Invited Paper: 2023 ICCAD CAD Contest Problem C: Static IR Drop Estimation Using Machine Learning

Gana Surya Prakash Kadagala and Vidya A. Chhabria
Arizona State University

***Abstract*—Power delivery network (PDN) analysis is a critical aspect of the design cycle to ensure the power grid meets the current demands of the chip. Static IR drop simulation, performed as a part of PDN analysis, is crucial to the estimation of the worstcase voltage drop (IR) drop of the chip which in turn determines chip frequency and functionality. Algorithmically, the static IR drop simulation amounts to solving a large system of linear equations with billions of variables and is computationally very expensive with significantly large runtimes. This contest aims at leveraging machine learning (ML) techniques to overcome this challenge. While previous research has introduced ML-based solutions for static IR drop simulation, their performance remains untested on standardized benchmarks, obscuring the true state-of-the-art ML model. The contest releases twenty real circuit (split as ten seen during training and ten as hidden) benchmarks and hundreds of synthetic benchmarks for training ML models to predict IR drop. The synthetic data serves as a training dataset, which can then be fine-tuned using limited real circuit data for accurate predictions on unseen testcases. The goal of the contest is to train novel ML algorithms to perform the prediction with high accuracy, F1 score, and low runtimes.**

I. Introduction

Power delivery network (PDN) analysis is crucial to successful IC design closure, particularly for designs implemented in lower technology nodes that suffer from large wire parasitics and high power densities. The on-chip PDN is responsible for transmitting voltages and currents to each cell in the design. However, due to the parasitics in the PDN, a voltage drop is induced between the power pads and the cells in the design. Large voltage drops in the PDN can hurt chip performance and, in the worst case, its functionality. Consequently, it is essential to check that the worst-case IR drop values in the PDN are within specified limits.

PDN analysis is typically performed several times in the design cycle. However, one of the major challenges with the analysis is the overhead of extremely large runtimes. The underlying algorithm simulates a network of conductances and current/voltage sources by solving a large system of equations with millions to billions of variables for static analysis. In modern industry designs, a single full-chip IR drop simulation can take several hours. Accelerating this analysis opens the door to optimizations that iteratively invoke these engines.

Prior acceleration methods such as [1] trade off accuracy for speed by increasing the coarseness of element discretization. However, the advent of machine learning (ML) has presented several fast and accurate solutions to this problems [2]–[7], trained on simulation data from systems of millions of nodes. However, none of these works, use an identical set of benchmarks, or datasets, and hence the state-of-the-art ML techniques for IR drop simulation are unknown. A part of the challenge has been the unavailability of a large public dataset to both train and test ML model. The only set of PDN benchmarks that are available to the public have been the IBM benchmarks released in 2008 which contain ten testcases which is insufficient to train large scale ML models. The recent release of a large synthetic public dataset [8], [9] could help alleviate this challenge through transfer learning. In this context, the contest has the following goals:

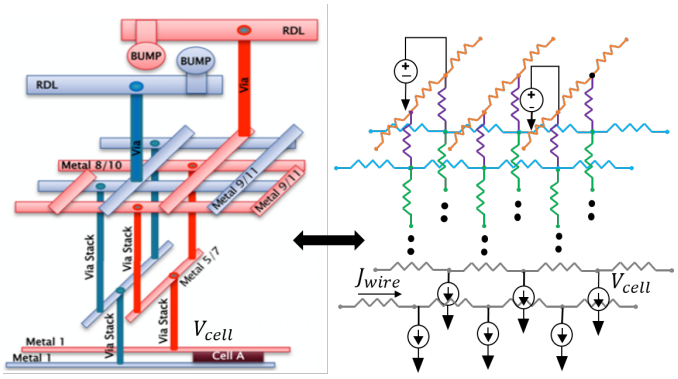
1) Lower the barriers to entry for non-EDA experts by converting a traditional EDA problem to an ML-solvable problem and incentivizing the use of novel ML techniques to improve accuracy.
2) Explore the use of transfer learning to address the limited dataset problem in the EDA community.
3) Release a set of PDN benchmarks and determine a state-of-the-art ML technique for IR drop estimation.

