Fast IR Drop Estimation with Machine Learning

Invited Paper

Zhiyao Xie[†], Hai Li[†], Xiaoqing Xu[‡], Jiang Hu[§], Yiran Chen[†]

[†] Duke Univeristy, Durham, NC, USA

[‡] ARM Inc., Austin, TX, USA

[§] Texas A&M University, College Station, TX, USA

zhiyao.xie@duke.edu hai.li@duke.edu

ABSTRACT

IR drop constraint is a fundamental requirement enforced in almost all chip designs. However, its evaluation takes a long time, and mitigation techniques for fixing violations may require numerous iterations. As such, fast and accurate IR drop prediction becomes critical for reducing design turnaround time. Recently, machine learning (ML) techniques have been actively studied for fast IR drop estimation due to their promise and success in many fields. These studies target at various design stages with different emphasis, and accordingly, different ML algorithms are adopted and customized. This paper provides a review to the latest progress in ML-based IR drop estimation techniques. It also serves as a vehicle for discussing some general challenges faced by ML applications in electronics design automation (EDA), and demonstrating how to integrate ML models with conventional techniques for the better efficiency of EDA tools.

1 INTRODUCTION

IR drop, or voltage drop, is the deviation of a power supply level from its specification that occurs when current flows through power grids. It must be restricted in order for a circuit to meet its timing target and function properly. As design and manufacturing technologies advance, the increased current load further exaggerates IR drop violations, which become a critical concern for both VLSI design and test [33].

In order to meet IR drop constraints, designers need to estimate and mitigate IR drop throughout design stages from placement to signoff in multiple iterations. It may also be measured during post-silicon verification. Obtaining an accurate estimation of IR drop through simulation-based commercial tools is very time consuming [1, 12, 35]. Thus, IR drop mitigation guided by frequent IR drop simulations is computationally costly and hampers the overall design turnaround time. To speed up this process, a fast yet accurate IR drop estimator becomes a critical need.

In recent years, machine learning (ML) applications in electronics design automation (EDA) have started to attract wide attention. They have enabled fast estimation on many important metrics for chip design, including timing [2], power [16, 42], design rule violation [20, 34, 39], crosstalk [21], testability [26], lithography hotspots [37, 40], clock tree's quality [25], placement solution [27], routing solution [43], and IR drop [10, 12, 15, 23, 28, 30, 35, 36]. There have been many ML-based IR drop estimators targeting at various design stages with different emphasis. The majorities claim orders-of-magnitude acceleration compared with simulations-based solutions provided by widely-adopted commercial tools [1].

The ML-based IR drop estimators can be classified into two major categories, based on whether they estimate static IR drop or dynamic IR drop. Figure 1 shows a comparison between the average current and dynamic current under different frequencies [29]. The static IR drop analyses in most commercial tools only measure the average current

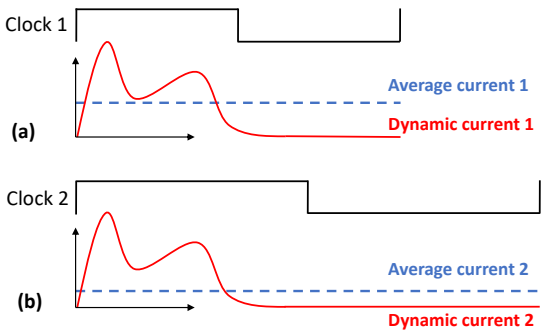

Figure 1: Static and dynamic analysis on current [29].

ICCAD '20, November 2–5, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8026-3/20/11…$15.00

https://doi.org/10.1145/3400302.3415763

Table 1: Comparison Among Different Works on IR drop estimation

Methods	Type of IR Drop	Time	ML Model	Cross-Design	Features	Objective
IncPIRD [15]	Static IR	2019	XGBoost	Yes	I_g, R_g, PDN, G	IR mitigation
XGBIR [30]		2020	XGBoost	-	I_g, R_g, PDN	PDN design
Yamato *et al.* [36]	Vector-based IR	2012	Linear Regression	No	P_c	IR-aware timing
Dhotre *et al.* [10]		2017	Clustering	Yes	r_{tog}, c	IR prediction
Lin *et al.* [23]		2018	ANN	No	P_c, R_c, t_c, r_{tog}, c	IR mitigation
Fang *et al.* [12]		2018	CNN, XGBoost	No	P_c, I_c, R_c, t_c, r_{tog}, c	IR mitigation
PowerNet [35]	Vectorless IR	2020	CNN	Yes	P_c, t_c, r_{tog}	IR mitigation
Mozaffari *et al.* [28]	Sillicon PSN	2019	ANN, CNN, NLP	Yes	r_{tog}, G	PSN prediction

drawn from power grids without considering switching activities [1, 3, 29]. It is widely used to identify the weakness of a power delivery network (PDN) at an early design stage when switching vectors are not available [1, 3]. There have been many traditional methods for fast power grids analysis [8, 17, 31, 32, 41, 44]. In contrast, dynamic IR drop captures the peak transient current value based on switching activities. Thus, it is a more strict constraint and more difficult to predict [22, 29]. The significant difference between static and dynamic IR drop leads to distinctive problem settings and corresponding ML solutions. We will introduce methods on static IR drop and dynamic IR drop separately.

