patterns consists of multiple lithography processes, which may bring noises into pattern labels. To tackle this issue, an intuitive idea is to model the loss of each instance within a margin, where loss means the training error. Therefore, we consider the hinge loss for our model fitting due to its robustness and sparseness, and the loss for each instance $\mathbf{X}_i$ is defined as follows

$$h_i(\mathbf{W}, b) = \{1 - y_i[\mathrm{tr}(\mathbf{W}^\top \mathbf{X}_i) + b]\}_+, \tag{1}$$

where $\mathbf{W}$ is the classifier matrix, b is the bias, $\mathbf{X}_i$ is the feature matrix of the i^{th} instance and y_i is the label of the i^{th} instance ($\mathbf{W} \in \mathbb{R}^{p\times q}$, $\mathbf{X}_i \in \mathbb{R}^{p\times q}$).

When learning the classifier matrix, another important issue is to take the structural correlation among each data matrix into consideration. Clearly, methods like SVM [24] cannot handle this issue though the hinge loss is also adopted, as it can only handle data in vector form. The Bilinear SVM proposed in [16] factorized a classifier matrix into two low rank matrices, but it is needed to indicate the rank before the optimization. Recently, machine learning models apply nuclear norm for low-rank modeling and obtain reasonable good result. The paper [17] considers the grouping effects of features and treat each singular value of the classifier matrix equally. However, because of light diffraction, layout features from different places within a clip may impose different light intensities to the clip center and thus the grouping effect considered in [17] may not be consistent with our lithography process.

Another important issue is that, singular values corresponding to the subspaces of the classifier matrix have clear physical meanings in certain applications. In our hotspot detection problem, since the mask is exposed to parallel lights during the lithography process, intuitively, layout feature from different places contributes inequally to the hotspot formation. As a result, it would be better to treat each singular value differently and shrink more the smaller singular values, which contributes less to the hotspot formation. When learning the classifier matrix, we apply weighted nuclear norm regularization and assign different weights to different singular values to keep the physical property.

The work [19] has discussed different weighting schemes for the weighted nuclear norm, such as the weights in non-ascending order, in arbitrary order and in non-descending order. The non-ascending order can ensure the convexity of the optimization problem empirically. However, it does not help to preserve the physical property for this problem. Therefore, for hotspot detection, we apply weights in the non-descending order meaning that we will shrink less those larger singular values of the classifier matrix in weighted nuclear norm minimization. Although weights in a non-descending order does not guarantee the convexity of the objective function, we still can get a fixed point solution in an analytical form, which will be discussed later.

The optimization problem for our proposed bilinear classifier is defined as follows. It is specifically designed for this hotspot detection problem and is different from all previous bilinear classifiers to the best of our knowledge,

$$\arg\min_{\mathbf{W},b} \ \lambda||\mathbf{W}||_{\mathcal{W},*} + C\sum_{i}^{n}\{1 - y_i[\text{tr}(\mathbf{W}^\top\mathbf{X}_i) + b]\}_+. \quad (2)$$

The first term in Eq. (2) represents a weighted nuclear norm penalty, which considers the importances of different singular values of $\mathbf{W}$ and is the key term to capture the inherent structure of the lithography layout patterns. The second term denotes the hinge loss.

4.2 Solver

Since the objective function in Eq. (2) contains both hinge loss and weighted nuclear norm, traditional methods such as the Nesterov method used in [25] is not applicable here. However, observing the structure of our objective function, we derive an efficient learning algorithm based on ADMM [26] with the restart rule [27] for numerical optimization, which can achieve relatively faster training speed compared to other machine learning methods used in hotspot detection. The optimization problem defined in Eq. (2) can be equivalently written as follows,

$$\begin{aligned}\arg\min_{\mathbf{W},b,\mathbf{S}} \ & \lambda||\mathbf{S}||_{\mathcal{W},*} + C\sum_{i}^{n}\{1 - y_i[\text{tr}(\mathbf{W}^\top\mathbf{X}_i) + b]\}_+, \quad (3)\\ \text{s.t.} \ & \mathbf{S} - \mathbf{W} = 0,\end{aligned}$$

In this way, the original optimization problem is split into two sub-problems with respect to $\{\mathbf{W}, b\}$ and the auxiliary variable $\mathbf{S}$. Then we apply Augmented Lagrangian Multiplier to develop an efficient ADMM method as follows:

$$\begin{aligned}L(\mathbf{W}, b, \mathbf{S}, \mathbf{\Lambda}) =& \lambda||\mathbf{S}||_{\mathcal{W},*} + C\sum_{i}^{n}\{1 - y_i[\text{tr}(\mathbf{W}^\top\mathbf{X}_i) + b]\}_+\\ &+ \text{tr}[\mathbf{\Lambda}^\top(\mathbf{S} - \mathbf{W})] + \frac{\rho}{2}||\mathbf{S} - \mathbf{W}||_F^2, \quad (4)\end{aligned}$$

where $\rho > 0$ is a coefficient parameter and $\mathbf{\Lambda}$ is a Lagrangian multiplier matrix.

