Pin Accessibility Prediction and Optimization with Deep Learning-based Pin Pattern Recognition*

Tao-Chun Yu, Shao-Yun Fang, Hsien-Shih Chiu, Kai-Shun Hu, Philip Hui-Yuh Tai, Cindy Chin-Fang Shen, and Henry Sheng

Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan

{d10407012, syfang}@mail.ntust.edu.tw

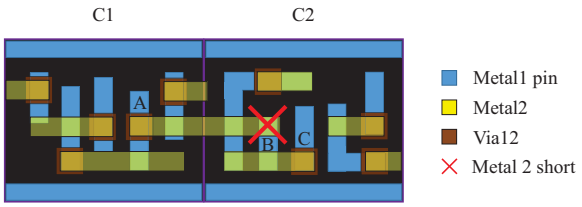

Figure 1: An example of the pin access problem.

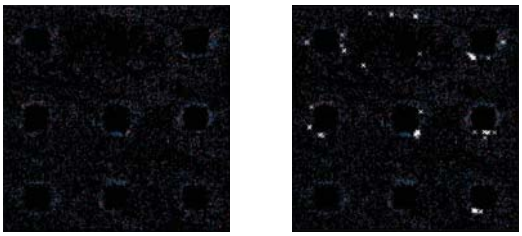

Figure 2: The low correlation between a GR congestion map and a DRV map. (a) A GR congestion map. (b) The corresponding DRV map.

1 INTRODUCTION

Since the process node of semiconductor keeps scaling down, standard cells become much smaller and cell counts are dramatically increased. Standard cells on the lower metal layers severely suffer from low routability due to high pin density, low pin accessibility, and limited routing resources. Fig. 1 gives an example of the bad pin accessibility of cell $C2$ affected by the adjacent cell $C1$. It can be observed that the access points of pin B are blocked by the metal 2 (M2) routing segments routed from Pin A and Pin C, so an M2 short design rule violation (DRV) will be induced when dropping a via12 on Pin B. This example shows that pin accessibility is not only determined by cell layout design but also strongly affected by adjacent cells.

In order to tackle the pin access problem, many previous works focus on the enhancement of pin accessibility during layout design of standard cells [8–16]. [8, 9, 12, 15, 16] consider pin access planning considering the design constraints of self-aligned double patterning (SADP) lithography. [14] solves the escape routing problem for a dense pin cluster. [10] models a weighted interval assignment approach for pin access optimization, and the problem is solved with a binary integer linear programming formulation followed by an iterative Lagrangian relaxation algorithm to maximize the number of pin access intervals. [11] proposes a placement co-optimization work considering pin accessibility measurement and cell layout redesign, where a greedy heuristic is used to solve whether a cell requires an extra whitespace or needs to be redesigned. [13] invents a cost function computing the total cell pin access penalty (TCPAP) and proposes a dynamic programming-based detailed placement to iteratively refine the cell position in a cell row. However, these studies may suffer from either of the two deficiencies: (1) Cell libraries provided by foundries should not be considerably redesigned because the optimized cell performance and manufacturability may be highly sensitive to cell layouts. (2) Deterministic approaches based on human knowledge have been shown to be less effective in advanced nodes for optimization problems such as DRV prediction and minimization because of the extremely high complexity through the overall design flow.

Due to the difficulty of analyzing the complicated causes of DRV occurrences, more and more recent works resort to the prediction ability of machine learning-based techniques [1–7]. [1] and [6] propose a support vector machine (SVM)-based methods to respectively predict the locations of DRVs and the routability of placements. [2] and [4] predict the number of DRVs and congested regions of a given cell placement by considering GR congestion maps. [7] proposes a CNN-based routability predictor to forecast overall routability of a placement with accuracy similar to that of global router while using less runtime. [5] and [3] respectively use the ensemble RUSBoost algorithm and a weighted neural network to predict whether a given tile has M2 short violations or not. For most of the above works focusing on routability prediction and DRV prediction, global routing (GR) congestion and pin density are commonly used as the main features for model training. Empirically, however, DRV occurrence is not necessary to be strongly correlated with the two features, especially for designs in advanced process nodes. 但是后面还是有很多这种工作

An example is shown in Fig. 2. Fig. 2(a) gives the GR congestion map of a revised industrial design, where the blue and red dots indicate congested regions, and the red dots are the most congested regions. Fig. 2(b) indicates the locations of all M2 DRVs with white crosses. It can be observed that most of the congested regions in the layout do not have DRVs, while some regions with DRVs are not so congested. Another example is given in Fig. 3. Figs. 3(a) and (b) respectively depict the snapshots of two local regions with the same area. Obviously, the pin density in Fig. 3(a) is greater than the pin density in Fig. 3(b). However, Fig. 3(a) is DRV-clean and Fig. 3(b) has two DRVs, which are caused by bad pin accessibility rather than high pin density. Observing from Fig. 3, although DRVs do not necessarily depend on pin density, they may have a great correlation with pin shapes and the proximity relationship among pins. As illustrated in Fig. 3(b), the two M2 shorts occur at the locations having the same pin pattern in the top cell-row and mid cell-row. This inspection inspires us that it is important to avoid generating the pin patterns that are already known to have bad pin accessibility and similar pin patterns that may also have poor pin accessibility.