The contest releases a curated fake and real circuit dataset for PDN analysis including current distributions, resistances, and topologies of the power grid with voltage pad locations. In this contest, participants must use the provided benchmarks to train an ML model to predict static IR drop with the highest possible accuracy (mean absolute error) and F1 score on the test data with the least inference runtime and model size. The team with the best combination of these metrics as described in Section V wins the contest. The contest also provides a bonus to the winners who open-source their ML models.

II. Preliminaries

A structure of the on-chip PDN is shown in Fig. 1(a). The PDN can be modeled as a network of voltage sources, current sources, and resistances, where the wires are a network of resistances, the power pad (C4 bumps) are voltage sources connected to the PDN wires, and the current sources are the cells/instances that draw current as shown in Fig. 1(b).

The goal of IR drop simulation is to find the voltage at the nodes of the PDN that connect to the instances (current sources). Traditionally, finding the voltage at every node amounts to solving a system of linear equations of the form



(a) (b)

Fig. 1. PDN structure and its model

$G\mathbf{V} = \mathbf{J}$ [10], [11] where G is a conductance matrix, $\mathbf{V}$ is the unknown vector of voltages, and $\mathbf{J}$ is the vector of currents.

The static IR drop distribution across the chip depends on the following three factors:

1) The location/distributions of all voltage sources (power pads) in the design
2) The topology of the PDN and resistance values of each resistor (via/metal layer) in the network
3) The distribution of current (power) in the design

Prior ML techniques have modeled the above inputs as images [4]–[7] and used image-based ML models to predict IR drop. For example, [7] represents the above three features as three distributions where the current source distribution is modeled as a current map (Fig. 2(a)); the PDN topology is modeled as a function of the density (spacing between power stripes per unit area) of the power grid (Fig. 2(b)) in different regions, and the voltage source distribution is modeled as an effective distance map which is the distance from each PDN node to the PDN node with the voltage source (Fig. 2(c)). The output is a distribution of IR drop at every node in the lowermost layer of the PDN, which can be represented as an IR drop map (Fig. 2(d)). Using such image-based representations, prior techniques have used CNNs and U-Nets [12], to perform IR drop prediction using a model trained on a diverse set of such data points (three inputs and one output). The trained ML model can perform inference on unseen testcases.

III. Contest Problem Description

This section describes the inputs, outputs, and benchmarks used in the contest.

A. Inputs Description

The input data is provided in two formats:

1) Image-based inputs: These are direct image-based representations of the three inputs and the IR drop output, as described in Section 2. These can serve as a direct dataset for the application of image-related ML models (such as CNNs, U-Nets, etc.) encouraging participants who have limited knowledge of EDA to participate in the contest as a purely image-based ML contest.

TABLE I
Example current map matrix (CSV) where each value represents the total current in a 1μm $\times 1\mu$m region of the chip.

4.43E-08	5.83E-08	8.05E-08	1.02E-07	....	1.15E-07
7.93E-08	1.04E-07	1.44E-07	1.82E-07	....	2.05E-07
1.54E-07	2.03E-07	2.80E-07	3.54E-07	....	3.97E-07
2.71E-07	3.56E-07	4.92E-07	6.21E-07	....	6.95E-07
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
4.14E-07	5.45E-07	7.54E-07	9.53E-07	....	1.07E-06

2) SPICE-based inputs for the detailed structure of PDN and additional features: These are SPICE netlists that also provide the value of each resistor in the PDN resistance network. These inputs can be used to create graph-based ML models (such as GNN, GCNs, etc.). These inputs enable ML in EDA experts to use complex features such as resistances of metal layers as features.