The algorithm flow of using ADMM to solve the problem is summarized in Algorithm 1. Take $\mathbf{W}$ as an example, in the k^{th} iteration, variable obtained before the restart rule is denoted as $\mathbf{W}^{(k)}$ and variable obtained in the restart rule is denoted as $\widehat{\mathbf{W}}^{(k)}$. The key steps of Algorithm 1 are the computations of $\mathbf{S}^{(k)}$ and $(\mathbf{W}^{(k)}, b^{(k)})$, we will derive them in the coming sections.

4.2.1 *Weighted Nuclear Norm Minimization*

Following the work [19], we discuss the weighted nuclear norm minimization. Without loss of generality, the general weighted nuclear norm minimization problem can be written as follows:

$$\min_{\mathbf{X}} \frac{1}{2}||\mathbf{Y} - \mathbf{X}||_F^2 + \lambda||\mathbf{X}||_{\mathcal{W},*}. \quad (5)$$

where $\mathbf{X}$ and $\mathbf{Y}$ are matrices, $\mathcal{W}$ is the weight vector.

The analytical solution for weights in non-descending order [19] is

$$\hat{\mathbf{X}} = \mathbf{U}\mathcal{D}_{\mathcal{W},\lambda}(\mathbf{\Sigma})\mathbf{V}^\top, \quad (6)$$

where $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is the Singular Value Decomposition (SVD) of $\mathbf{Y}$, and $\mathcal{D}_{\mathcal{W},\lambda}$ is the generalized soft-thresholding operator with weight vector $\mathcal{W}$ and $\mathcal{D}_{\mathcal{W},\lambda}(\mathbf{\Sigma})$ is a diagonal matrix with

$$\mathcal{D}_{\mathcal{W},\lambda}(\mathbf{\Sigma})_{ii} = \max(\mathbf{\Sigma}_{ii} - \lambda w_i, 0). \quad (7)$$

4.2.2 *Optimization for Auxiliary Variable*

We first derive the optimization method for solving the auxiliary variable $\mathbf{S}$. The first subproblem for solving $\mathbf{S}$ in Eq. (4) can be equivalently written as follows

$$\arg\min_{\mathbf{S}} \ \lambda||\mathbf{S}||_{\mathcal{W},*} + \text{tr}(\widehat{\mathbf{\Lambda}}^{(k)\top}\mathbf{S}) + \frac{\rho}{2}||\mathbf{W}^{(k)} - \mathbf{S}||_F^2. \quad (8)$$

Then we can get $\mathbf{S}^{(k)}$ in the k^{th} iteration by solving the problem in Eq. (8). We can get a equation with structure similar to Eq. (5) by simple equation transformation, which is shown as follows. The optimizatiopn problem Eq. 8 is equivalent to

$$\arg\min_{\mathbf{S}} \ \lambda||\mathbf{S}||_{\mathcal{W},*} + \frac{\rho}{2}||(\mathbf{W}^{(k)} - \mathbf{S} - \frac{\mathbf{\Lambda}^{(k)}}{\rho})||_F^2, \quad (9)$$

which can be further transformed into

$$\arg\min_{\mathbf{S}} \ \lambda||\rho\mathbf{S}||_{\mathcal{W},*} + \frac{1}{2}||(\rho\mathbf{W}^{(k)} - \mathbf{\Lambda}^{(k)}) - \rho\mathbf{S}||_F^2. \quad (10)$$

Thus, we can get the fixed point solution of Eq. (10) by applying Eq. (6), which is

$$\mathbf{S}^{(k)} = \mathbf{U}\mathcal{D}_{\mathcal{W},\lambda}(\rho\mathbf{W}^{(k)} - \mathbf{\Lambda}^{(k)})\mathbf{V}^\top, \quad (11)$$

where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \rho\mathbf{W}^{(k)} - \mathbf{\Lambda}^{(k)}$.

4.2.3 *Optimization for Classifier Matrix and Bias*

We first derive the subproblem for solving $(\mathbf{W}^{(k)}, b^{(k)})$ as follows.

$$\begin{aligned}\arg\min_{\mathbf{W},b}\ & C\sum_{i}^{n}\{1 - y_i[\mathrm{tr}(\mathbf{W}^\top\mathbf{X}_i) + b]\}_+ \\ & + \mathrm{tr}[\mathbf{\Lambda}^\top(\mathbf{S} - \mathbf{W})] + \frac{\rho}{2}||\mathbf{S} - \mathbf{W}||_F^2,\end{aligned} \tag{12}$$