Another deficiency of many existing works is that their DRV prediction models can hardly be applied to help DRV reduction during physical design. [1] is the only work using a naive approach that adopts their model to refine placement. First, a set of local windows are found where the potential DRV hotspots are identified with their prediction model. Then, a white space redistribution method is used to add small temporary keepout regions adjacent to the cells in the windows. Finally, cells are spread with re-legalization to increase cell spacings in the windows. However, the DRVs in a window may be only due to two non-ideally abutted cells, and thus their method may misguide the legalizer to do

*This work was partially supported by Synopsys, TSMC, and MOST of Taiwan under Grant No's MOST 108-2636-E-011-002.

DAC '19, June 2–6, 2019, Las Vegas, NV, USA

ACM ISBN 978-1-4503-6725-7/19/06…$15.00

https://doi.org/10.1145/3316781.3317882

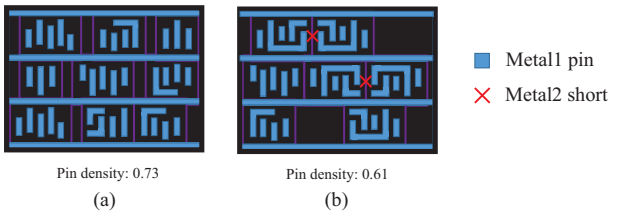

Figure 3: Two snapshots with the same area but different placement locations. (a) A partial snapshot with higher pin density. (d) A partial snapshot with lower pin density.

some redundant moves. In addition, spreading cells in local windows may cause new DRVs in the neighboring regions, resulting in ripple effects of DRV generation.

In this paper, we propose a work of pin accessibility prediction and optimization with a convolution neural network (CNN)-based model, which uses pin pattern as the main feature to predict DRV occurrence. Note that we only target on predicting M2 short DRVs as they are most relevant to the pin access problem [3]. The major contributions of this paper are summarized as follows:

- To the best of the authors knowledge, this is the first work to apply pin pattern as the input features of DRV prediction models. By regarding the pin pattern as the main feature, two models, the pin pattern recognition (PPR) model and the design feature-aware pin pattern recognition (DFPPR) model, are proposed that can precisely identify the locations of pin patterns easily causing DRVs.
- Unlike most of existing models that can only be used for DRV prediction, we successfully apply the pre-trained PPR model for pin accessibility optimization during physical design. A three-stage model-guided detailed placement algorithm is proposed to avoid generating pin patterns with bad pin accessibility.
- The experimental results show that in terms of all quantitative metrics, the proposed PPR model and DFPPR model are greatly superior than those of previous works. In addition, the numbers of M2 shorts and overall DRVs are dramatically reduced by 79% and 51% on average by applying the proposed model-guided detailed placement algorithm, which are significant improvements compared to the design methodology proposed by [1].

The rest of this paper is organized as follows: Section 2 introduces the problem formulation. In Section 3, the proposed models and the model-guided detailed placement algorithm are detailed. Experimental results are presented in Section 4. Finally, we give the conclusion of our work in Section 5.

2 PROBLEM FORMULATION

The goal of this paper is divided into two stages, which are given in the following two problem formulations:

Problem 1: Given a number of routed designs with reported DRV violations, the task is to develop deep learning-based models using pin pattern as the major feature to precisely predict whether a given layout clip has DRVs or not.

Problem 2: Given a legalized placement and the pre-trained model, the objective is to minimize the number of DRVs by avoiding pin patterns with bad pin accessibility predicted by the pre-trained model during detailed placement.

3 PROPOSED METHODOLOGY

3.1 Training Data Preparation

The overview of the proposed approach is given, as shown in Fig. 4, which can be divided into three stages: training data preparation, model training, and model-guided detailed placement. Two models are proposed: the pin pattern recognition (PPR) model uses pin pattern as the only feature to predict DRVs caused by bad pin accessibility, and the design feature-aware pin pattern recognition (DFPPR) model adopts more design features to further enhance the prediction accuracy. The

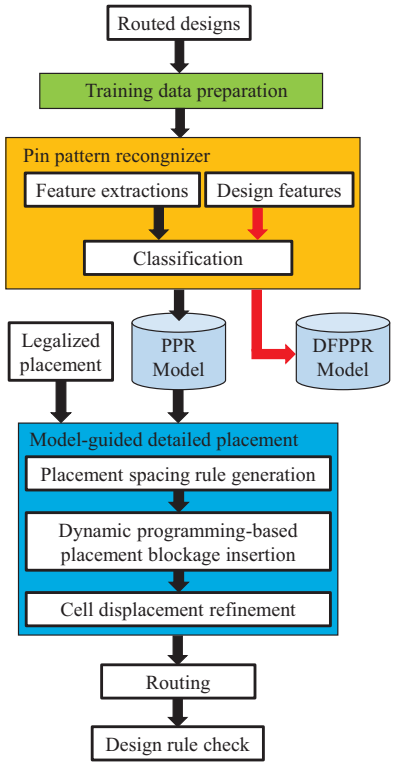

Figure 4: The overview of the proposed approach.

