

Received December 1, 2021, accepted January 6, 2022, date of publication January 12, 2022, date of current version January 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3142330

Fast and Accurate Machine Learning Compact Models for Interconnect Parasitic Capacitances Considering Systematic Process Variations

MOHAMED SALEH ABOUELYAZID^{ID1,2}, (Member, IEEE), SHERIF HAMMOUDA¹, AND YEHEA ISMAIL^{ID2}, (Fellow, IEEE)

¹Siemens EDA, Cairo 11835, Egypt

²Department of Electronics and Communications Engineering, School of Science and Engineering, The American University in Cairo, Cairo 11835, Egypt

Corresponding author: Mohamed Saleh Abouelyazid (mohamed_saleh@mentor.com)

ABSTRACT A novel modeling methodology is developed for interconnect parasitic capacitances in rule-based extraction tools. Traditional rule-based extraction tools rely on pattern matching operations to match every interconnect structure with corresponding pre-characterized capacitance formulas. Such a method suffers from three main problems including limited pattern coverages, potential pattern mismatches, and limited handling of systematic process variations. These problems prohibit rule-based methods from coping with the new capacitance extraction accuracy requirements in advanced process nodes. The proposed methodology overcomes these problems by providing machine learning compact models for interconnect parasitic capacitances that cover varieties of realistic cross-section metal patterns. Those models efficiently include the impact of systematic process variations on parasitic capacitances. Moreover, each model can handle thousands of patterns replacing thousands of existing capacitance formulas. The input to the models is a cross-section pattern that is represented by a novel vertex-based pattern representation. The models are implemented using two different machine learning methods: neural networks and support vector regressions. The two methods are tested and compared to each other. The proposed methodology is tested over thirteen test chips of 28nm, 14nm, and 7nm process nodes with more than 6.7M interconnect cross-section patterns. The results show that the proposed methodology provided outstanding accuracy as compared to field-solvers and rule-based models with an average error $< 0.15\%$ and a standard deviation $< 3.3\%$, whereas the average errors and standard deviations of rule-based models exceed 6%, for the same test chips. Also, the computational runtimes of the compact models are almost 2.5X faster than rule-based models.

INDEX TERMS Machine learning, process variations, parasitic capacitance, parasitic extraction.

I. INTRODUCTION

During the past decades, the semiconductor industry has developed considerably. There is a continuous increase in market demand to integrate more functionalities together on a single chip at a much lower cost and higher speed. Such an increasing demand motivated process technology nodes to scale down in a continuous manner. Therefore, the density of integrated circuits keeps increasing, and the dimensions of metal wires (i.e., interconnects) keep decreasing from one technology generation to the next. This resulted in an increase in the impact of interconnect parasitic elements on

chip performances, which is one of the major problems in advanced process technology nodes [1]–[4].

Interconnect parasitic elements represent the unintended passive circuit elements, such as resistors, capacitors, and inductors, that are not included in original circuit designs but exist in final chips. Such parasitic elements are associated with circuit routes (i.e., interconnects) that connect circuit devices together. The impact of such parasitic elements on circuit performances keeps increasing from one technology generation to the next. In recent advanced process technology nodes, the impact of interconnect parasitic elements on integrated circuits significantly increased degrading the overall circuit performances.

A layout parasitic extraction is an essential step in integrated circuit (IC) design flows. It is used to extract the

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Asif^{ID}.

parasitic elements of a given layout and associate the extracted parasitic elements with the corresponding circuit's netlist (i.e., parasitic netlist). The parasitic netlist is later used to perform a post-layout simulation in order to verify the performance of the corresponding layout. In case of any violation in post-layout simulation results, the layout designer would adjust his layout, re-extract its parasitic elements, and re-simulate it. Such a process is repeated until the simulation results meet the required circuit specifications. In other words, the current IC design flow requires multiple parasitic extraction and simulation runs in order to meet the required circuit specifications. Therefore, any inaccuracy in extracted parasitic elements would generate misleading post-layout simulation results. Such misleading results would degrade the yield and increase the turn-around time of a circuit design. Moreover, in advanced process technology nodes, the accuracy requirements of the parasitic capacitance extraction significantly increased (< 5% error). This increase added more challenges on parasitic capacitance extraction tools in order to meet such new requirements [1], [5], [6]. As a result, more accurate parasitic extraction tools are required to help circuit designers in developing more efficient layouts that meet the required specifications [1], [2], [4], [5].