Following the derivation in the work [17], the optimal values of $(\mathbf{W}^{(k)}, b^{(k)})$ are:

$$\begin{aligned}\mathbf{W}^* &= \frac{1}{\rho}(\sum_{i=1}^{N}\alpha_i^* y_i \mathbf{X}_i + \mathbf{\Lambda} + \rho\mathbf{S}) \\ b^* &= \frac{1}{|\mathcal{I}^*|}\sum_{i\in\mathcal{I}^*}\{y_i - \mathrm{tr}[(\mathbf{W}^*)^\top\mathbf{X}_i]\},\end{aligned} \tag{13}$$

where $\mathcal{I}^* = \{i : 0 < \alpha_i^* < C\}$, and $\alpha^* \in \mathbb{R}^n$ is the solution of the following box constraint quadratic programming problem.

$$\begin{aligned}\arg\min_{\alpha}\ & \frac{1}{2}\alpha^\top\mathbf{K}\alpha - \mathbf{q}^\top\alpha, \\ s.t\ & \mathbf{0} \le \alpha \le C\mathbf{1}_n, \\ & \sum_{i=1}^{n}\alpha_i y_i = 0.\end{aligned} \tag{14}$$

Here $\mathbf{K} \in \mathbb{R}^{n\times n}$ and $\mathbf{q} \in \mathbb{R}^n$ are coefficient matrix and coefficient vector for variable α; specifically,

$$\begin{aligned}K_{ij} &= y_i y_j \frac{\mathrm{tr}(\mathbf{X}_i^\top\mathbf{X}_j)}{\rho}, \\ q_i &= 1 - \frac{\mathrm{tr}[(\mathbf{\Lambda} + \rho\mathbf{S})^\top\mathbf{X}_i]}{\rho}.\end{aligned}$$

Several methods can be used to solve the optimization problem in Eq. (14), such as the sequential minimization optimization algorithm [28, 29]. With the derivation of above problems and optimization algorithms, we can update $(\mathbf{W}^{(k)}, b^{(k)})$.

4.2.4 *Summary*

We have presented the key steps for solving the problem in Eq. (4), where the classifier parameters $(\mathbf{W}, b)$ and the auxiliary variable $\mathbf{S}$ are solved in an iterative manner. In addition, we also need to update the Lagrangian parameter $\mathbf{\Lambda}$. Here we update $\mathbf{\Lambda}$ in a single gradient step as follows.

$$\mathbf{\Lambda} = \widehat{\mathbf{\Lambda}}^{(k)} - \rho(\mathbf{W}^{(k)} - \mathbf{S}^{(k)}). \tag{15}$$

Then we summarize the whole flow in Algorithm 1, where ADMM [26] with the restart rule [27] is applied here.

Algorithm 1 ADMM for problem in Eq. (4)

```
Initialize $\mathbf{S}^{(-1)} = \widehat{\mathbf{S}}^{(0)} \in \mathbb{R}^{p\times q},\ \mathbf{\Lambda}^{(-1)} = \widehat{\mathbf{\Lambda}} \in \mathbb{R}^{p\times q},\ \rho > 0,\ t^{(1)} = 1,\ \eta \in (0,1)$.
for $k = 0, 1, 2, 3...$ do
    $(\mathbf{W}^{(k)}, b^{(k)}) = \arg\min_{\mathbf{W},b} C\sum_i^n\{1 - y_i[\mathrm{tr}(\mathbf{W}^\top\mathbf{X}_i) + b]\}_+ + \mathrm{tr}[\widehat{\mathbf{\Lambda}}^{(k)\top}(\widehat{\mathbf{S}}^{(k)} - \mathbf{W})] + \frac{\rho}{2}||\widehat{\mathbf{S}}^{(k)} - \mathbf{W}||_F^2$
    $\mathbf{S}^{(k)} = \arg\min_{\mathbf{S}}\lambda||\mathbf{S}||_{\mathcal{W},*} + \mathrm{tr}(\widehat{\mathbf{\Lambda}}^{(k)\top}\mathbf{S}) + \frac{\rho}{2}||\mathbf{W}^{(k)} - \mathbf{S}||_F^2$
    $\mathbf{\Lambda}^{(k)} = \widehat{\mathbf{\Lambda}}^{(k)} - \rho(\mathbf{W}^{(k)} - \mathbf{S}^{(k)})$
    $c^{(k)} = \rho^{-1}||\mathbf{\Lambda}^{(k)} - \widehat{\mathbf{\Lambda}}^{(k)}||_F^2 + \rho||\mathbf{S}^{(k)} - \widehat{\mathbf{S}}^{(k)}||_F^2$
    if $c^{(k)} < \eta c^{(k-1)}$ then
        $t^{(k+1)} = \frac{1 + \sqrt{1 + 4t^{(k)^2}}}{2}$
        $\widehat{\mathbf{S}}^{(k+1)} = \mathbf{S}^{(k)} + \frac{t^{(k)-1}}{t^{(k+1)}}(\mathbf{S}^{(k)} - \mathbf{S}^{(k)-1})$
        $\widehat{\mathbf{\Lambda}}^{(k+1)} = \mathbf{\Lambda}^{(k)} + \frac{t^{(k)-1}}{t^{(k+1)}}(\mathbf{\Lambda}^{(k)} - \mathbf{\Lambda}^{(k)-1})$
    else
        $t^{(k+1)} = 1$
        $\widehat{\mathbf{S}}^{(k+1)} = \mathbf{S}^{(k-1)}$
        $\widehat{\mathbf{\Lambda}}^{(k+1)} = \mathbf{\Lambda}^{(k-1)}$
        $c^{(k)} = \eta^{-1}c^{(k-1)}$
    end if
end for
```