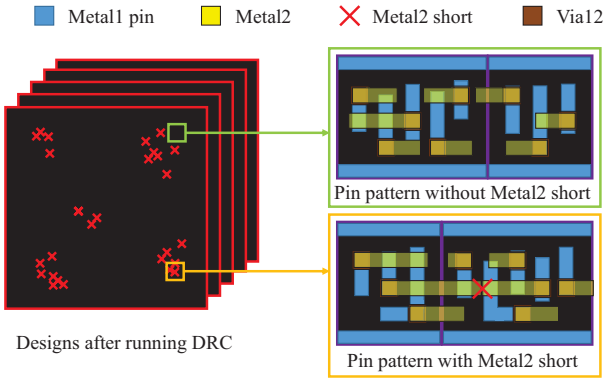

Figure 5: Training data extraction from given designs.

model-guided detailed placement is then proposed to avoid generating pin patterns with bad pin accessibility during detailed placement. The proposed model-guided detailed placement is composed of three stages: the placement spacing rules are first generated by applying the PPR model, which serve as the hard constraints for detailed placement. Then, a dynamic programming-based detailed placement algorithm is applied to minimize the number of inserted placement blockages of each cell row. Finally, cell displacement refinement is used to further optimize the placement.

In order to apply supervised learning, it is necessary to obtain all labels of training data. Since our objective is to identify whether a given pin pattern will induce M2 short or not, we need to generate some routed designs for collecting sufficient amount of pin patterns as training data. As shown in Fig. 5, we first generate some routed designs with reported

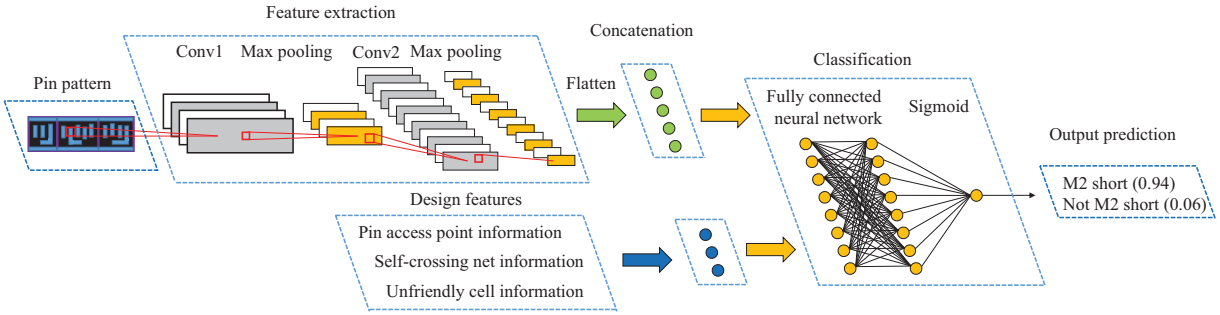

Figure 6: The architecture of the proposed CNN model.

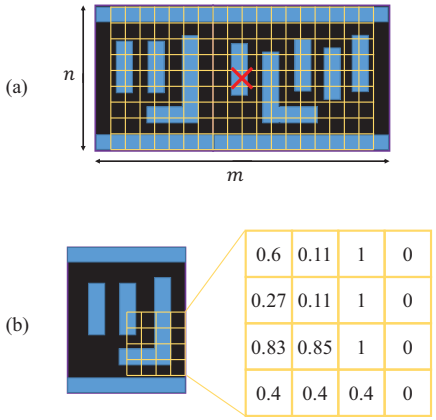

Figure 7: Pin pattern image quantification. (a) Divide the given image into pixels. (b) Compute tha pixel values according to pin covering ratios.

M2 shorts location. Subsequently, we extract all the pin patterns with M2 shorts and some pin patterns without M2 shorts to construct our training data set. Positioning an M2 short at the middle of the pin pattern, a pin pattern image can be obtained by respectively setting the width and the height of the image as 2×single-row-height and single-row-height, as shown in Fig. 7(a). By using the above width setting, an pin pattern image can usually cover at least two cells, and thus pin accessibility affected by adjacent cells can be considered.

3.2 Model Training

In this section, two deep learning models are proposed for DRV prediction. The first one is the pin pattern recognition (PPR) model, whose training flow is shown in Fig. 6 without the block of design features. In the feature extraction stage, each input pin pattern image is fed into the CNN to extract the representative features. Then, all features are flattened and connected to the fully connected neural network for classification. The other one is the design feature-aware pin pattern recognition (DFPPR) model. Different from the PPR model, the representative features obtained by the CNN computation will be concatenated with other design-related features before being fed into the classification block. The proposed two models are detailed as follows.

