2023 Design, Automation & Test in Europe Conference (DATE 2023)

Routability Prediction using Deep Hierarchical Classification and Regression

Daeyeon Kim, Jakang Lee, and Seokhyeong Kang*
EE Department, POSTECH, Pohang, South Korea
*shkang@postech.ac.kr

***Abstract*—Routability prediction can forecast the locations where design rule violations occur without routing and thus can speed up the design iterations by skipping the time-consuming routing tasks. This paper investigated (i) how to predict the routability on a continuous value and (ii) how to improve the prediction accuracy for the minority samples. We propose a deep hierarchical classification and regression (HCR) model that can detect hotspots with the number of violations. The hierarchical inference flow can prevent the model from overfitting to the majority samples in imbalanced data. In addition, we introduce a training method for the proposed HCR model that uses Bayesian optimization to find the ideal modeling parameters quickly and incorporates transfer learning for the regression model. We achieved an R2 score of 0.71 for the regression and increased the F1 score in the binary classification by 94% compared to previous work [6].**

I. Introduction

The miscorrelation between placement and routing in advanced technology nodes has increased due to complex design rules and higher cell density [1]. The incorrect planning of the routing resources during placement results in more design iterations to obtain a routable design. However, the design turn-around time (TAT) required for routability optimization is now negligible, as the routing runtime on advanced nodes takes up to several weeks.

Routability prediction is one of the techniques to reduce design TAT. For example, the routability optimization techniques, such as inserting whitespace [2] or generating spacing rules for placement [3], utilize the routability predictor that forecasts the locations of design rule violation (DRV) without the routing. The predicted DRV information is also used to optimize the cost parameters in global routing [4]. Those collaborations between the predictor and optimizer reduce the time-consuming routing process, making the design loop smaller.

Recent studies have used machine learning (ML) to train models from big data because routability modeling suffers from the black-box characteristics and noise of place and route (P&R) tools due to stacking heuristics. Previous works [5]–[7] predicted routability using a pixel-wise binary classification model that detects where DRV has occurred inside a grid cell (Gcell). Although violations tend to cluster in poor routability, previous works annotated labels using only two classes (hotspot or non-hotspot), regardless of the number of DRVs within the Gcell. This binary classification model cannot identify the routability of Gcell with a continuous value. Sophisticated routability optimization requires both where the hotspot is located and how many DRVs exist within the hotspot, but previous work has focused only on violation detection.

Another challenge is the data imbalance problem. Data obtained from real-world observations do not have an ideal uniform distribution and show a skewed distribution toward specific observation values [8]. Most ML algorithms are based on a balanced dataset, and the model will be biased toward the majority classes if a data imbalance exists. In the physical design, hotspots are extremely rare compared to non-hotspots because they are observed when the router does not find a feasible solution. Therefore, it is difficult to obtain high predictive accuracy if general ML techniques are applied to imbalanced data.

In this study, we investigated (i) how to predict the routability on a continuous value and (ii) how to improve the prediction accuracy for the minority samples.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT).(No.2022-0-01172, DRAM PIM Design Base Technology Development). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-00754, Software Systems for AI Semiconductor Design).

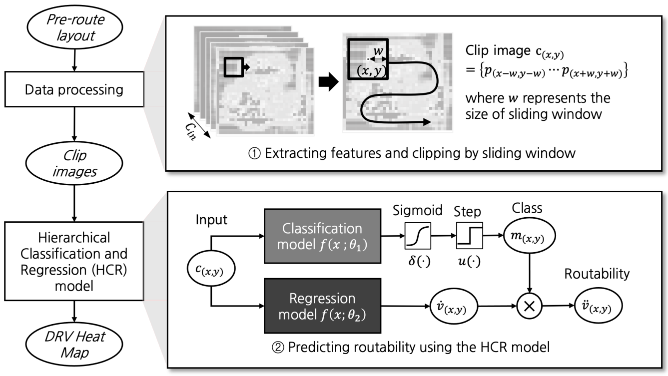

Fig. 1. Routability prediction flow using the HCR model.

II. Proposed Methods

We define the routability prediction as a pixel-wise regression problem. For accurate regression in the imbalanced data, we propose a deep hierarchical classification and regression (HCR) model that detects DRV hotspots with the number of DRVs. The hierarchical structure can reduce the regression error caused by the data imbalance problem by filtering out the majority samples from the classification in advance [9]. We also introduce a training method of the proposed HCR model that (i) searches modeling parameters using the Bayesian optimization (BO) algorithm for the classification model and (ii) trains the regression model using transfer learning only for the data classified as hotspots.