Power supply noise (PSN) is sometimes also loosely referred to as IR drop [28]. But the PSN actually comprises both $L * di/dt$ and the dynamic IR drop [6]. The $L * di/dt$ component is an inductive effect caused by rapid current changes through power grids. Most works reviewed in this paper focus on the IR drop without considering $L * di/dt$.

In this paper, we will summarize the latest progress in ML-based techniques for both static and dynamic IR drop. The remainder of this paper is organized as follows. Section 2 presents an overview of the ML-based IR drop estimators. Section 3 covers these estimators in detail. In Section 4, we discuss some challenges and the integration of ML estimators into EDA tools.

2 METHOD OVERVIEW

2.1 Estimation on IR Drop

Table 1 summarizes the ML-based IR drop estimators from different perspectives. The criteria shown in Table 1 are introduced below.

2.1.1 Type of IR Drop. Different types of IR drop will result in largely different problem settings and ML solutions. Table 1 covers both the static and dynamic IR drop. For dynamic IR drop, the commercial simulators [1, 3] provide both vector-based and vectorless options. The vector-based analysis relies on switching vectors as input, which is limited by the late availability of switching activities in many design processes [1, 3, 35]. The vectorless option allows simulation by the toggling probability instead of switching vectors [3, 22]. All estimators on both static and dynamic IR drop use the simulation result from the commercial tool as the ground-truth label for training. One exception is Mozaffari *et al.* [28], which directly uses the PSN measured on silicon as their label. This method is also included in the comparison among dynamic IR drop estimators.

2.1.2 ML Model. These methods can be differentiated by their ML models. The frequently used ML algorithms include artificial neural network (ANN), XGBoost [7], and convolutional neural network (CNN) [18]. ANN and XGBoost typically process one-dimensional inputs, while CNN handles two-dimensional features with spatial information. In addition, the NLP model in Table 1 means neural network models used in natural language processing.

Here we briefly introduce these frequently applied ML models. ANN is also referred to as multilayer perceptron (MLP). It consists of multiple layers of nonlinearly-activating nodes. XGBoost is an efficient implementation of gradient boosting decision tree (GDBT) [13], which builds decision trees sequentially, with each tree built on the error of the previous ones. A typical structure of CNN is composed of convolutional layers, pooling layers, and fully-connected (FC) layers. Convolutional and pooling layers perform downsampling, and FC layers at the end generate the output.

2.1.3 Cross-Design. A cross-design ML estimator means the estimator applies to new designs that are not in the training set. Here we require the *new* design to be different from training designs at the netlist level. Thus the following examples are not viewed as cross-design: 1) Models trained and tested with different layout implementation of the same netlist. 2) Models trained and tested on multiple designs, which appear in both training and testing set.

2.1.4 Feature Selection. Features selection is one of the most critical steps in ML applications, which decides models' performance and efficiency. Table 1 provides a high-level summary of the feature types extracted in each method. Some straightforward features, like the location of cells, are not listed. These features are introduced below.

- *Resistance* $\{R_c, R_g\}$ - Resistance is a determining factor of IR drop. Some works directly use the total resistance measured on the path from power pad to each cell instance as input, denoted as R_c. Some others use different types of resistance measured on power grids and power nodes, denoted as R_g.
- *Current* $\{I_c, I_g\}$ - Current I is the other determining factor for IR drop. Like resistance, some works use the average or peak current measured on each cell instance as input, denoted as I_c. Some others use current loads or the total current on power grids, referred to as I_g.
- *Power* $\{P_c\}$ - Power dissipation is an important input in IR drop simulation. It correlates with current and is easier to simulate. The power consumption on cell instance is P_c. Power dissipation may include internal power, switching power, and leakage power.
- *Time* $\{t_c\}$ - The timing interval or window of switching for each cell instance. It comprises of estimated earliest and latest signal arrival time.
- *PDN* - The information about PDN. It includes geometric information about power grids, voltage source, and current loads. It also includes metal and via resistance on the PDN.
- *Toggling activity* $\{r_{tog}\}$ - The switching activity of each cell instance. It is usually measured by the toggle rate, the number of toggles of each cell divided by the number of cycles within a timing window. It can also represent the percentage of toggled flip-flops versus overall flip-flops.
- *Cell information* $\{c\}$ - The information about each cell instance except current, power and resistance. It includes area, cell load, cell type, etc.
- *Global information* $\{G\}$ - The information about the whole design or layout. It includes the process, voltage, temperature, frequency, and the size of the layout.