3.2.1 Pin Pattern Recognition (PPR) Model. The architecture of the proposed PPR model consists of the feature extraction block and the classification block. Before feature extraction, an input pin pattern image needs to be quantified. We first split each pin pattern into $m \times n$ pixels, as shown in Fig. 7(a). Both the width and height of each pixel are set as the minimum spacing of the M1 layer in order to prevent a pixel from being occupied by two different pins. The value of each pixel is computed by the corresponding occupied pin ratio. As shown in Fig. 7(b), the occupied pin ratio of each pixel is computed by the area covered by the overlapped pin over the pixel area. After that, these quantified pixel values are fed into the feature extraction stage. The feature extraction stage includes two convolution layers interleaved with two max pooling layers. The convolution layers extract the critical characteristic features of the given pin patterns, which are weighted by the trainable filters. The max pooling layers are used to reduce the amount of parameters and keep those representative parameters by filtering out noises. Once the feature extraction is completed, all extracted features are flattened and fed into the classification stage. The classification block consists of a fully connected neural network and a sigmoid function. The fully connected neural network is a deep neural network (DNN) containing several layers of neurons to provide more levels of abstraction for complicated features. The sigmoid function is usually applied to be the final layer of binary classification problems, which is used to scale the output value to the range between 0 and 1. This scaling can help the model to decide whether the given input pin pattern will induce M2 short or not based on a predefined threshold value. If the output value is no less than 0.5, the model will regard a given input pin pattern as an M2 short candidate.

3.2.2 Design Feature-aware Pin Pattern Recognition (DFPPR) Model. The architecture of the proposed DFPPR model is roughly the same as that of PPR model. The main difference is that some features extracted from a given design can be additionally considered, which shows the great flexibility that users can adopt any feature correlated with DRV occurrence in combination with the features of pin pattern to further enhance the prediction accuracy. As shown in Fig. 6, not only the feature extracted from pin patterns but three design features are concatenated before the classification block. The design features we adopt in this work are detailed as follows:

- *Pin access point information*: This feature is computed by the ratio of the number of pins to the number of access points in the input pin pattern. ? ?
- *Self-crossing net information*: A self-crossing net is caused by the fly lines of two local nets that intersect to each other. A fly line is the straight line connecting the left-bottom access points of two pins. This feature is computed by the ratio of the number of self-crossing nets to the number of local nets in the input pin pattern.
- *Unfriendly cell information*: This feature is inspired by the feature of "unfriendly cells" used in [1]. With routed designs, we first compute the frequency that each library cell induces M2 shorts. Then, this feature is obtained by the ratio of the the sum of the frequencies of the cells covered by the pin pattern to the total frequency of all cells.

3.3 Model-guided Detailed Placement

Based on the proposed PPR model, we proposed a model-guided detailed placement algorithm to guide the detailed placer to avoid generating pin patterns having high probability to cause M2 shorts. The proposed model-guided detailed placement contains three stages: (1) placement spacing rule generation, (2) dynamic programming-based placement blockage insertion, and (3) cell displacement refinement, which are detailed in the following.

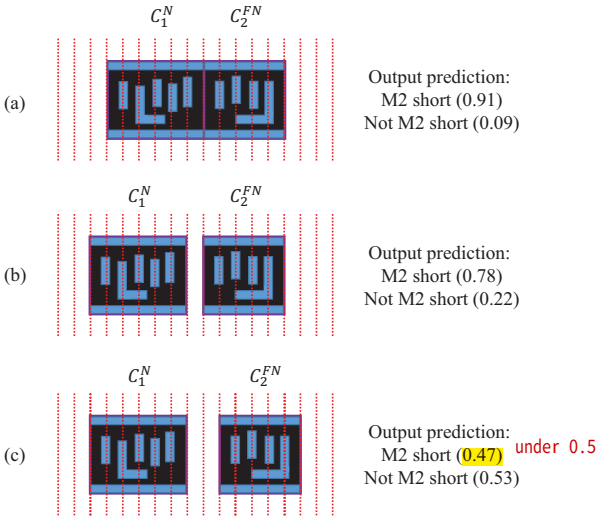

Figure 8: An example for generating placement spacing rule. (a) The pin pattern of two abutting cells. (b) The pin pattern with one-site spacing. (c) The pin pattern with two-site spacing.

3.3.1 Placement Spacing Rule Generation. Despite the fact that the proposed PPR model can precisely identify whether a pin pattern will induce M2 short or not, users cannot simply use it to directly avoid generating such bad pin patterns during physical design. Therefore, it is desirable to develop a DRV-aware detailed placer based on the pre-trained model. Our idea is to use the proposed PPR model to generate a set of required cell spacing rules, which is a one-time process and the rule set can be applicable to any design using the same cell library. Fig. 8 gives an example of placement spacing rule generation. Suppose that we have a pin pattern composed of C_1 with orientation N, C_1^N, and C_2 with orientation FN, C_2^{FN}, with M2 short predicted by the pre-trained model, as shown in Fig. 8(a). Without shifting the cells C_1^N and C_2^{FN}, the output predicted value returns 0.94, which could be regarded as the probability of having an M2 short. In order to prevent this undesirable pin pattern from being placed on a given design, we try to iteratively increase the spacing between the cells until they have sufficient pin accessibility predicted by our model. As shown in Fig. 8(b), we first shift the cell C_1 one site to the left and feed the modified pin pattern into the pre-trained model. The output predicted value is given by 0.78, which is still greater than the default threshold value 0.5. Therefore, we keep shifting the cell C_2 one site to the right, as shown in Fig. 8(c). The output predicted value of the updated pin pattern is now 0.47, which is smaller than the default threshold value. Thus, we set the required spacing rule between Cells C_1 and C_2 as two sites. Based on the idea above, a set of spacing rules will be generated between every pair of cell combination. The following formula gives the required number of placement sites (RPS) for the i^{th} cell with orientation m C_i^m and the j^{th} cell with orientation n C_j^n in the cell library.