5. THEORETICAL JUSTIFICATIONS

Now we analyze the excess risk of the proposed bilinear classifier theoretically. Excess risk means the difference between the empirical risk (see Definition 3) and the expected risk (see Definition 4). In our theoretical analysis, we assume that each entry of a feature matrix follows unit Gaussian distribution.

The proposed optimization problem as defined in Eq. 2 can be reformulated as follows

$$\begin{aligned}\arg\min_{\mathbf{W},b}\ & \sum_{i=1}^{n} h(\mathbf{W}, b, \mathbf{X}_i, y_i), \\ \text{s.t. } & ||\mathbf{W}||_{\mathcal{W},*} \le B,\end{aligned} \tag{16}$$

where B is a constant value, and $h(\mathbf{W}, b, \mathbf{X}_i, y_i) = \{1 - y_i[\mathrm{tr}(\mathbf{W}^\top\mathbf{X}_i) + b]\}_+$ is the hinge loss function. The loss function can be easily rewritten as follows with respect to $\mathbf{W}$ given the relation between $\mathbf{W}$ and b as in Eq. 13

$$\hat{h}(\mathbf{W}, \mathbf{X}_i, y_i) = \{1 - y_i[\mathrm{tr}(\mathbf{W}^\top(\hat{\mathbf{X}}_i))]\} + c. \tag{17}$$

The loss function is L-Lipschitz continuous with c as a constant, and $\hat{\mathbf{X}}_i = \mathbf{X}_i - \frac{1}{n}\sum_{j=1}^{n}\mathbf{X}_j$, where the second term is the empirical expectation of each data and the value of each entry should tend to be zero when n is large, thus by removing empirical expectation of each data, we do not need to consider the bias b.

Before defining and proposing the excess risk bound, we first derive the dual norm of our weighted nuclear norm in Lemma 1, which will be used to derive inequalities later. To the best of our knowledge, this is the first time that the dual norm of the weighted nuclear norm is analyzed.

Lemma 1. *The dual norm of the weighted nuclear norm* $||\mathbf{W}||_{\mathcal{W},*}$ *is*

$$||\mathbf{W}||_{\mathcal{W},*}^* = \max_i \frac{1}{w_i}\mathbf{\Sigma}_{ii} \tag{18}$$

where $\mathbf{W} = \mathbf{U\Sigma V}^\top$ through SVD.

Proof. Following the definition of dual norm, we need to prove the following equation:

$$\sup_{||\mathbf{Q}||_{\mathcal{W},*}^*\le 1}\langle\mathbf{Q}, \mathbf{A}\rangle = \sup_{||\mathbf{Q}||_{\mathcal{W},*}^*\le 1}\mathrm{tr}(\mathbf{Q}^\top\mathbf{A}) = ||\mathbf{A}||_{\mathcal{W},*}. \tag{19}$$

To prove this, let $\mathbf{A} = \mathbf{U\Sigma V}^\top$ through the SVD, with $\mathbf{\Sigma}$ containing d singular values. Then, we can simply set $\mathbf{Q}$ with $\mathbf{Q} = \mathbf{UZV}^\top$ through SVD, where $\mathbf{Z}$ is diagonal matrix with $\mathbf{Z}_{ii} = w_i$, and the constrain that$||\mathbf{Q}||_{\mathcal{W},*}^* \le 1$ can be naturally satisfied. In this case, we have

$$\begin{aligned}\langle\mathbf{Q}, \mathbf{A}\rangle &= \langle\mathbf{UZV}^\top, \mathbf{U\Sigma V}^\top\rangle \\ &= \mathrm{tr}(\mathbf{Z\Sigma}) \\ &= ||\mathbf{A}||_{\mathcal{W},*}.\end{aligned} \tag{20}$$