Image-based data There are three inputs to determine static IR drop as described in Section II, including a current map, an effective distance to C4 bump map, and a PDN density map. The image-based data is provided as a matrix in a comma-separated value (CSV) file where every value in the file represents either the current, effective distance, or PDN density, in a 1μm $\times 1\mu$m area of the chip represented as a matrix (Note that the PDN node pitch in the lowermost layer will be larger). E.g.: A 80μm $\times 80\mu$m area chip will consist of 80×80 values in the current map CSV as shown in Table. I. The effective distance to C4 bump map, and PDN density map use an identical matrix CSV format and are also matrices of size 80×80.

SPICE-based data In addition to image-based data, for each datapoint we also provide the PDN model as shown in Fig. 1(b) as a SPICE netlist. The SPICE netlist encodes the node locations, the value of resistances between nodes, the current source nodes and their locations, and the voltage source nodes and locations. An example SPICE netlist snippet is below:

```
R645 n1_m1_108000_17920 n1_m1_102600_179200 0.14
R646 n1_m1_113400_179200 n1_M3_113400_179200 4.23
I7 n1_m1_113400_179200 0 4.24901e-08
V0 n1_m7_81000_106230 0 1.1
```

The generic description of the SPICE format is as follows

```
<electric component> <node1> <node2> <value>
```

where the node is defined with the following convention

```
<netname>_<layer-idx>_<x-cordinate>_<y-cordinate>
```

In the above example, $R646$ is via since they share the same x and y coordinate but the layer changes. $I7$ is a current source connected to the node at location $(113400, 179200)$, and $V0$ is the voltage source at node $(81000, 106230)$. The locations in the SPICE netlist can be converted to the CSV file by using a 2000 db units (dbu). For instance, in the above example, 113400 is location $113400/2000 = 56.7\mu$m. The SPICE netlist has all the information needed to generate the image-based representations through a Python conversion

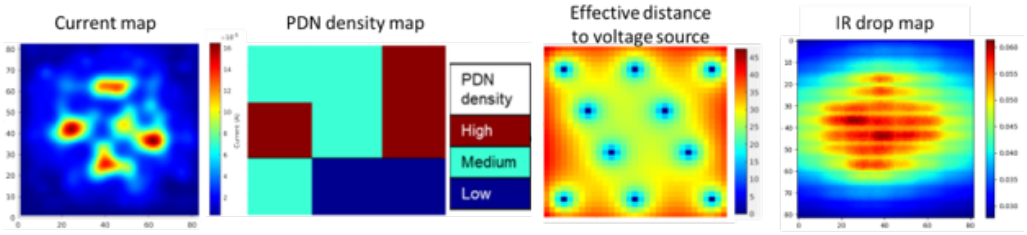


Fig. 2. Image-based representations of (a) input current map, (b) PDN density, (c) voltage sources, and (d) IR drop distributions across the die.

script. Further, the SPICE format also enables golden data voltage generation by simply running a SPICE simulation.

B. Output Description

The released benchmarks also consist of the output IR drop files used as golden data (labels) for training. The IR drop output is represented as an image in the matrix CSV format. The participants are required to generate the output as an image to ensure easy and fair evaluation between participants irrespective of the input format used (image-based or SPICE-based). The size of the output matrix is the same as the size of the input matrices and each pixel in the output image represents the worst-case IR drop in the 1μm $\times 1\mu$m region. These IR drop files are generated through a golden solver that evaluates $GV = J$ to solve for voltages. The matrix represents the IR drop values of those PDN nodes at the lowermost metal layer as these have the worst-case IR drop given the voltage pads are connected to the highest metal layer.

C. Contest Participant Objective

The contestants need to train an ML model to predict static IR drop with the highest possible accuracy (mean absolute error) and F1 score on the test data with the least inference runtime and model size. The evaluation will be performed on hidden testcases which are unseen during training.

IV. Benchmarks

A. Synthetic and Real Circuit Benchmarks

The contest releases a set of synthetic benchmarks from [8], and real circuit benchmarks from [13] for participants to train their ML models. The benchmarks are based on an open-source 45nm technology [14]. The synthetic current maps are created using a generative adversarial network as described in [9]. Power grids of different densities are synthesized on the generated current maps and golden IR drop numbers are generated by solving $GV = J$. The real-circuit data is generated from [13] where a given design is run through place and route. The PDN is extracted and the IR drops are computed in the same way as for the synthetic benchmarks by solving $GV = J$.