$$RPS(C_i^m, C_j^n) = k. \tag{1}$$

Taking Fig. 8(c) as an example, since the required spacing between C_1^N and C_2^{FN} equals the width of two placement sites, we will set a rule for this cell pair as $RPS(C_1^N, C_2^{FN}) = 2$.

3.3.2 Dynamic Programming-based Placement Blockage Insertion. Having a set of minimum required numbers of placement sites between each cell pair, we integrate these spacing rules into a detailed placer for optimization. A dynamic programming-based placement blockage insertion is proposed in this section to minimize the total amount of inserted placement blockages in a cell row considering cell orientations. Some of the notations are first defined in Table 1.

Table 1: The notations of dynamic programming-based placement blockage insertion.

C	A set of cells in the considered row.
c_i^o	i^{th} cell in the considered row with orientation o.
PB	A set of placement blockages in the considered row.
$pb_{i,j}^{mn}$	The smallest width of placement blockage that is required to be inserted between c_i^m and c_j^n 这个怎么算的？
S	Source node.
T	Target node.
$f_o[i]$	Given that the i^{th} cell is with orientation o, the minimum width of all placement blockages required by the subset of cells from the 1^{st} cell to the i^{th} cell with optimal cell orientations.
w_i	The width of the i^{th} cell.

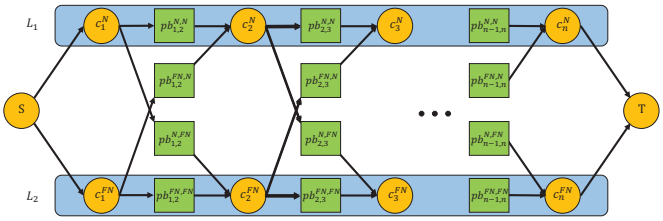

Figure 9: The architecture of the dual-path graph.

The minimal width of required placement blockage between each cell pair can be obtained by Equation (1). For instance, if we have a spacing rule $RPS(C_1^N, C_2^{FN}) = 1$, the minimal width of required placement blockage between C_1^N and C_2^{FN} will be 1 site. In order to minimize the number of DRVs during detailed placement, we propose a dual-path graph $G = \{C, PB, S, T\}$, whose architecture is depicted in Fig. 9. The source node S and target node T are created to the leftmost and rightmost of the graph, respectively. For each cell, we can only use one of two orientation sets $\{N, FN\}$ or $\{S, FS\}$ without violating the power/ground rail constraint (we take N, FN as our example in Fig. 9). The cell nodes with orientation N are created and positioned to Line 1 (L_1), and the cell nodes with orientation FN are created and positioned to Line 2 (L_2). Four different placement blockage nodes are connected from c_i^o to c_{i+1}^o since there are four orientation combinations for a pair of adjacent cells, which give the minimum required spacing between each cell combination.

With the notations and the proposed dual-path graph above, we have the following formulas to compute the minimum cost of $f_N[i]$ and $f_{FN}[i]$ using the following recursions:

如何确定移动哪个cell

动态规划

$$f_N[i] = \begin{cases} 0, & \text{if } i = 0, \\ f_N[i-1] + w_i, & \text{if } i = 1, \\ w_i + \min(f_N[i-1] + pb_{i-1,i}^{N,N}, & \\ \quad f_{FN}[i-1] + pb_{i-1,i}^{FN,N}), & \text{if } i \geq 2. \end{cases} \tag{2}$$

$$f_{FN}[i] = \begin{cases} 0, & \text{if } i = 0, \\ f_{FN}[i-1] + w_i, & \text{if } i = 1, \\ w_i + \min(f_{FN}[i-1] + pb_{i-1,i}^{FN,FN}, & \\ \quad f_N[i-1] + pb_{i-1,i}^{N,FN}), & \text{if } i \geq 2. \end{cases} \tag{3}$$

After solving the formulas above, the optimal orientation of each standard cell and the minimal width of required placement blockage between each pair of two consecutive cells will be respectively obtained. The computational complexity of this dynamic programming-based algorithm is $O(RC)$, where R is the number cell rows and C is the maximum number of cells in a row.