In this way, we have $\sup_{||\mathbf{Q}||^*_{\mathcal{W},*}\leq 1} \langle \mathbf{Q}, \mathbf{A}\rangle \geq ||\mathbf{A}||_{\mathcal{W},*}$. Now, we further show the other direction of the inequality. Let $\mathbf{u}_i$ and $\mathbf{v}_i$ be the ith column vector respectively, we have

$$\begin{aligned}\sup_{||\mathbf{Q}||^*_{\mathcal{W},*}\leq 1} \langle \mathbf{Q}, \mathbf{A}\rangle &= \sup_{||\mathbf{Q}||^*_{\mathcal{W},*}\leq 1} \operatorname{tr}(\mathbf{Q}^\top \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top) \\ &= \sup_{||\mathbf{Q}||^*_{\mathcal{W},*}\leq 1} \sum_{i=1}^{d} w_i \mathbf{\Sigma}_{ii} \frac{1}{w_i}\mathbf{u}_i\mathbf{Q}\mathbf{v}_i^\top \\ &\leq \sup_{||\mathbf{Q}||^*_{\mathcal{W},*}\leq 1} \sum_{i=1}^{d} w_i \mathbf{\Sigma}_{ii} ||\mathbf{Q}||^*_{\mathcal{W},*} \\ &= ||\mathbf{A}||_{\mathcal{W},*}\end{aligned} \tag{21}$$

Combining both Eq. 20 and Eq. 21, we have proved Eq. 19. Then following the definition of dual norm, we finish the proof of Lemma 1. □

We further provide the definition of empirical and expected risks with respect to our loss function following [30].

Definition 3. *The standard form of empirical risk without bias term for loss function $\hat{h}(\mathbf{W}, \mathbf{X}_i, y_i)$ can be formulated as*

$$\hat{R}(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} \hat{h}(\mathbf{W}, \mathbf{X}_i, y_i). \tag{22}$$

Definition 4. *The standard form of expected risk without bias term for loss function $\hat{h}(\mathbf{W}, \mathbf{X}_i, y_i)$ can be formulated as*

$$R(\mathbf{W}) = \mathbb{E}_{(\mathbf{X}_i, y_i)\sim\mu}\hat{h}(\mathbf{W}, \mathbf{X}_i, y_i), \tag{23}$$

with $\mathbb{E}$ as the expectation operator, and μ as the probability distribution that each pair of $\{\mathbf{X}_i, y_i\}$ is sampled.

Here, we set $\mathbf{W}^o$ as the optimal solution with respect to the expected risk with

$$\mathbf{W}^o = \arg\min_{\mathbf{W}} R(\mathbf{W}), \qquad \text{s.t. } ||\mathbf{W}||_{\mathcal{W},*} \leq B, \tag{24}$$

and $\hat{\mathbf{W}}$ as the optimal solution with respect to the empirical risk with

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \hat{R}(\mathbf{W}), \qquad \text{s.t. } ||\mathbf{W}||_{\mathcal{W},*} \leq B. \tag{25}$$

Then we can provide the upper bound of the excess risk of our method in the following theorem.

Theorem 1: *With probability at least $1-\delta$, the excess risk of our method, for each data $\mathbf{X}_i \in \mathbb{R}^{d_1\times d_2}$, is bounded as*

$$\begin{aligned}R(\hat{\mathbf{W}}) - R(\mathbf{W}^o) \leq &\frac{2BL}{\sqrt{n}}\max_i(\frac{1}{w_i}) \\ &\cdot(\sqrt{d_1}+\sqrt{d_2}) + \sqrt{\frac{\ln(1/\delta)}{2n}}.\end{aligned} \tag{26}$$

Proof. Following the proof in the paper [30], we can first reformulate the excess risk with respect to $\mathbf{W}^o$ and $\hat{\mathbf{W}}$ as follows

$$\begin{aligned}R(\hat{\mathbf{W}}) - R(\mathbf{W}^o) = &[R(\hat{\mathbf{W}}) - \hat{R}(\hat{\mathbf{W}})] \\ &+ [\hat{R}(\hat{\mathbf{W}}) - \hat{R}(\mathbf{W}^o)] + [\hat{R}(\mathbf{W}^o) - R(\mathbf{W}^o)]\end{aligned} \tag{27}$$

Here, the second term is negative naturally. Following the *Hoeffding's inequality*, the third one can be bounded as $\sqrt{\ln(1/\delta)/2n}$, with probability $1-\delta/2$.

Different from the paper [30], our derivation of Eq. (30) and Eq. (31) follows the L-Lipschitz continuous property of the loss function and Lemma 1. For the first term, it is shown in [30] that

$$R(\hat{\mathbf{W}}) - \hat{R}(\hat{\mathbf{W}}) \leq \sup_{||\mathbf{W}||_{\mathcal{W},*}\leq B}[R(\mathbf{W}) - \hat{R}(\mathbf{W})]. \tag{28}$$