2.1.5 Objective. Besides estimating different types of IR drop with various input features, previous works also assume different application scenarios for their methods. Below are their objectives with their IR drop estimators.

- *Prediction* - Some works focus on estimation and do not give details on how the estimation will be applied.
- *Mitigation* - Most works estimate IR drop hotspots to guide mitigation. Mitigation solutions include power grid enhancement, cell or macro movement, decoupling cap insertion, etc.

Table 2: IR Drop-Related ML Works

Works	ML Model	Label to estimate
Lee *et al.* [19]	Special Neural Network	Weights under distortion
Chan *et al.* [4]	Boosting SVM	IR-aware timing
Liu *et al.* [24]	ANN, SVM	
Ye *et al.* [38]	SVM	
Chang *et al.* [5]	Gaussian process regression	Routing wirelength
Chhabria *et al.* [9]	CNN	Optimal PDN Template

- *PDN Design* - Some works focus on optimizing PDN structure with the fast IR drop estimation. This is similar to IR drop mitigation by power grid enhancing, but it usually takes multiple different PDN structures as candidates for evaluation.
- *Timing* - Some works estimate IR drop or PSN so that they can measure timing based on a more accurate supply voltage for each instance.

2.2 IR Drop-Related Estimation

Besides the IR drop estimators covered in Table 1, some other ML works consider IR drop without directly estimating it. Instead, they predict some design objectives related to IR-drop. A summary of IR drop-related ML works is provided in Table 2.

2.2.1 Impact of IR Drop. Instead of predicting IR drop, some works directly estimate the impact of it. The work in [19] predicts the distorted multiplication caused by the IR drop on ReRAM-based neural networks. Some other works directly estimate timing under the impact of IR drop and/or Ldi/dt, which is referred to as PSN-aware timing analysis or IR-aware timing analysis [4, 24, 38]. Notice the difference between [4, 24, 38] and [36]. The method in [36] estimates dynamic IR drop and uses the predicted IR drop to estimate timing, while models in [4, 24, 38] directly estimate IR drop-affected timing. They use completely different labels during training.

2.2.2 IR Drop as A Constraint. Some other works only view IR drop as one constraint and focus on other design objectives. For example, when designing PDN with machine learning models, works in [5, 9] optimize routing resources and PDN qualities while considering the IR drop and electromigration (EM) as constraints. As shown in Table 2, what they actually predict is the routing wirelength [5] or the optimal PDN template [9]. After that, their solutions are verified to meet IR drop specifications.

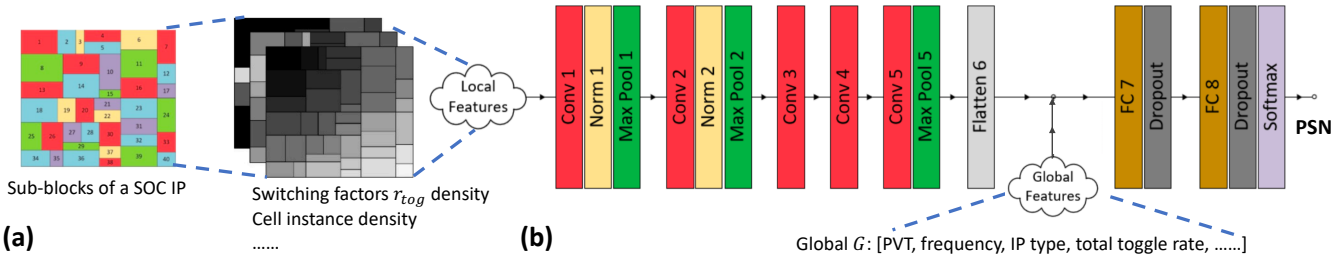

Figure 2: Mozaffari *et al.* (a) Generate density maps based on sub-blocks. (b) The CNN method with both local and global features. [28]

3 ESTIMATORS IN DETAIL

3.1 Static IR Drop Estimation

The two representative methods **IncPIRD** [15] and **XGBIR** [30] on static IR drop perform fast power grid analysis with ML models. Unlike works on dynamic IR drops, their features and labels are measured on the power nodes on PDN instead of on cells.

These two works [15, 30] share many similarities in terms of both features selection and the ML model. Table 3 and 4 describe the major features used in each work, respectively. They both use two major types of features: one is the PDN's topology, the other is electrical features. For electrical features, they both define features representing the *pullup* and *pulldown* for each analyzed node, based on the amount of current drawn and the effective resistance between each power node, voltage sources, and current loads. Lastly, they both adopt the XGBoost model.

Based on its XGBoost model, **IncPIRD** [15] proposes a flow to mitigate IR drop incrementally with many iterations.