There are two main parasitic capacitance extraction methods including field-solver and rule-based extraction methods. Field-solvers provide very accurate parasitic capacitance results relative to measurements; however, they are very slow and have a limited capacity [7]. Field-solvers mainly use numerical methods to perform electrostatic (or electromagnetic) simulations over a given layout. This is done by solving Maxwell equations across the entire layout domain using any of computational methods such as finite difference (FDM), finite element (FEM), and boundary element (BEM) methods. On the other hand, rule-based extraction methods, also known as 2.5D extraction methods, are way faster than field-solvers, and they can handle full chips with a reasonable accuracy. Rule-based extraction methods use pattern matching operations to match every layout pattern with corresponding pre-characterized analytical or empirical parasitic capacitance formulas that are stored in a database (i.e., library) of pre-characterized formulas [1], [6], [8], [9].

The current rule-based methods cannot cope with the new parasitic capacitance extraction accuracy requirements in advanced process technology nodes (< 5% error) [1], [6]. To improve the accuracy of rule-based extraction methods, one solution is to extract the parasitic capacitances of complicated and problematic layout structures using a field-solver; however, this is not a sustainable solution because the efficiency of existing rule-based methods is decreasing from one technology generation to the next, and the size of layout designs keeps increasing. Therefore, more layout patterns would be extracted by field-solvers impacting the performance and the capacity of parasitic capacitance extraction processes. As a result, there is a strong need to improve the accuracy of rule-based parasitic capacitance extraction models in order to cope with the new accuracy requirements

and handle the complicated and denser layout designs in advanced process nodes [1], [2], [10], [11].

The current rule-based extraction methods have three main problems including 1- a limited pattern coverage, 2- potential pattern mismatches, and 3- a limited handling of systematic process variations. With regards to the limited pattern coverage, the current rule-based extraction tools rely on limited pre-characterized layout patterns. Such patterns are generated using a limited number of geometrical parameters, such as widths and spacings, that are used to create corresponding parasitic capacitance formulas. Such formulas cannot cope with the complicated layout patterns, with arbitrary distributed polygons, in recent layout designs as they do not have enough geometrical parameters to accurately represent such patterns. Therefore, detailed and multi-dimensional models are required to capture all required geometrical parameters that impact parasitic capacitances in a certain layout pattern.

Regarding the potential pattern mismatch, it means that parasitic capacitances of a certain layout pattern are extracted using inappropriate capacitance formulas. This results in extracting wrong parasitic capacitance values. There is a tradeoff between pattern coverages and pattern mismatches, where increasing the number of pre-characterized patterns increases the probability of pattern mismatches.

As for systematic process variations, they represent physical variations in layout interconnects and devices. Such variations are layout-dependent, and they mainly occur during layout manufacturing processes. The most common systematic variations of interconnects include metal thickness variations, loading effects (i.e., inter layer dielectric thickness variations), metal width variations (e.g., etching), and trapezoidal variations (i.e., sidewall slope of metals). The impact of such variations on parasitic capacitances significantly increased in advanced process nodes, where the dimensions of metal wires are smaller, and systematic variations started to represent considerable portions of metal dimensions. Therefore, layout parasitic capacitance extraction processes must consider systematic process variations in order to provide accurate parasitic netlists [5], [12].

The current rule-based extraction tools handle the impact of systematic process variations on parasitic capacitances independently using sensitivity formulas that represent the sensitivity of a certain capacitance component to a certain variation parameter. Such formulas are pre-characterized with limited geometrical parameters [13]–[15]. Therefore, they also suffer from potential pattern mismatch and limited pattern coverage problems. To consider systematic process variations during the parasitic capacitance extraction, each capacitance component is calculated using a single capacitance formula and multiple sensitivity formulas. This way of handling systematic variations neglects the cross-dependency impact of different variation parameters on parasitic capacitances, where the capacitance sensitivity to each variation parameter is calculated independently while keeping other parameters fixed. Moreover, the computational runtime of capacitance calculations significantly increased due to using

multiple pre-characterized formulas to calculate a single capacitance component.