3.3.3 Cell Displacement Refinement. Having all optimal cell orientations and required placement blockages, we then apply Abacas proposed in [18] to minimize the displacements of all standard cells row by row. With the pre-inserted placement blockages, all pin patterns with bad pin accessibility predicted by the pre-trained model can be effectively

removed. For those rows having insufficient capacities to place the standard cells due to required placement blockages, we will apply the global-move and global-swap techniques proposed in [17] to refine the positions of some cells to their optimal regions. Note that cells can only be moved or swapped without generating additional bad pin patterns predicted by our PPR model.

4 EXPERIMENTAL RESULTS

The EDA tool ICC2 is used to generate different routed designs [22]. We implemented training data preparation part with tool command language (TCL). The PPR and DFPPR training flow are written by Python with Keras neural networks API. After the PPR model is trained and saved, we then transfer the pre-trained model into C++ interface with Tensorflow C++ API [23]. The proposed model-guided detailed placement is also implemented by using the C++ programming language. Experiments were conducted on a Linux machine with 2.6 GHz CPU and 264 GB memory. The testcases for experiments are modified by a real industrial case with the same cell library but different placement settings in order to obtain different placement results and thus more pin patterns. Experimental results are divided into two parts: (1) performance comparison between existing models proposed by the two state-of-the-art works [1, 3] and the two models proposed in this paper, and (2) model-guided placement comparison between the method proposed by [1] and our algorithm. In the model performance comparison, the confusion matrices are given and several quantitative metrics are computed to compare the qualities of these models. In the model-guided placement comparison, we derive placement results respectively by applying the model and the placement refinement method proposed in [1] and by applying our PPR model and the detailed placement flow proposed in this paper. Finally, the amount of DRVs are reported to show the effectiveness of our placement approach compared to [1].

4.1 Model Performance Comparison

Confusion matrix is usually used to summarize the quality of binary classification approaches. The template of confusion matrix is shown in Table 2. The entry TP presents the number of pin patterns with DRVs that are correctly predicted by a model. FP gives the number of pin patterns without DRVs while a model asserts that they will induce DRVs. FN presents the number of pin patterns with DRVs while a model predicts that they will not cause DRVs. Finally, FN shows the number of pin patterns without DRVs that are correctly predicted by a model. Based on the four numbers, five important quantitative metrics are listed as follows:

- *Accuracy* ($\frac{(TP+TN)}{TP+TN+FP+FN}$): Give the probability that the model will produce a correct prediction.
- *True positive rate* ($\frac{TP}{TP+FN}$): Also known as recall, which is used to measure the proportion of actual pin patterns with DRVs that are correctly identified by a model.
- *Positive predictive value* ($\frac{TP}{TP+FP}$): Also known as precision, which is used to measure the proportion of predicted pin patterns with DRVs that are really bad pin patterns.
- *False positive rate* ($\frac{FP}{FP+TN}$): Also known as false alarm, which is used to measure the proportion of predicted pin patterns with DRVs that are actually DRV-free pin patterns.
- *F-measure* ($2\times\frac{Recall\times Precision}{Recall+Precision}$): Give the harmonic average of the precision and recall, which is larger if precision and recall are maximized and the small difference of precision and recall is minimized.

Table 3 gives the confusion matrices of the models proposed by [3], [1], and ours. All entries are marked by the order as [3]/[1]/PPR/DFPPR. In order to collect enough amount of training data and testing data, we randomly generate 50 different routed designs. Since the amount of M2 shorts is limited, we extract all pin patterns with M2 shorts. The number of extracted pin patterns without M2 shorts is four times the number of those with M2 shorts. The amount of training data and testing data of [3] and [1] may be fewer than ours since their models predict whether a local window has DRVs and it is possible for a local window to have more than one DRV. In contrast, our approach extracts a pin pattern for each DRV.

Table 2: The template of confusion matrix.

Testing		Actual	
		Positive	Negative
Prediction	Positive	TP	FP
	Negative	FN	TN

Table 3: Confusion matrix of the model proposed by [3]/[1]/PPR/DFPPR

Testing		Actual	
		Positive	Negative
Prediction	Positive	78/65/95/105	29/37/26/13
	Negative	29/42/20/2	399/391/460/415

According to the confusion matrix shown in Table 3, we compute the five quantitative metrics, which are summarized in Table 4. A better machine learning model should have higher accuracy, recall, precision, and F-measure and have lower false alarm. In addition, in the DRV prediction problem, the F-measure metric plays the most important role. For a model with the higher F-measure, the actual DRVs can be precisely identified with a small false alarm. Obviously, the proposed PPR model outperforms the models proposed by [1] and [3] for all quantitative metrics, which shows that pin pattern can serve as the critical feature to identify M2 shorts caused by bad pin accessibility. Moreover, the DFPPR model performs even better compared to the PPR model, which shows the great potential of further model improvement by adopting the proposed deep learning architecture.