Table 3: Description of Features of IncPIRD [15]

Category	Description of Feature on node n
Chip / PDN	Pitch of all metal layers Width / height of the chip
Electrical	*Pullup*: The effective resistance at n *Pulldown*: The symbolic IR drop at n Pullup and Pulldown of n's neighbors

Table 4: Description of Features of XGBIR [30]

Category	Description of Feature on node n
Chip / PDN	The number of power tracks Distance between n and boundary Power track segmental resistance
Electrical	*Pullup*: Voltage sources' impact on n *Pulldown*: Current loads' impact on n *V2I*: Resistance between voltage sources and current loads

In each iteration, based on the prediction, the designer performs incremental mitigation and generates an updated layout. To avoid retraining the ML model in each iteration, it defines a model update condition, determining whether the existing ML model still applies to the updated layout. The model is only retrained when testing data satisfies the update condition. Such an update condition actually considers the model's robustness quantitatively, which is rarely discussed in most ML applications in EDA.

3.2 Dynamic IR on Individual Cells

Yamato *et al.* [36] provide an intuitive method for dynamic IR drop estimation. According to their observation, most cells show a high correlation between their power and IR drop. It indicates that a simple linear model on power for each individual cell will perform well. Assuming m is the number of test patterns for training, each cell's training set consists of m power numbers as input and m corresponding IR drop labels. Then a linear model is trained for it.

Considering the huge number of cell instances in industrial designs, devoting a separate model for each cell can be costly in computation. Designers may not need such a fine granularity in practice. But the simplicity of linear models compensates the computation cost to a certain extent. In addition, one unique model for each cell instance may not generalize very well. If there is a small change in the placement or the PDN, the power-IR relations of some affected cells change accordingly; then the relevant models may need to be retrained.

The model for each individual cell ignores the impact of its neighboring cells. The IR drop at each cell instance depends on the current demand in a region enclosing the cell. For each region, all cells inside contribute to the overall current demand, which flows through a common metal path on PDN. Thus, all cells in a high IR drop region suffer from an excessive IR drop. To capture this, many works consider the IR drop in the granularity of regions or partitions instead of individual cells.

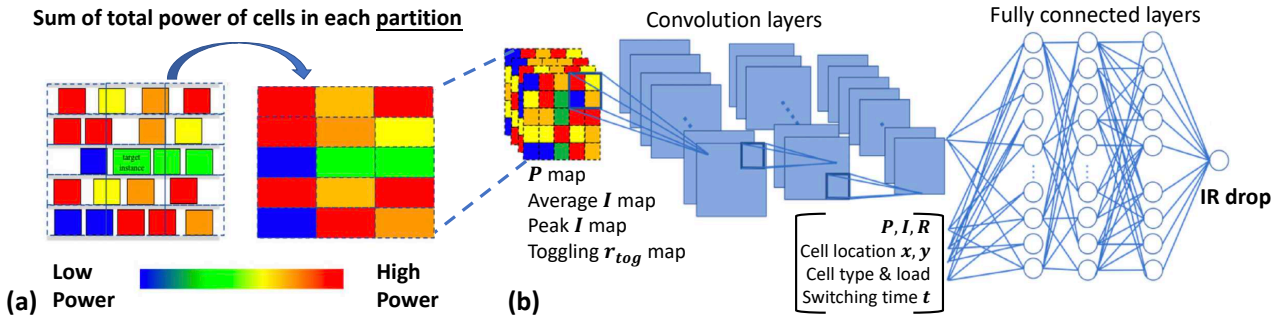

Figure 3: Fang *et al.* (a) Generate power maps around each cell. (b) ML model for each cell. [12]

3.3 Dynamic IR on Regions

Dhotre *et al.* [10] assign all cell instances to multiple clusters by clustering algorithms like k-means or DBSCAN [11]. Each cell's power is approximated by its toggling activity. All cells are clustered based on their locations, area, and toggling activity. Then the regions with high-toggling-activity clusters are viewed as both power and IR drop hotspots. This method considers locality information by clustering the neighboring cells with similar information. As an unsupervised algorithm, it applies to different designs.

Mozaffari *et al.* [28] adopt more complex models. Figure 2 shows their method with CNN structure. For each test pattern, the input includes the information from the whole layout, and accordingly, the label is the PSN value measured on the chip. Hence, this method is very coarse-grained. In Figure 2(a), it builds two-dimensional 'local features' with detailed information from each sub-block. The features include togging rate density and cell density. It also builds a vector with the chip-level global information and total toggling rate, representing 'global features'. In Figure 2(b), the two-dimensional 'local features' firstly go through convolutional layers, then the flattened output is concatenated with one-dimensional 'global features' for the fully-connected layers. In this way, the CNN model learns from features in different dimensions, from both local and global perspectives. By incorporating global features of the whole chip, it claims that this method applies to new chips, which means the method is cross-design.

Dhotre *et al.* [10] and **Mozaffari *et al.*** [28] consider the localized power demand by estimating IR drop in regions or partitions. However, as introduced in Figure 1, the dynamic IR drop captures the worst transient IR drop. The high-current-consumption cells in the same region may not switch simultaneously, which does not result in a transient peak in IR drop. It means models may give false alarms on regions with high power consumption if they do not consider the signal arrival time of cells.