This work mainly focuses on implementing a new interconnect parasitic capacitance compact models for 2D cross-section layout patterns in rule-based extraction tools. The new models use a novel input pattern representation that considers systematic process variations efficiently. The new models are compact, have high pattern coverage, mitigate pattern mismatches, and provide a faster layout parasitic capacitance extraction process. Also, the proposed compact models can replace thousands of existing capacitance and sensitivity formulas, where each model can calculate a coupling capacitance between two certain polygons using a single computation instead of multiple computations (using multiple capacitance and sensitivity formulas) in traditional rule-based methods. The contributions of this paper are:

- Machine learning compact models that predict interconnect parasitic capacitances in layouts of a certain process node. Each model predicts parasitic coupling capacitances of cross-section patterns covering a certain set of metal layers with arbitrary distributed polygons considering systematic process variations. Unlike existing models that require multiple computations to calculate a capacitance component, the compact models can calculate a certain capacitance component using a single computation. Therefore, there is no need to invoke multiple capacitance and sensitivity formulas to calculate a certain capacitance component anymore. This resulted in a higher extraction accuracy and a lower computational runtime.
- A new vertex-based representation of 2D cross-section layout patterns is proposed. The new pattern representation accounts for systematic process variations including metal thickness variations, loading effects, metal width variations, and trapezoidal variations. Also, it can handle layout patterns with arbitrary distributed polygons.
- The compact models are generated using two different machine learning methods including Neural Networks (NN) and Support Vector Regression (SVR) methods. The compact models are almost 2.5X faster than existing rule-based models.
- The proposed methodology is validated on thirteen real designs covering 28nm, 14nm, and 7nm nodes.

This work mainly focuses on two-dimensional integrated circuit (2DIC) technologies [16], where the die is mounted on a single plane inside the package. As for three-dimensional integrated circuit (3DIC) technologies, they have special parasitic modeling requirements as they contain special layer types such as through-silicon-vias (TSVs) and interposers that are used to connect multiple dies together [17], [18].

This paper is organized as follows. Section II provides a discussion on related works. Section III provides a background on rule-based parasitic capacitance extraction methods and systematic process variations. Section IV describes the proposed compact models. Section V provides the experimental results. Section VI provides the conclusion and

future works. Moreover, Table 1 shows a list of common abbreviations that are used in this work.

TABLE 1. List of abbreviations.

Abbreviation	Description
NN	Neural network
SVR	Support vector regression
RIE	Reactive ion etching
3DIC	Three-dimensional integrated circuit
TSV	Through-silicon-via
ILD	Inter layer dielectric
ADAM	Adaptive moment estimation
SGD	Stochastic gradient descent
RELU	Rectified linear activation unit
RBF	Radial Basis Function
MSE	Mean square error
STDEV	Standard deviation
RO	Ring oscillator
SRAM	Static read access memory
DRAM	Dynamic read access memory
VCO	Voltage-controlled oscillator
CM	Cache memory
DAC	Digital to analog converter
SPR	Sum of positive ranks
SNR	Sum of negative ranks

II. RELATED WORK

Many efforts were done to improve the accuracy of rule-based extraction methods [8], [9], [13], [14], [19]–[23]; however, most of them either use simplified models to improve pattern matching as in [21], [22], or tackle specific interconnect structures by using analytical models as in [8], [9], [23]. As for systematic process variations, all previous efforts of modeling the impact of systematic variations on parasitic capacitances, in rule-based methods, focused only on the accuracy of the capacitance and sensitivity formulas. They completely ignored other sources of inaccuracies such as pattern mismatches and pattern coverages. Also, they did not consider the impact on the extraction runtime after incorporating their formulas [13], [19].

In [13], a modeling methodology was developed to improve the accuracy of the 2.5D parasitic capacitance extraction method by considering reactive ion etching (RIE) variations using sensitivity formulas. This effort used traditional sensitivity methods to handle interconnect thickness variations that are caused by RIE. Such an approach has three main problems. First, it has a limited pattern coverage as it only considers basic three wires patterns. Therefore, it cannot be generalized on complicated layout patterns. Second, it adds more computational runtime on parasitic extraction tools as it introduces additional sensitivity formulas to be computed on top of existing formulas. Third, it only considers RIE effects and completely ignores the cross-dependency impact of different variation parameters on parasitic capacitances. As a result, it is not suitable for advanced process nodes.

In [19], a modeling methodology for interconnect parasitic capacitances considering lithography effects was developed.

The methodology uses a lithography simulator across many 3D layout patterns to incorporate lithography effects, such as metal width variations, into the generated 3D layout patterns. Then, it passes the modified layout patterns to a 3D field-solver to extract their parasitic capacitances. After that, the modified patterns and their parasitic capacitances are stored in a pre-characterized library to be later used by parasitic extraction tools. This effort has several problems. First, it is not applicable for 2.5D extraction methods since it only considers 3D layout patterns. Second, the lithography simulator would generate a lot of curvilinear layout shapes that add a lot of complications on layout parasitic extraction processes. The complications include more pattern mismatches, more parasitic extraction runtime due to running a lithography simulator on layouts, and a huge pre-characterization runtime due to running a 3D field-solver over many curvilinear shapes. Third, such a methodology has a very limited pattern coverage, and it cannot provide good accuracy on full chips. Moreover, the authors did not introduce any solution for the pattern coverage and pattern mismatch problems.