B. Diversity in Benchmarks

Each benchmark in the provided synthetic and real-circuit data varies in the area. Therefore, the size of the matrices can vary across the dataset as they represent chips of different sizes. However, a single datapoint will have all four matrices (current, PDN, voltage source, and IR drop) of the same size. In our dataset, each benchmark varies in the following ways:

- Current maps: These will vary in maximum, average, and peak values with different hotspot locations.
- PDN density maps: The PDN will be a region-wise uniform PDN where the density in a specific region is constant [15], [16]. Each region can use one of three possible densities as shown in Fig 2(b). Across the training and test dataset, the data points will either use a combination of the three PDN densities in different regions or a uniform power grid across all regions.
- Effective distance to voltage source maps [6]: These maps will vary in the number of voltage sources and their location across the different training and test benchmarks.

A sample from the benchmark dataset is shown in Fig. 3. The figure highlights two data points from the synthetic training dataset generated by [8] in the first two rows. Each datapoint consists of current maps (Fig. 3(a)), effective distance to voltage source maps(Fig. 3(b)), PDN density maps(Fig. 3(c)), and the golden IR drop map(Fig. 3(d)). The first row of the synthetic data represents current maps without macros and the second row represents current maps with macros. The figure also highlights one datapoint from the real circuit data that was visible to the contestants for training their model, in the second row, and a sample hidden testcase that the contestant's models are tested on, in the third row.

V. Contest Evaluation

The contestants will be provided with training (hundreds of fake data points and few real-circuit data points). However, the contestants' ML models will be evaluated on unseen (hidden) test dataset (real circuit data) and will be evaluated for:

Mean absolute error (MAE): The average of the absolute difference between a prediction and the actual value, is calculated for each example in a dataset. The goal is to have a low MAE which is the absolute difference between the predicted and original values of IR drop the matrix CSVs.

Below we describe a small example for computing each of these metrics. For a given golden IR drop matrix,

```
4.63E-03 5.23E-03
5.93E-03 0.04E-03
```