Further using the *McDiarmid's inequality*, we can obtain the *Rademacher complexity* with probability $1-\delta$, with

$$\mathcal{R} = \frac{2}{n}\mathbb{E}\sup_{||\mathbf{W}||_{\mathcal{W},*}\leq B}\sum_{i=1}^{n}\sigma_i\hat{h}(\mathbf{W}, \mathbf{X}_i, y_i), \tag{29}$$

where $\sigma_i \in \{-1, 1\}$ represents the *Rademacher variables.* Let $\hat{\mathbf{M}} = \sum_{i=1}^{n}\sigma_i\hat{\mathbf{X}}_i$, and following that our loss function is L-Lipschitz continuous, we can obtain the upper bound of $R(\hat{\mathbf{W}}) - \hat{R}(\hat{\mathbf{W}})$ as follows

$$\begin{aligned}R(\hat{\mathbf{W}}) - \hat{R}(\hat{\mathbf{W}}) &\leq \mathcal{R} \\ &\leq \frac{2L}{n}\mathbb{E}\sup_{||\mathbf{W}||_{\mathcal{W},*}\leq B}\sum_{i=1}^{n}\sigma_i\operatorname{tr}(\mathbf{W}\hat{\mathbf{X}}_\mathbf{i}) \\ &= \frac{2L}{n}\mathbb{E}\sup_{||\mathbf{W}||_{\mathcal{W},*}\leq B}\operatorname{tr}(\mathbf{W}\hat{\mathbf{M}}).\end{aligned} \tag{30}$$

Further applying the *Hölder's inequality* and Lemma 1, we have

$$\begin{aligned}R(\hat{\mathbf{W}}) - \hat{R}(\hat{\mathbf{W}}) &\leq \frac{2L}{n}\mathbb{E}\sup_{||\mathbf{W}||_{\mathcal{W},*}\leq B}||\hat{\mathbf{M}}||_{\mathcal{W},*}||\mathbf{W}||^*_{\mathcal{W},*} \\ &\leq \frac{2LB}{n}\mathbb{E}||\hat{\mathbf{M}}||^*_{\mathcal{W},*}.\end{aligned} \tag{31}$$

Since $\hat{\mathbf{M}}$ is the sum of random variables, it should tend to be normal distributed with the *Central Limit Theorem*, with variance equal to $\max_i(\frac{1}{w_i})\sqrt{n}$. Thus, following [30], with the *Gordan's theorem*, we have

$$\mathbb{E}||\hat{\mathbf{M}}||^*_{\mathcal{W},*} \leq \max_i(\frac{1}{w_i})\sqrt{n}(\sqrt{d_1}+\sqrt{d_2}). \tag{32}$$

Combining all the above together, we can obtain the upper bound of the excess risk with probability at least $1-\delta$ as follows

$$\begin{aligned}R(\hat{\mathbf{W}}) - R(\mathbf{W}^o) \leq &\frac{2BL}{\sqrt{n}}\max_i(\frac{1}{w_i}) \\ &\cdot(\sqrt{d_1}+\sqrt{d_2}) + \sqrt{\frac{\ln(1/\delta)}{2n}}.\end{aligned} \tag{33}$$

□

6. EXPERIMENTAL RESULTS

We implement our MCCS feature extraction method in the programming language `Python`, whose speed is further accelerated by `Cython`. In addition, we utilize the advantages of matrix calculation of `Matlab`, and implement our proposed bilinear classifier in `Matlab`. We executed the program on a machine with Quad Intel Xeon E7-4830 v2 CPUs and 1TB memory. Experiments are conducted on 5 industrial circuit layout designs, which consists of one 32*nm* and four 28*nm* circuit layouts. These circuit designs are released by [23], the details of which can be found in the paper [10]. In the experiments, all the coefficient parameters are selected via cross validation. The weight vector of weighted nuclear norm is set

Table 1: Comparisons with three classical methods

	VCCS-SVM			VCCS-Adaboost			DBF-Adaboost [9]			Ours			
	M-CPU(s)	Accuracy	FA#	M-CPU(s)	Accuracy	FA#	CPU(s)	Accuracy	FA#	CPU(s)	M-CPU(s)	Accuracy	FA#
Case 1	1.09	**100.00%**	**0**	1.37	99.55%	1	7.00	**100%**	**0**	**2.09**	**0.20**	**100.00%**	**0**
Case 2	1.81	94.78%	4	5.44	96.78%	**0**	351.00	98.60%	**0**	**10.70**	**0.33**	**99.40%**	**0**
Case 3	3.26	95.52%	94	4.73	97.62%	4	297.00	97.20%	**0**	**20.56**	**2.34**	**97.78%**	2
Case 4	1.74	80.23%	31	9.45	84.10%	**0**	170.00	87.01%	1	**8.09**	**0.38**	**96.05%**	**0**
Case 5	1.30	95.12%	**0**	2.27	97.56%	**0**	69.00	92.86%	**0**	**5.84**	**0.49**	**97.56%**	**0**
avg.	1.84	93.13%	25.8	4.65	95.12%	1.00	178.80	95.13%	**0.20**	**9.45**	**0.75**	**98.16%**	0.40
ratio	2.46	-	-	6.21	-	-	18.92	-	-	**1.0**	**1.0**	-	-

Table 2: Comparisons with three state-of-the-art hotspot detectors [4, 10, 20]