4.2 Model-Guided Placement Comparison

We further generate eight different legalized placements and compare the routing results by adopting our model and the proposed model-guided detailed placement flow with those of original placements and those derived by adopting the model and the placement refinement approach proposed in [1]. The experimental results are summarized in Table 5. The numbers of total DRVs and M2 shorts are respectively denoted as "#DRVs" and "#M2 sh". "Avg dis." denotes the average cell displacement in sites, and "WL" denotes the total wire length in nanometer. The eight placement testcases are marked as P0–P7. Compared to the default placements, the naive placement refinement approach proposed by [1] can reduce 26% M2 shorts on average, while our flow can considerably reduce M2 shorts by 79% with only 1% wirelength overhead. In addition, The total number of DRVs is also greatly reduced by 51%. The results not only verify the good prediction ability of the proposed PPR model again but also show the effectiveness of the proposed detailed placement flow.

Figs. 10(a), (b), and (c) separately depict the partial placement results of the same location produced by the default placement, the placement refinement approach proposed by [1], and our detailed placement flow. In order to solve the M2 short DRV in the right local window shown in Fig. 10(a), the placement refinement approach uses a white spacing redistribution methodology to add the small temporary keepout region between the red cell and the yellow cell to spread them. The legalizer moves the red cell to the left side to resolve the DRV, as shown in Fig. 10(b). However, the red cell become closer to the orange cell in the adjacent window and induce a new M2 short DRV due to bad pin accessibility, which illustrates the ripple effect that cannot be considered by refining local regions. Fig. 10(c) illustrates a DRV-clean result produced by adopting our detailed placement flow, where two placement blockages are inserted to simultaneously consider the pin accessibility between the two pairs of cells.

5 CONCLUSIONS

We have proposed the first work of pin access-aware DRV prediction by regarding pin pattern as the main feature to train the models. The PPR model is first proposed to precisely identify whether a given pin pattern will induce M2 short or not. Then, several design features can also be considered with pin pattern to train the DFPPR model. Finally, a model-guided detailed placement flow is proposed to optimize pin accessibility during placement. Experimental results show that not only the proposed models outperform those proposed in state-of-the-art works in terms

Table 4: Comparison of five quantitative metrics computed by confusion matrix.

	[3]		[1]		PPR		DFPPR	
	Value	Comp.	Value	Comp.	Value	Comp.	Value	Comp.
Accuracy	89.16%	0.97	85.23%	0.92	92.35%	1.00	97.20	1.05
True positive rate	72.90%	0.88	60.75%	0.74	82.61%	1.00	98.13	1.19
Positive predictive value	72.90%	0.93	63.73%	0.81	78.51%	1.00	88.98	1.13
False positive rate	6.78%	1.27	8.64%	1.61	5.35%	1.00	3.04	0.57
F-measure	72.90	0.91	62.20	0.77	80.51	1.00	93.33	1.16

Table 5: Comparisons between default placement, placement refinement in [1], and our model-guided detailed placement flow.

	Default			[1]				Ours				
	#DRVs	#M2 sh	WL	#DRVs	#M2 sh	Avg dis.	WL	#DRVs	#M2 sh	Avg dis.	WL	CPU time (s)
P0	1971	61	5532549	1437	29	0.34	5535994	1007	14	0.42	5606690	391
P1	572	7	5518484	412	6	0.37	5517359	232	0	0.27	5586467	385
P2	663	10	5510817	390	9	0.37	5508906	295	4	0.26	5578543	387
P3	796	19	5503626	675	9	0.37	5505548	288	5	0.26	5573211	388
P4	496	5	5499569	837	18	0.37	5501735	348	3	0.27	5567659	389
P5	678	13	5502588	924	16	0.37	5503557	471	5	0.27	5570773	385
P6	693	21	5503861	726	11	0.37	5501959	251	1	0.26	5572687	390
P7	633	12	5501298	554	17	0.37	5498997	298	2	0.29	5570713	389
Avg	813	19	5509093	744	14	0.37	5509257	399	4	0.29	5578343	388
Comp.	1.00	1.00	1.00	0.92	0.74	1.00	1.00	0.49	0.21	0.78	1.01	挺慢的-

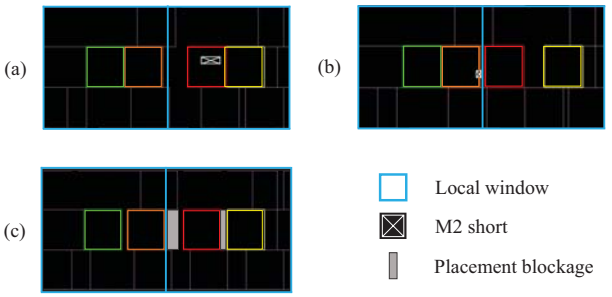

Figure 10: A partial placement result produced by (a) the default placement, (b) the placement refinement approach proposed by [1], and (c) our detailed placement flow.

of all quantitative metrics but also the proposed model-guided detailed placement flow can effectively reduce the numbers of total DRVs and M2 shorts.

REFERENCES

[1] W. T. J. Chan, P.-H. Ho, A. B. Kahng, and P. Saxena, "Routability Optimization for Industrial Designs at Sub-14nm Process Nodes Using Machine Learning," In *Proc. ACM International Symposium on Physical Design (ISPD)*, 2017.