In [8] and [9], field-based parasitic capacitance formulas for metal wires were developed. Such formulas consider the different 3D parasitic effects of a metal wire including fringing and corner coupling capacitances; however, those formulas are only valid for isolated wires. They do not consider the impact of surrounding metals and systematic process variations. Hence, they are not efficient for full chip interconnect parasitic extraction in advanced process nodes.

In [24], a neural-network model was developed for several 3DIC interconnect structures around through-silicon-vias (TSVs). This model uses a single dielectric structure, and it is only limited to certain interconnect structures around TSVs. Hence, such a model is not efficient for multi-dielectric environment and full chip extraction.

In [22], a pattern matching classifier was developed using neural networks in order to assign each layout pattern to a corresponding capacitance model. Also, interconnect parasitic capacitance models using neural networks were developed for 2D cross-section layout patterns. Such an approach managed to reduce pattern mismatches and improve the accuracy of parasitic capacitance extraction results; however, it has three main problems. First, the proposed models use layout patterns with limited geometrical parameters. Second, the models completely ignore systematic process variations. Third, the proposed models were only verified on simple 2D cross-section patterns, and they were not verified on real layout patterns.

In [6], a preliminarily work was done to reduce pattern mismatches and improve pattern coverages. The work mainly focused on implementing neural-network models for 28nm process node. Each model represents 2D cross-section layout patterns with arbitrary distributed metal polygons. Each model handles a pre-defined set of metal layers (i.e., metal collections), e.g., metal1-metal2-metal3. Two different pattern representations were proposed to represent each set of metal layers. The proposed representations provide detailed

geometrical information for each cross-section layout pattern. The first cross-section pattern representation is called the ratio-based representation, whereas the second one is called the dimension-based representation. In the ratio-based representation, each metal layer in a pattern is represented by a vector of segments, each segment holds the ratio between the segment width and the overlapping polygon's width. On the other hand, in the dimension-based representation, each metal layer in a pattern is represented by a vector of widths and displacements. The proposed models managed to reduce pattern mismatches and improve the extraction accuracy. However, this effort has four main problems. First, the models did not consider the different systematic process variations. Second, the models did not introduce any performance improvement as compared to existing rule-based models. Third, the models were only implemented using neural networks, and they were not compared with any other machine learning method. Fourth, the models were only verified on 28nm process technology node.

Table 2 summarizes the contributions and limitations of our work and related works. Table 3 provides a comprehensive comparison among related works including our work. The comparison includes ten factors as below:

- a. The considered systematic process variations.
- b. The modeling methodology of systematic process variations, where the impact of systematic variations on parasitic capacitances can be implicitly considered while predicting parasitic capacitances by using the capacitance models (i.e., embedded inside the model), or it can be modeled using sensitivity formulas or lithography simulators.
- c. The pattern coverage, where some models may only cover a limited number of layout patterns.
- d. Type of input layout patterns (2D cross-section or 3D layout patterns).
- e. The pattern matching mechanism, which is an essential step in 2.5D extraction flows. The pattern matching is used to match layout patterns with corresponding pre-characterized capacitance formulas (or models) in order to calculate the corresponding parasitic capacitance numbers. There are two types of pattern matching that include geometry-based and layer-based. In the geometry based, the pattern matching is performed based on the geometrical structures of layout patterns, whereas in the layer-based, the pattern matching is performed based on the layer names, where each set of metal layers is handled by a specific model (regardless of its geometrical structure).
- f. The possibility of pattern mismatches that occur when parasitic capacitances of a layout pattern are calculated using inappropriate capacitance formulas.
- g. The support of multi-dielectric environment.
- h. The modeling method, which represents the method that is used to implement the models, such as analytical formulas, curve fitted formulas, lookup tables, neural networks, and support vector regressions.

TABLE 2. Contributions and limitations of this work and related works.