3.4 Dynamic IR with Timing

Fang *et al.* [12] estimate the IR drop of each cell by considering both neighboring cells' current consumption and the switching timing window of cells. The timing window of each cell includes the minimum and maximum signal arrival time. Figure 3(a) shows how it generates feature maps for each cell. The entire layout is tessellated into an array of small partitions. Then the values of power, peak current, average current, and toggling rate within each partition are calculated. For each cell, to quantify the impact from neighboring cells in the local region, features maps are constructed based on the 21×11 local partitions around this cell. These maps are the input of the CNN model. In addition, all the available information about the estimated cell is stacked into a vector, including its power, current, resistance, switching time, location, etc. This vector is provided through FC layers, as shown in Figure 3(b).

When comparing Figure 3 to Figure 2, **Mozaffari *et al.*** [28] and **Fang *et al.*** [12] use highly similar CNN architectures to incorporate both two-dimensional and one-dimensi onal inputs. The two-dimensional inputs in both works describe the distribution of power or toggling activity. The difference is that one of them estimates the IR drop on each cell, while the other estimates that on the entire layout. Due to such difference, the vector provided to FC layers in Figure 3(b) describes each cell, while in Figure 2(b) it provides global features about the entire chip.

One significant difference between **Fang *et al.*** [12] and the aforementioned methods is the employment of instances' switching time window. It enables this model to capture simultaneous switchings among cells. However, the switching time and cell locations themselves do not directly correlate with IR drop. Providing them directly to FC layers may overfit the ML model to the training data, which potentially limits the generalization of the model. Instead of being cross-design, this model infers the same design in training, with merely cell location changes.

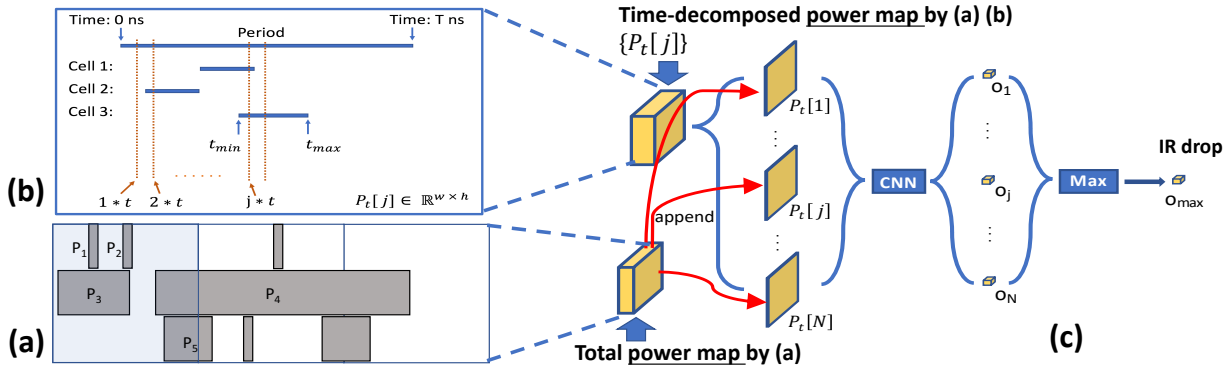

Figure 4: PowerNet method. (a) Generate power maps around each grid. (b) Timing-decomposed power map. (c) ML model for each grid. [35]

3.5 Cross-Design Dynamic IR with Timing

PowerNet [35] chooses a different direction to utilize timing information. Instead of directly using switching time as the input, it incorporates the switching time of cells into multiple power distribution maps, or named power maps.

In Figure 4(a), the layout is firstly tessellated into an array of uniform grid tiles, then the internal, switching, and leakage power are measured for each tile. For each tile, its input features include the power of the 30×30 local tiles around it, denoted as 'total power map' in Figure 4(c). This process is similar to the feature generation scheme in **Fang *et al.*** [12]. Besides that, power maps with timing information are constructed in Figure 4(b). It divides one clock cycle into N different time instants and measures N corresponding time-decomposed power maps $P_t[1] - P_t[N]$. For each instant j, only the cells that can possibly switch at that instant contribute their power consumption to $P_t[j]$. Thus, the power map $P_t[j]$ only measures the dynamic power consumed at the j's instant.

Based on the time-decomposed power maps $P_t[1] - P_t[N]$ corresponding to N instants, **PowerNet** [35] captures the one that leads to the highest transient IR drop at each grid. To achieve this, the same CNN model processes all N time-decomposed power maps in parallel, as shown in Figure 4(c), each output corresponds to the transient IR drop at that instant. Then the maximum among N outputs is the final estimation. Notice the 'total power map' without timing information is also used as input together with every time-decomposed power map $P_t[j]$ for each instant j.