A. Hierarchical Inference Flow

Figure 1 shows the overall routability prediction flow using the HCR model. The model consists of a classification model $f(\mathtt{x};\theta_1)$ and a regression model $f(\mathtt{x};\theta_2)$. Both models share the same network architecture but have different weight parameters (θ_1, θ_2), and take a clipped image $c_{(x,y)}$ as the input x. The clip image $c_{(x,y)}$ represents a set of Gcell features $\{p_{(x-w,y-w)}, \cdots, p_{(x+w,y+w)}\}$ as far as the distance w around (x, y). The output of the classification model is converted into a probability of being *hotspot* using the sigmoid function $\delta(\mathtt{x})$. The probability is transformed into an output class $m_{(x,y)}$ representing *hotspot* or *non-hotspot* through a step function $u(\mathtt{x}) \in \{0, 1\}$, where $u(\mathtt{x})$ is one if x is larger than 0.5, and zero otherwise. The output of the regression model with parameter θ_2 is defined as $\dot{v}_{(x,y)} = f(\mathtt{x};\theta_2)$. The regression output is masked when the output class is *hotspot*. The final prediction is given by $\ddot{v}_{(x,y)} = \dot{v}_{(x,y)} \cdot m_{(x,y)}$. At last, we construct a DRV heat map using the final prediction for every (x, y) coordinate.

B. Training Method

We propose a training method for the proposed HCR model. Figure 2 shows the overall training procedure, which consists of two steps: (i) modeling parameter search for the classification model and (ii) transfer learning for the regression model.

1) Modeling parameter search: The performance of the binary classification model can vary significantly depending on the data configuration and hyper-parameters owing to the data-imbalance problem. Because humans take extensive time to find the optimal hyper-parameters with the data configuration manually, we use automation

978-3-9819263-7-8/DATE23/© 2023 EDAA

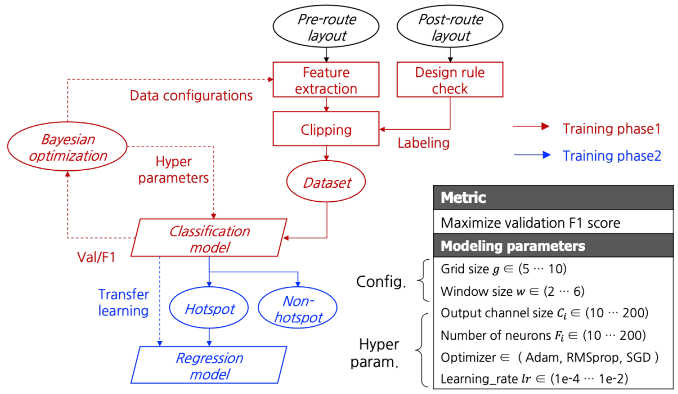

Fig. 2. Training method for the HCR model.

to quickly find the modeling parameters that satisfy the target performance using the BO algorithm [10]. The BO algorithm aims to find the optimal solution that maximizes the value of the objective function and is usually used in cases where it takes a long time to find a solution or in black-box applications. It is more efficient than manually searching for a combination of hyper-parameters and data configurations. Since the F1 score can evaluate the predictive performance of a minority class in an imbalanced binary classification, we set the F1 score as the objective function on the validation dataset. The parameters include the grid size (g), window size (w), size of output channel (C_i), number of neurons (F_i), and learning rate (lr) with the optimizer. We used the binary cross entropy as a loss function and applied 5-fold cross validation for the training. The number of epochs for each fold was set to 20.

2) Transfer learning: The regression model has the same network architecture as the classification model. We initialized the weight parameter of the regression model θ_2 as the weight parameter θ_1 and switched the training dataset to the *hotspot* data. This is called transfer learning, a method for delivering the knowledge of a pre-trained model to partially related tasks with minimal retraining [11]. The regression model can quickly converge to the *hotspot* class. We set the number of epochs to 10 to prevent overfitting and used the root mean square error (RMSE) as the loss function.

III. Experimental Results

A. Design of Experiments

We used artificial designs for training and real-world designs for all tests. The training dataset consisted of 50 artificial circuits generated using *ANG* [12], and the test dataset used five *OpenCores* circuits [13]. We created 180 layouts for each circuit by sweeping the layout utilization, aspect ratio, top routing layer, and P&R clock period using a commercial P&R tool [14]. The training dataset consisted of 9000 layouts, and the test dataset consisted of 900 layouts. All designs used in the experiment were created using Intel 22nm technology libraries.

We explored the modeling parameters using a BO algorithm and selected the top five models by sorting them according to a score function. We evaluated the model performance as the average score for the five models. This evaluation method shows that the proposed model and training method provide robust results for data with a long-tail distribution without requiring an ML expert.

B. Evaluations for the HCR Model

We evaluated the predictive performance of the HCR model: regression performance using R2 and Pearson; classification performance using Accuracy and F1. In the regression performance, the R2 and Pearson for the test dataset were 0.70 and 0.71, respectively. The hierarchical inference flow helps filter out false alarms regarding poor routability from the regression results, enabling us to obtain robust regression performance even in imbalanced data. We also measured the classification performance by labeling the final prediction value; the class is *hotspot* if $\ddot{v}_{(x,y)} \geq \mathtt{DRV}_{th}$, and *non-hotspot* otherwise. Compared to the classification alone, the Accuracy and F1 scores increased by 0.03 and 0.11 for the test dataset. We also compared these results with those of previous studies [6], [7]. The HCR model has a 94% higher performance for F1 than [6]. This result means that the HCR model classifies the minority class well.