	TCAD'14 [4]			TCAD'15 [20]			ICCAD'16 [10]			Ours		
	CPU(s)	Accuracy	FA#	CPU(s)	Accuracy	FA#	CPU(s)	Accuracy	FA#	CPU(s)	Accuracy	FA#
Case 1	11	100.00%	1714	38	94.69%	1493	10	**100.00%**	788	**4**	**100.00%**	**783**
Case 2	287	**99.80%**	4058	234	98.20%	11834	103	99.40%	**544**	**17**	99.40%	700
Case 3	417	93.80%	9486	778	91.88%	13850	110	97.51%	**2052**	**49**	**97.78%**	2166
Case 4	102	91.00%	**1120**	356	85.94%	3664	69	**97.74%**	3341	**14**	96.05%	2132
Case 5	49	87.80%	199	20	92.86%	1205	41	95.12%	94	**9**	**97.56%**	**52**
avg.	173.2	94.48%	3315.4	285.2	92.71%	6409.2	66.6	97.95%	1363.8	**18.4**	**98.16%**	**1166.6**
ratio	9.40	-	2.84	15.50	-	5.49	3.62	-	1.17	**1.0**	-	**1.0**

as $w_i = 2^{i-1}$. For each test case, a set of training data are used to construct our bilinear classifier, while another set of data are used to evaluate the performance of the classifier. Since the feature extraction for each clip is a separate procedure, we use multi-core processing techniques to accelerate the feature extraction step.

6.1 Performance Comparison with Classical Classifiers

In the first experiment, we compare our proposed BL-HSD framework with other classical hotspot detection frameworks. SVM [11] and Adaboost [11] classifiers are widely used in many applications, which also demonstrate their superiority in hotspot detection [8, 9]. Thus in this experiment, we investigate the performance of our BL-HSD with two classical classifiers and one recent work: 1) VCCS feature + SVM classifier (denoted as VCCS-SVM); 2) VCCS feature + Adaboost classifier (denoted as VCCS-Adaboost); 3) Density based feature + Adaboost classifier (denoted as DBF-Adaboost) [9]. Note that during the testing layout scanning in the detector of [9], only the core area of each clip will be verified. In order to have a fair comparison, our BL-HSD framework also scans the core area.

A detailed comparison of our BL-HSD and other classical classifiers are shown in Table 1. For each method in the Table 1, Columns "**M-CPU(s)**", "**CPU(s)**", "**Accuracy**" and "**FA#**" list the runtime of the model in seconds (sum of training and testing time of the model), the runtime of the overall flow in seconds (including the feature extraction time, and model training and testing time), the accuracy of the method, and the number of false alarms. We can see from Table 1 that our BL-HSD method outperforms all the other three classical methods in terms of accuracy, false alarm number and runtime performance. Particularly, comparing with both VCCS-SVM and VCCS-adaboost, our framework can achieve at least 2× speed-up in model CPU performance and can increase detection accuracy from 95.12% to 98.16%. In addition, our method also outperforms the recent hotspot detector [9], where our approach achieves 19× speed-up in the overall CPU performance and improves the accuracy by 3.03%. Meanwhile, for all five test cases, only 2 false alarms are reported in our framework.

Our framework achieves extremely fast speed in model runtime (only model training and testing time, denoted as M-CPU(s)) compared to other classical methods, which may be related to the proposed bilinear classifier, where the model complexity is reduced by incorporating weighted nuclear norm penalty.

6.2 Performance Comparison with state-of-the-art methods

In the second experiment, we further compare our framework with three state-of-the-art hotspot detection frameworks [4, 10, 20]. The details of comparisons are listed in Table 2, and for each detector, columns "**CPU(s)**", "**Accuracy**" and "**FA#**" are the same as those in Table 1. In this experiment, we first decompose the layout designs of each test case of the ICCAD-2012 benchmark suite [23] into a set of independent clips, whose size is the same as the core area in the training layout. We then scan all grids and verify their labels in our BL-HSD hotspot detector.

It can be observed from Table 2 that our method outperforms all other state-of-the-art methods in terms of runtime, accuracy and the number of false alarms on average. Our method achieves around 9× better running time performance comparing to [4], 15× comparing to [20], and 4× comparing to [10]. The superiority of the runtime performance is related to the simplicity of both the MCCS feature which only samples several points from the layout, and bilinear classifier which achieves faster convergence speed by discovering structural correlations and reducing model complexity. Besides, our method improves the accuracy by 3.68%, 5.45% and 0.21% on average compared with [4], [20] and [10]. We also achieve around 3× and 5× reduction in false alarms compared with [4] and [20] respectively. Although the work [10] performs feature optimization in the hotspot detection framework, it only put local correlation in the layout patterns into consideration; hence, we still on average reduce around 200 false alarms compared to [10]. Our method considers the global correlations of different points and circles during feature extraction and classifier construction, which allows us to capture the hidden structural information in lithography process. It can be observed that our method is more efficient compared to other state-of-the-art methods.