By devoting a time-decomposed power map for each instant, **PowerNet** [35] uses a CNN model with the maximum structure to capture both localized power demand and simultaneous switching activities. Both of them lead to dynamic IR drop hotspots. But as a trade-off, compared with a regular CNN model, processing N power maps in parallel also leads to higher computation cost.

4 RETHINKING ML FOR IR DROP

4.1 Challenges and Future Directions

The ML methods have achieved significant progress in IR drop estimation. Besides further improving the models with more advanced techniques, here we discuss three more general challenges that can be addressed in the future. They may also apply to other ML applications in EDA.

4.1.1 Evaluation and Comparison. We have discussed many estimators on both static and dynamic IR drops. But it is difficult to make a fair comparison among them and figure out the best solution in different application scenarios. Although there are already many open-sourced benchmarks, researchers synthesize these designs with different design parameters and technology libraries, which result in distinct training and testing data. This can be addressed by an open-source benchmark suite dedicated to ML applications on multiple design objectives. It will enable rapid and clear comparisons among different methods, ensure high-quality training data, relieve researchers from data generation, and attract people from the ML community to contribute.

4.1.2 Model Development and Maintenance. Most aforementioned IR drop estimators are tuned manually in terms of both feature selection and model architecture on a specific dataset. Later on, when the training dataset is updated, or the application scenario is adjusted, the ML model may need to be fine-tuned or even re-developed from scratch. The development and maintenance of every single ML model heavily rely on human expertise. As ML estimators gain popularity in EDA, the high engineering cost on development and maintenance will grow accordingly. This can be addressed by algorithms like AutoML [14], which automatically search the optimal features combination and ML model structure for any given dataset.

4.1.3 Model Robustness. The aforementioned IR drop estimators are verified to be accurate based on their own application scenarios. However, the estimator is not likely to perform well on every test case it sees, especially for those cross-design models. This is because designs differ greatly from each other in terms of architecture, technology node, design flow, etc. Hence, it can be risky every time when the estimator infers a brand new test case. To address this, we consider developing an algorithm to measure how likely the model will perform well on a new test case. This can be achieved by quantifying the similarities between the test case and training data. It bears similarity to the idea of 'update condition' in IncPIRD [15].

4.2 Integrating ML Model into EDA Tools

Most of the IR drop estimators we introduced are standalone studies. Based on existing ML models with good performance, an interesting topic is how to integrate the models with existing EDA techniques to obtain more efficient EDA tools.

4.2.1 Early Report. For an ML-integrated EDA tool, the most straightforward application is to allow queries on predicted design objectives like IR drop at very early design stages. In addition, EDA tools can enable IR drop-aware timing analysis based on predicted IR drop, as proposed by Yamato *et al.* [36].

4.2.2 Objective-Oriented Design Guidance. ML models can provide guidance for each individual design stage at a low cost. For example, IR estimators may enable an IR drop-aware design flow, which allows designers to address IR drop at all necessary design stages, from placement to signoff. In this case, mitigation by cell movement at IR drop hotspots can happen as early as placement, and the power strap enhancement can be done during PDN design in the early iterations. This would largely reduce the number of design iterations compared with addressing IR drop at the signoff stage. Some of these IR-drop-aware methods may already be available in the backend tools, but they may be expensive to use without ML models as their IR drop estimators.

4.2.3 Design Flow Guidance. ML models can guide decisions in design flows by their early feedback. ML-integrated tools may allow a fast sweep over multiple design flow settings. In modern industrial chip design, the design flow tuning requires to go through a large amount of parameter setting, and the evaluation of every single trial is costly. ML estimators can evaluate candidate design flows at an early stage, then prune away the flows predicted to be worse than a certain threshold. Such evaluation may require multiple ML models on different design objectives.

4.2.4 Model Training. When starting on a new design, the prediction may be based on some cross-design ML methods trained on other similar designs. After a few design iterations, based on collected features and labels, the tool may build a non-cross-design model dedicated to this specific design. As iteration continues, the model may be updated by retraining, or incrementally improved by fine-tuning or ensemble learning.

5 CONCLUSION

In this paper, we summarize the latest progress in developing ML models for fast IR drop estimation. We introduce the innovations and technical details of representative methods in two main categories, static and dynamic IR drop. In addition, we discuss some general challenges in ML estimators and how they may be better integrated into current EDA tools.

REFERENCES

[1] Ansys. 2020. Ansys RedHawk: power noise and reliability sign-off solution. https://www.ansys.com/products/semiconductors/ansys-redhawk

[2] Erick Carvajal Barboza, Nishchal Shukla, Yiran Chen, and Jiang Hu. 2019. Machine learning-based pre-routing timing prediction with reduced pessimism. In *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

[3] Cadence. 2020. Voltus IC Power Integrity Solution User Guide. http://www.cadence.com

[4] Wei-Ting J Chan, Kun Young Chung, Andrew B Kahng, Nancy D MacDonald, and Siddhartha Nath. 2016. Learning-based prediction of embedded memory timing failures during initial floorplan design. In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 178–185.