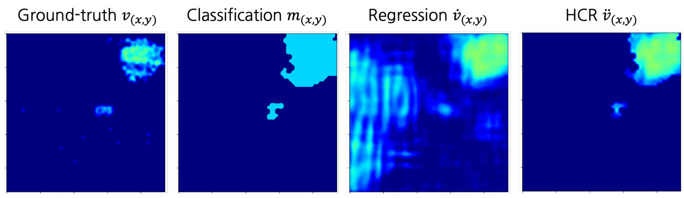

Fig. 3. An example of the routability prediction using the HCR model for a test design (Nova, utilization 0.9, aspect ratio 1.0, top layer m6, P&R clock 2.6 ns). The x and y axes represent the Gcell grid coordinates.

TABLE I
Comparison of the Evaluation Metrics with Previous Works

	Previous works		Ours
Metric	FCN [6]	J-Net [7]	HCR
Accuracy	0.94 (1.0)	0.93 (0.99)	**0.98** (1.04)
Recall	0.29 (1.0)	0.35 (1.23)	**0.79** (2.76)
Precision	0.76 (1.0)	0.67 (0.88)	**0.85** (1.11)
F1	0.42 (1.0)	0.46 (1.11)	**0.81** (1.94)
R2	NA	NA	**0.71**
Correl	NA	NA	**0.70**

IV. Conclusion

We proposed a routability prediction method using an HCR model. This model consists of a binary classification model and a regression model. The hierarchical structure improves the predictive performance for imbalanced data. We also proposed a training method for the HCR model, which optimizes the modeling parameters to maximize the classification performance and trains the regression model using transfer learning. We proved that this training method performs well, even for imbalanced datasets with a long-tail distribution. Compared to a previous study [6], we increased the classification performance by 4% in Accuracy and 94% in F1.

References

[1] A. B. Kahng, "Machine Learning Applications in Physical Design: Recent Results and Directions", *Proc. ISPD*, 2014, pp. 68-73.
[2] W. J. Chan, P.-H. Ho, A. B. Kahng, and P. Saxena, "Routability Optimization for Industrial Designs at Sub-14nm Process Nodes Using Machien Learning", *Proc. ISPD*, 2017.
[3] T.-C. Yu, S.-Y. Fang, H.-S. Chiu, K.-S. Hu, P. H.-Y. Hu, C. C.-F. Shen, and H. Sheng, "Pin Accessibility Prediction and Optimization With Deep-Learning-Based Pin Pattern Recognition," *IEEE Transactions on CAD*, 2020, pp. 2345-2356.
[4] J. Chen, J. Kuang, G. Zhao, D. J.-H. Huang and E. F.Y. Young, "PROS: A Plug-in for Routability Optimization applied in the State-of-the-art commercial EDA tool using deep learning", *Proc. ICCAD*, 2020, pp. 1-8.
[5] A. F. Tabrizi, N. K. Darav, L. Rakai, I. Bustany, A. Kennings and L. Behjat, "Eh?Predictor: A Deep Learning Framework to Identify Detailed Routing Short Violations From a Placed Netlist", *IEEE Transaction on CAD*, 2019, pp. 1177-1190.
[6] Z. Xie, Y.-H. Huang, G.-Q. Fang, H. Ren, S.-Y. Fang, Y. Chen and J. Hu, "RouteNet: Routability prediction for Mixed-Size Designs Using Convolutional Neural Network", *Proc. ICCAD*, 2018, pp. 1-8.
[7] R. Liang, H. Xiang, D. Pandey, L. Reddy, S. Ramji, G.-J. Nam, and J. Hu, "DRC Hotspot Prediction at Sub-10nm Process Nodes Using Customized Convolutional Network", *Proc. ISPD*, 2020, pp 135–142.
[8] Yang, Yuzhe and Xu, Zhi, "Rethinking the Value of Labels for Improving Class-Imbalanced Learning", *arXiv*, 2020.
[9] G. An, M. Akiba, K. Omodaka, T. Nakazawa and H. Yokota, "Hierarchical deep learning models using transfer learning for disease detection and classifcation based on small number of medical images", *Nature scientific*, 2021.
[10] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms", *arXiv*, 2012.
[11] C. Tan, F. Sun, T. Kong, W. Zhang, and C. Liu, "A survey on deep transfer learning.", *arXiv*, 2018.
[12] D. Kim, H. Kwon, S.-Y. Lee, S. Kim, M. Woo and S. Kang, "Machine Learning Framework for Early Routability Prediction with Artificial Netlist Generator", *Proc. DATE*, 2021, pp. 1809-1814.
[13] OpenCores: Open Source IP-Cores, http://www.opencores.org
[14] Cadence Innovus User Guide, https://www.cadence.com