[2] Q. Zhou, X. Wang, Z. Qi, Z. Chen, Q. Zhou, and Y. Cai. "An accurate detailed routing routability prediction model in placement." In *Proc. Asia Symposium on Quality Electronic Design (ASQED)*, 2015.

[3] A. F. Tabrizi, N. K. Darav, S. Xu, L. Rakai, I. Bustany, A. Kennings and L. Behjat. "A Machine Learning Framework to Identify Detailed Routing Short Violations from a Placed Netlist." *Proc. ACM/IEEE Design Automation Conference (DAC)*, 2018.

[4] Z. Qi, Y. Cai and Q. Zhou. "Accurate Prediction of Detailed Routing Congestion using Supervised Data Learning." In *Proc. 2014 IEEE International Conference on Computer Design (ICCD)*, 2014.

[5] A. F. Tabrizi, N. K. Darav, L. Rakai, A. Kennings and L. Behjat. "Detailed Routing Violation Prediction During Placement Using Machine Learning." In *Proc. International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2017.

[6] W. T. J. Chan, Y. Du, A. B. Kahng, S. Nath and K. Samadi. "BEOL stack-aware routability prediction from placement using data mining techniques" In *Proc. IEEE International Conference on Computer Design (ICCD)*, 2016.

[7] Z. Xie, Y. H. Huang, G. C. Fang, H. Ren, S. Y. Fang, Y. Chen and J. Hu. "RouteNet: Routability Prediction for Mixed-Size Designs Using Convolutional Neural Network." In *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018.

[8] X. Xu, B. Yu, J. R. Gao, C. L. Hsu and D. Z. Pan. "PARR: Pin Access Planning and Regular Routing for Self-Aligned Double Patterning." *Proc. ACM/IEEE Design Automation Conference (DAC)*, 2015.

[9] X. Xu, B. Yu, J. R. Gao, C. L. Hsu and D. Z. Pan. "PARR: Pin Access Planning and Regular Routing for Self-Aligned Double Patterning." *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 21, no. 3, article 42, 2016.

[10] X. Xu, Y. Lin, V. Livramento and D. Z. Pan. "Concurrent Pin Access Optimization for Unidirectional Routing." *Proc. of ACM/IEEE Design Automation Conference (DAC)*, 2017.

[11] J. Seo, J. Jung, S. Kim and Y. Shin. "Pin Accessibility-Driven Cell Layout Redesign and Placement Optimization." *Proc. of ACM/IEEE Design Automation Conference (DAC)*, 2017.

[12] W. Ye, B. Yu, D. Z. Pan, Y. C. Ban and L. Liebmann. "Standard Cell Layout Regularity and Pin Access Optimization Considering Middle-of-Line." *Proc. Great Lakes Symposium on VLSI (GLSVLSI)*, 2015.

[13] Y. Ding, C. Chu and W. K. Mak. "Pin Accessibility-Driven Detailed Placement Refinement." In *Proc. ACM International Symposium on Physical Design (ISPD)*, 2017.

[14] M. M. Ozdal. "Detailed-Routing Algorithms for Dense Pin Clusters in Integrated Circuits." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 28, no. 3, pp. 340-349, 2009.

[15] X. Xu, B. Cline, G. Yeric, B. Yu and D. Z. Pan. "Self-Aligned Double Patterning Aware Pin Access and Standard Cell Layout Co-Optimization." In *Proc. ACM International Symposium on Physical Design (ISPD)*, 2014.

[16] X. Xu, B. Cline, G. Yeric, B. Yu and D. Z. Pan. "Self-Aligned Double Patterning Aware Pin Access and Standard Cell Layout Co-Optimization." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 34, no. 5, pp. 699–712, 2015.

[17] M. Pan, N. Viswanathan and C. Chu. "An Efficient and Effective Detailed Placement Algorithm." In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, pp. 48–55, 2005.

[18] P. Spindler, U. Schlichtmann, and F. M. Johannes. "Abacus: Fast Legalization of Standard Cell Circuits with Minimal Movement." In *Proc. ACM International Symposium on Physical Design (ISPD)*, 2008.

[19] K. H. Tseng, Y. W. Chang and C. C. C. Liu. "Minimum-Implant-Area-Aware Detailed Placement with Spacing Constraints." *Proc. of ACM/IEEE Design Automation Conference (DAC)*, 2016.

[20] Z. W. Lin and Y. W. Chang. "Detailed Placement for Two-Dimensional Directed Self-Assembly Technology." *Proc. of ACM/IEEE Design Automation Conference (DAC)*, 2017.

[21] P. Debacker, K. Han, A. B. Kahng, H. Lee, P. Raghavan and L. Wang. "Vertical M1 Routing-Aware Detailed Placement for Congestion and Wirelength Reduction in Sub-10nm Nodes." *Proc. of ACM/IEEE Design Automation Conference (DAC)*, 2017.

[22] Synopsys ICC2 user guide. https://www.synopsys.com

[23] Tensorflow C++ Reference. https://www.tensorflow.org/api_docs/cc/