[5] Wen-Hsiang Chang, Chien-Hsueh Lin, Szu-Pang Mu, Li-De Chen, Cheng-Hong Tsai, Yen-Chih Chiu, and Mango C-T Chao. 2017. Generating routing-driven power distribution networks with machine-learning technique. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36, 8 (2017), 1237–1250.

[6] Howard H Chen and David D Ling. 1997. Power supply noise analysis methodology for deep-submicron VLSI chip design. In *Proceedings of the 34th annual Design Automation Conference*. 638–643.

[7] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (KDD)*. ACM.

[8] Tsung-Hao Chen and Charlie Chung-Ping Chen. 2001. Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods. In *Proceedings of the 38th annual Design Automation Conference*. 559–562.

[9] Vidya A Chhabria, Andrew B Kahng, Minsoo Kim, Uday Mallappa, Sachin S Sapatnekar, and Bangqi Xu. 2020. Template-based PDN Synthesis in Floorplan and Placement Using Classifier and CNN Techniques. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 44–49.

[10] Harshad Dhotre, Stephan Eggersglüß, and Rolf Drechsler. 2017. Identification of efficient clustering techniques for test power activity on the layout. In *2017 IEEE 26th Asian Test Symposium (ATS)*. IEEE, 108–113.

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[12] Yen-Chun Fang, Heng-Yi Lin, Min-Yan Sui, Chien-Mo Li, and Eric Jia-Wei Fang. 2018. Machine-learning-based dynamic IR drop prediction

for ECO. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–7.

[13] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[14] Xin He, Kaiyong Zhao, and Xiaowen Chu. 2019. AutoML: A Survey of the State-of-the-Art. *arXiv preprint arXiv:1908.00709* (2019).

[15] Chia-Tung Ho and Andrew B Kahng. 2019. IncPIRD: Fast Learning-Based Prediction of Incremental IR Drop. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–8.

[16] Donggyu Kim, Jerry Zhao, Jonathan Bachrach, and Krste Asanović. 2019. Simmani: Runtime Power Modeling for Arbitrary RTL with Automatic Signal Selection. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. 1050–1062.

[17] Joseph N Kozhaya, Sani R Nassif, and Farid N Najm. 2002. A multigrid-like technique for power grid analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 21, 10 (2002), 1148–1160.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[19] Sugil Lee, Giju Jung, Mohammed Fouda, Jongeun Lee, Ahmed Eltawil, and Fadi Kurdahi. 2020. Learning to Predict IR Drop with Effective Training for ReRAM-based Neural Network Hardware. In *2020 Design Automation Conference (DAC)*. IEEE.

[20] Rongjian Liang, Hua Xiang, Diwesh Pandey, Lakshmi Reddy, Shyam Ramji, Gi-Joon Nam, and Jiang Hu. 2020. DRC Hotspot Prediction at Sub-10nm Process Nodes Using Customized Convolutional Network. In *Proceedings of the 2020 International Symposium on Physical Design*. 135–142.

[21] Rongjian Liang, Zhiyao Xie, Jinwook Jung, Vishnavi Chauha, Yiran Chen, Jiang Hu, Hua Xiang, and Gi-Joon Nam. 2020. Routing-Free Crosstalk Prediction. In *2020 International Conference on Computer-Aided Design (ICCAD)*.

[22] Shen Lin, Makoto Nagata, Kenji Shimazake, Kazuhiro Satoh, Masaya Sumita, Hiroyuki Tsujikawa, and Andrew T Yang. 2004. Full-chip vectorless dynamic power integrity analysis and verification against 100uV/100ps-resolution measurement. In *Proceedings of the IEEE 2004 Custom Integrated Circuits Conference*. IEEE, 509–512.

[23] Shih-Yao Lin, Yen-Chun Fang, Yu-Ching Li, Yu-Cheng Liu, Tsung-Shan Yang, Shang-Chien Lin, Chien-Mo Li, and Eric Jia-Wei Fang. 2018. IR drop prediction of ECO-revised circuits using machine learning. In *2018 IEEE 36th VLSI Test Symposium (VTS)*. IEEE, 1–6.

[24] Yu-Cheng Liu, Cheng-Yu Han, Shih-Yao Lin, and James Chien-Mo Li. 2016. PSN-aware circuit test timing prediction using machine learning. *IET Computers & Digital Techniques* 11, 2 (2016), 60–67.

[25] Yi-Chen Lu, Jeehyun Lee, Anthony Agnesina, Kambiz Samadi, and Sung Kyu Lim. 2019. GAN-CTS: A generative adversarial framework for clock tree prediction and optimization. In *2019 International Conference on Computer-Aided Design (ICCAD)*. 1–8.