7. CONCLUSION

In this paper, we propose a novel BL-HSD hotspot detection framework, which incorporates a novel MCCS feature extraction method and an efficient bilinear machine learning model. With MCCS feature, the hidden structural information of the circuit layout patterns is well preserved. With bilinear machine learning model, the hidden information that comes from light propagation and interference can be efficiently and accurately captured. In addition, we proved the excess risk bound of the bilinear model theoretically. More importantly, our framework outperform state-of-the-art methods in all evaluation terms on average. With accurate capture of lithography process phenomenon, our method can be widely used not only in the hotspot detection problem, but also in other research problem in DFM which involves lithography process, such as OPC, SRAF insertion and EPE value prediction.

8. REFERENCES

[1] Jingyu Xu, Subarna Sinha, and Charles C. Chiang. Accurate detection for process-hotspots with vias and incomplete specification. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 839–846, 2007.

[2] Yen-Ting Yu, Ya-Chung Chan, Subarna Sinha, Iris Hui-Ru Jiang, and Charles Chiang. Accurate process-hotspot detection using critical design rule extraction. In *ACM/IEEE Design Automation Conference (DAC)*, pages 1167–1172, 2012.

[3] Sheng-Yuan Lin, Jing-Yi Chen, Jin-Cheng Li, Wan-Yu Wen, and Shih-Chieh Chang. A novel fuzzy matching model for lithography hotspot detection. In *ACM/IEEE Design Automation Conference (DAC)*, pages 68:1–68:6, 2013.

[4] Wan-Yu Wen, Jin-Cheng Li, Sheng-Yuan Lin, Jing-Yi Chen, and Shih-Chieh Chang. 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.

[5] Dragoljub G. Drmanac, Frank Liu, and Li-C. Wang. Predicting variability in nanoscale lithography processes. In *ACM/IEEE Design Automation Conference (DAC)*, pages 545–550, 2009.

[6] Duo Ding, J. Andres Torres, and David 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 (TCAD)*, 30(11):1621–1634, 2011.

[7] Duo Ding, Bei Yu, Joydeep Ghosh, and David Z. Pan. EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 263–270, 2012.

[8] Yen-Ting Yu, Geng-He Lin, Iris Hui-Ru Jiang, and Charles Chiang. Machine-learning-based hotspot detection using topological classification and critical feature extraction. In *ACM/IEEE Design Automation Conference (DAC)*, pages 671–676, 2013.

[9] Tetsuaki Matsunawa, Jhih-Rong Gao, Bei Yu, and David Z. Pan. A new lithography hotspot detection framework based on AdaBoost classifier and simplified feature extraction. In *Proceedings of SPIE*, volume 9427, 2015.

[10] Hang Zhang, Bei Yu, and Evangeline FY Young. Enabling online learning in lithography hotspot detection with information-theoretic feature optimization. In *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2016.

[11] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2001.

[12] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). 28(2):337–407, 2000.

[13] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[14] Tetsuaki Matsunawa, Shigeki Nojima, and Toshiya Kotani. Automatic layout feature extraction for lithography hotspot detection based on deep neural network. In *SPIE Advanced Lithography*, pages 97810H–97810H. International Society for Optics and Photonics, 2016.

[15] Lior Wolf, Hueihan Jhuang, and Tamir Hazan. Modeling appearances with low-rank svm. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007.

[16] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Bilinear classifiers for visual recognition. In *Advances in neural information processing systems*, pages 1482–1490, 2009.

[17] Luo Luo, Yubo Xie, Zhihua Zhang, and Wu-Jun Li. Support matrix machines. In *International Conference on Machine Learning (ICML)*, 2015.

[18] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[19] Shuhang Gu, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International Journal of Computer Vision*, pages 1–26, 2016.

[20] Yen-Ting Yu, Geng-He Lin, Iris Hui-Ru Jiang, and Charles Chiang. Machine-learning-based hotspot detection using topological classification and critical feature extraction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 34(3):460–470, 2015.

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

[22] Tetsuaki Matsunawa, Bei Yu, and David Z. Pan. Optical proximity correction with hierarchical bayes model. In *Proceedings of SPIE*, volume 9426, 2015.

[23] Andres J. Torres. ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 349–350, 2012.

[24] Alexander J Smola. *Advances in large margin classifiers*. MIT press, 2000.

[25] Hua Zhou and Lexin Li. Regularized matrix regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):463–483, 2014.

[26] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[27] Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014.

[28] John Platt et al. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[29] S. Sathiya Keerthi and Elmer G Gilbert. Convergence of a generalized smo algorithm for svm classifier design. *Machine Learning*, 46(1-3):351–360, 2002.

[30] Andreas Maurer and Massimiliano Pontil. Excess risk bounds for multitask learning with trace norm regularization. In *Conference on Learning Theory (COLT)*, volume 30, pages 55–76, 2013.