if the predicted IR drop matrix is

```
4.43E-03 5.83E-03
5.93E-03 1.04E-03
```

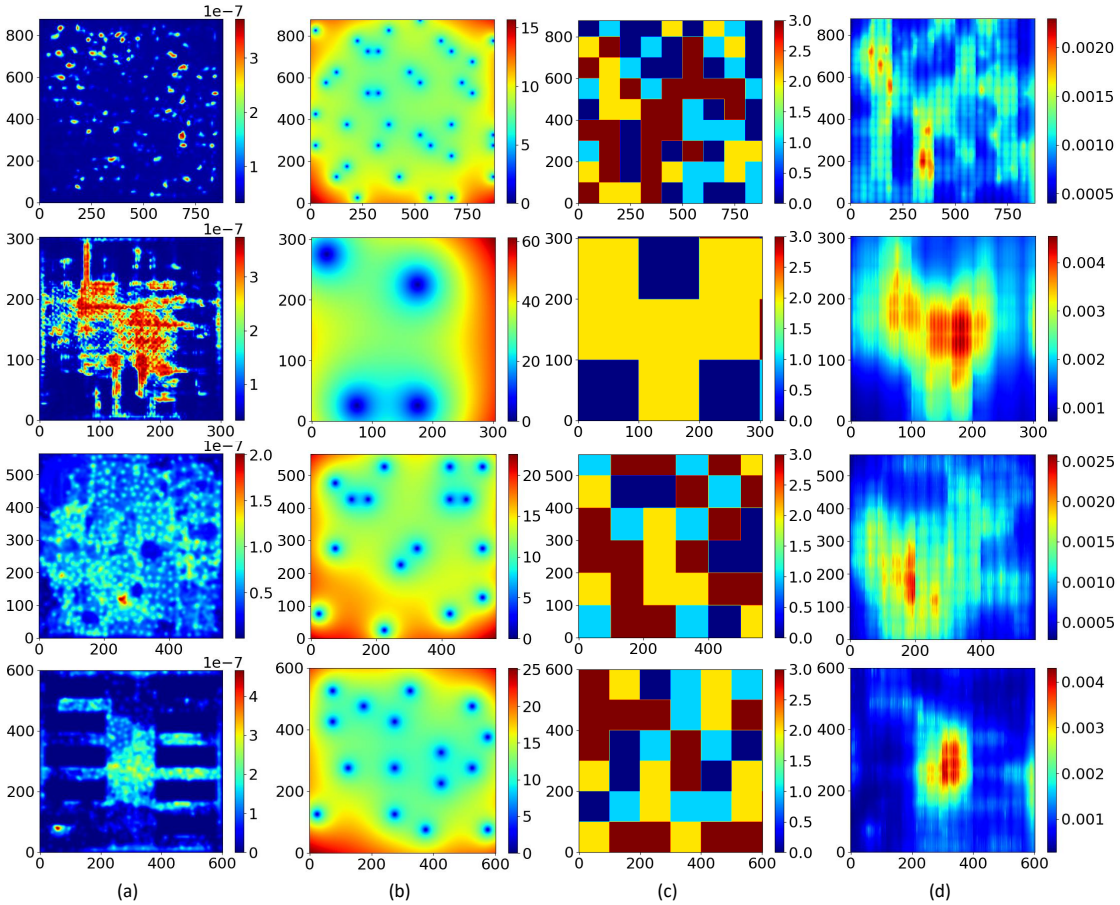


Fig. 3. The figure highlights sample benchmarks from our dataset. The four columns represent the (a) current maps (in Ampere), (b) the voltage source distribution, (c) PDN density in every region of the chip, and (d) IR drop map (in Volts). The first two rows show samples from our synthetic dataset, and the last two rows show a sample datapoint from the real circuit dataset.

then, the MAE matrix is given by

```
0.2E-03  0.6E-03
0.0      1.0E-03
```

The MAE value for the above testcase is the average of all values in the MAE matrix and for the example is 0.45mV.

F1 score: In addition to MAE, since the IR drop problem requires accurately estimating the regions with IR drop hotspots, we also use the F1 score as an evaluation metric. We represent the IR drop problem as a binary classification problem that uses 10% of the actual IR drop of each benchmark as a threshold to perform classification. The F1 score is given by:

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

where precision and recall are given by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

where TP is the number of true positive, FP is the number of false positives, TN is the number of true negatives, and FN is the number of false negatives. The positive class is PDN nodes with the top 10% of IR drop. The goal is to have a high F1 score ensuring that FNs are penalized. Each element in the predicted IR drop matrix is classified as an IR drop hotspot if its value is greater than 90% of the maximum IR of that testcase. For the running example in this section, an element that has an IR drop value larger than 90% of 5.93mV = 5.337mV is classified as an IR drop hotspot. This value is different for every testcase depending on the maximum IR drop value. Therefore, the golden hotspot matrix for the golden IR drop matrix is:

```
0  1
1  0
```

The predicted hotspot matrix for the predicted IR drop matrix:

```
0  0
1  0
```

Therefore, TP = 1; TN = 2; FP = 0; FN = 1;
Precision = TP/(TP + FP) = 1/1 = 1;
Recall = TP/(TP + FN) = 1/2 = 0.5;
F1 score = 1/1.5 = 0.67

Run time: Inference time of the ML model. The goal is to have low runtime and a fast inference.

Overall score: The overall score is a function of MAE, F1 score, and inference runtime. We will use a 60% weightage to

MAE and 30% to F1 and 10% to runtimes. Each participating team will be given a score for each metric. For each hidden testcase, the team with the best MAE will receive 60 points for that testcase, the team with the best F1 score will receive 30 points for that testcase, and the team with the best runtime will receive ten points for that testcase. The scores for the rest of the teams will be scaled based on a normal distribution for each metric (relative scoring). This will lead to a single score for each testcase (out of 100) for each team. The scores will be added across all hidden testcases and the top three teams with the highest will win the contest. Since there are 10 hidden testcases, the maximum score a team can get is 1000 if they have the best MAE, F1 score, and runtime across all testcases.