[26] Yuzhe Ma, Haoxing Ren, Brucek Khailany, Harbinder Sikka, Lijuan Luo, Karthikeyan Natarajan, and Bei Yu. 2019. High performance graph convolutional networks with applications in testability analysis. In *2019 Design Automation Conference (DAC)*. 1–6.

[27] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Sungmin Bae, et al. 2020. Chip Placement with Deep Reinforcement Learning. *arXiv preprint arXiv:2004.10746* (2020).

[28] Seyed Nima Mozaffari, Bonita Bhaskaran, Kaushik Narayanun, Ayub Abdollahian, Vinod Pagalone, Shantanu Sarangi, and Jonathon E Colburn. 2019. An Efficient Supervised Learning Method to Predict Power Supply Noise During At-speed Test. In *2019 IEEE International Test Conference (ITC)*. IEEE, 1–10.

[29] SK Nithin, Gowrysankar Shanmugam, and Sreeram Chandrasekar. 2010. Dynamic voltage (IR) drop analysis and design closure: Issues and challenges. In *2010 11th International Symposium on Quality Electronic Design (ISQED)*. IEEE, 611–617.

[30] Chi-Hsien Pao, An-Yu Su, and Yu-Min Lee. 2020. XGBIR: an xgboost-based IR drop predictor for power delivery network. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1307–1310.

[31] Haifeng Qian, Sani R Nassif, and Sachin S Sapatnekar. 2005. Power grid analysis using random walks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 24, 8 (2005), 1204–1224.

[32] Haihua Su, Emrah Acar, and Sani R Nassif. 2003. Power grid reduction based on algebraic multigrid principles. In *Proceedings 2003. Design Automation Conference (IEEE Cat. No. 03CH37451)*. IEEE, 109–112.

[33] Mohammad Tehranipoor and Kenneth M Butler. 2010. Power supply noise: A survey on effects and research. *IEEE Design & Test of Computers* 27, 2 (2010), 51–67.

[34] Zhiyao Xie, Yu-Hung Huang, Guan-Qi Fang, Haoxing Ren, Shao-Yun Fang, Yiran Chen, and Jiang Hu. 2018. RouteNet: Routability prediction for mixed-size designs using convolutional neural network. In *2018 International Conference on Computer-Aided Design (ICCAD)*. 1–8.

[35] Zhiyao Xie, Haoxing Ren, Brucek Khailany, Ye Sheng, Santosh Santosh, Jiang Hu, and Yiran Chen. 2020. PowerNet: Transferable Dynamic IR Drop Estimation via Maximum Convolutional Neural Network. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 13–18.

[36] Yuta Yamato, Tomokazu Yoneda, Kazumi Hatayama, and Michiko Inoue. 2012. A fast and accurate per-cell dynamic IR-drop estimation method for at-speed scan test pattern validation. In *2012 IEEE International Test Conference (ITC)*. IEEE, 1–8.

[37] Haoyu Yang, Luyang Luo, Jing Su, Chenxi Lin, and Bei Yu. 2017. Imbalance aware lithography hotspot detection: a deep learning approach. *Journal of Micro/Nanolithography, MEMS, and MOEMS* 16, 3 (2017), 033504.

[38] Fangming Ye, Farshad Firouzi, Yang Yang, Krishnendu Chakrabarty, and Mehdi B Tahoori. 2014. On-chip voltage-droop prediction using support-vector machines. In *2014 IEEE 32nd VLSI Test Symposium (VTS)*. IEEE, 1–6.

[39] Cunxi Yu and Zhiru Zhang. 2019. Painting on placement: Forecasting routing congestion using conditional generative adversarial nets. In *2019 Design Automation Conference (DAC)*. 1–6.

[40] Hang Zhang, Bei Yu, and Evangeline FY Young. 2016. Enabling online learning in lithography hotspot detection with information-theoretic feature optimization. In *Proceedings of the 35th International Conference on Computer-Aided Design*. 1–8.

[41] Min Zhao, Rajendran V Panda, Sachin S Sapatnekar, and David Blaauw. 2002. Hierarchical analysis of power distribution networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 21, 2 (2002), 159–168.

[42] Yuan Zhou, Haoxing Ren, Yanqing Zhang, Ben Keller, Brucek Khailany, and Zhiru Zhang. 2019. PRIMAL: Power Inference using Machine Learning. In *Proceedings of the 56th Annual Design Automation Conference 2019*. 1–6.

[43] Keren Zhu, Mingjie Liu, Yibo Lin, Biying Xu, Shaolan Li, Xiyuan Tang, Nan Sun, and David Z Pan. 2019. GeniusRoute: A New Analog Routing Paradigm Using Generative Neural Network Guidance.. In *ICCAD*. 1–8.

[44] Cheng Zhuo, Jiang Hu, Min Zhao, and Kangsheng Chen. 2008. Power grid analysis and optimization using algebraic multigrid. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27, 4 (2008), 738–751.