VI. Conclusion

The contest released synthetic and real circuit benchmarks for participants to train ML models to perform static IR drop prediction based on an open-source 45nm technology [14]. The benchmarks were released as both image-based and SPICE-based data to cater to both EDA experts and nonexperts. The contestants take part in the contest to train ML models to predict static IR drop to obtain the best accuracy and F1 score with the least runtimes.

References

[1] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "A multigrid-like technique for power grid analysis," *IEEE T. Comput. Aid. D.*, vol. 21, no. 10, pp. 1148–1160, Oct. 2002.

[2] S.-Y. Lin, Y.-C. Fang, Y.-C. Li, Y. Liu, T. Yang, S.-C. Lin, C.-M. J. Li, and E. J.-W. Fang, "IR drop prediction of ECO-revised circuits using machine learning," in *Proc. VTS*, 2018.

[3] C.-T. Ho and A. B. Kahng, "IncPIRD: Fast learning-based prediction of incremental IR drop," in *Proc. ICCAD*, 2019.

[4] Z. Xie, H. Ren, B. Khailany, Y. Sheng, S. Santosh, J. Hu, and Y. Chen, "PowerNet: Transferable dynamic IR drop estimation via maximum convolutional neural network," in *Proc. ASP-DAC*, 2020, pp. 13–18.

[5] V. A. Chhabria, Y. Zhang, H. Ren, B. Keller, B. Khailany, and S. S. Sapatnekar, "MAVIREC: ML-aided vectored IR-drop estimation and classification," in *Proc. DATE*, 2021, pp. 1825–1828.

[6] V. A. Chhabria, V. Ahuja, A. Prabhu, N. Patil, P. Jain, and S. S. Sapatnekar, "Thermal and IR drop analysis using convolutional encoder-decoder networks," in *Proc. ASP-DAC*, 2021, pp. 690–696.

[7] ——, "Encoder-decoder networks for analyzing thermal and power delivery networks," 2021.

[8] V. A. Chhabria, K. Kunal, M. Zabihi, and S. S. Sapatnekar, "BeGAN-benchmarks," 2021. [Online]. Available: https://github.com/UMN-EDA/BeGAN-benchmarks

[9] ——, "BeGAN: Power grid benchmark generation using a process-portable GAN-based methodology," in *Proc. ICCAD*, 2021.

[10] Y. Zhan, S. V. Kumar, and S. S. Sapatnekar, "Thermally-aware design," *Found. Trends Electron. Des. Autom.*, vol. 2, no. 3, pp. 255–370, Mar. 2008.

[11] Y. Zhong and M. D. F. Wong, "Fast algorithms for IR drop analysis in large power grid," in *Proc. ICCAD*, 2005, pp. 351–357.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*, 2015, pp. 234–241.

[13] T. Ajayi, V. A. Chhabria, M. Fogaça, S. Hashemi, A. Hosny, A. Kahng, M. Kim, J. Lee, U. Mallappa, M. Neseem, G. Pradipta, S. Reda, M. Saligane, S. Sapatnekar, C. Sechen, M. Shalan, W. Swartz, L. Wang, Z. Wang, M. Woo, and B. Xu, "Toward an open-source digital flow: First learnings from the OpenROAD project," in *Proc. DAC*, 2019, pp. 1–4.

[14] "NanGate 45nm FreePDK and cell library," 2022, https://si2.org/open-cell-library.

[15] V. A. Chhabria and S. S. Sapatnekar, "OpeNPDN: A neural-network-based framework for power delivery network synthesis," 2021.

[16] V. A. Chhabria, A. B. Kahng, M. Kim, U. Mallappa, S. S. Sapatnekar, and B. Xu, "Template-based PDN synthesis in floorplan and placement using classifier and CNN techniques," in *Proc. ASP-DAC*, 2020, pp. 44–49.