Research	Contribution	Limitation
Karsilayan et al. [13]	<ul style="list-style-type: none"> Sensitivity formulas were developed to model the impact of a reactive ion etching (RIE) on interconnect parasitic capacitances. This effort used traditional sensitivity methods to handle interconnect thickness variations that are caused by RIE. 	<ul style="list-style-type: none"> It has a limited pattern coverage as it only considers basic three wires patterns. It adds more computational runtime on parasitic extraction tools. It only considered reactive ion etching (RIE) effects and completely ignored the cross-dependency impact of different variation parameters on parasitic capacitances.
Tsai et al. [16]	<ul style="list-style-type: none"> A modeling methodology for interconnect parasitic capacitances considering lithography effects was developed. A lithography simulator was used to incorporate lithography effects into pre-characterized 3D layout patterns. Then, the parasitic capacitances of the modified patterns were extracted using a field-solver and stored in a pre-characterized library. In the extraction phase, a lithography simulator was used to incorporate lithography effects into a given layout. Then, a pattern matching operation was performed to match layout patterns with pre-characterized patterns. 	<ul style="list-style-type: none"> Limited pattern coverage as the method only supports 3D layout patterns. The lithography simulators would generate a lot of curvilinear layout shapes that add a lot of complications on layout parasitic extraction processes that include more pattern mismatches, more parasitic extraction runtime, and a huge pre-characterization runtime. It only considered metal width variations. It is neither suitable for advanced process nodes nor three-dimensional integrated circuits (3DIC) technologies.
Zhang et al. [8] and [9].	<ul style="list-style-type: none"> Analytical capacitance formulas were developed to calculate the different parasitic capacitance components of an isolated metal wire. Such formulas consider the different 3D parasitic effects of a metal wire such as fringing and corner coupling capacitances. 	<ul style="list-style-type: none"> The proposed formulas are only valid for isolated metal wires. They are not efficient for full chip interconnect parasitic extraction in advanced process nodes. The formulas do not consider systematic process variations. They are neither suitable for advanced process nodes nor 3DIC technologies.
Li and Shi [19].	<ul style="list-style-type: none"> A pattern matching classifier was developed using neural networks in order to assign each layout pattern to a corresponding capacitance model. Interconnect parasitic capacitance models using neural networks were developed for cross-section layout patterns. This approach managed to reduce the pattern mismatches. 	<ul style="list-style-type: none"> Limited pattern coverage as it used limited number of pre-characterized patterns. The method did not mitigate pattern mismatches. The method does not consider systematic process variations. It is not suitable for 3DIC technologies.
Abouelyazid et al. [6]	<ul style="list-style-type: none"> Neural network models were developed for 28nm process node to predict interconnect parasitic capacitances of 2D cross-section layout patterns efficiently. The proposed models use two different pattern representations that include: ratio-based and dimensions-based representations. The proposed models managed to mitigate the pattern mismatches and improve the extraction accuracy. 	<ul style="list-style-type: none"> The models only consider metal width variations, and they do not consider the different systematic process variations. The models are slower than Calibre PEX, rule-based tool, by 1.3X. The models were only verified on 28nm process technology node. The models are not suitable for 3DIC technologies.
This work	<ul style="list-style-type: none"> Machine learning compact models were developed to predict interconnect parasitic capacitances of 2D cross-section layout patterns efficiently. The models can handle layout patterns of arbitrary distributed polygons and mitigate pattern mismatches. The models use a new vertex-based layout representation to help in handling systematic process variations. The models are almost 2.5X faster than existing rule-based models. The models consider the cross-dependency impact of different variation parameters on parasitic capacitances. 	<ul style="list-style-type: none"> The generated models are only valid for 2DIC technologies, where the die is mounted on a single plane inside the package. As for 3DIC technologies, multiple dies of different (or same) process nodes would be connected together by using either TSVs or interposers [16]. This requires special modeling to predict the parasitic coupling interactions among TSVs and interconnects of different dies.

- i. The testing and validation methodologies, which indicate whether the related work was verified on real process technology nodes or not.
- j. The overhead on the actual parasitic capacitance extraction runtime, where some models require additional computations in order to predict the impact of systematic process variations on parasitic capacitances.

III. BACKGROUND

A. RULE-BASED PARASITIC CAPACITANCE EXTRACTION METHODS

Rule-based parasitic extraction methods are used in several commercial extraction tools, such as Calibre xRC [25] and

StarRC [26], because they can handle full chips efficiently. Rule-based methods employ 2.5D extraction approaches in order to extract interconnect parasitic capacitances of a given layout. In 2.5D approaches, parasitic extraction tools scan a given layout in the x and y directions to obtain all corresponding 2D cross-section layout patterns. For each cross-section pattern, plate and fringing coupling capacitances (per unit length) are calculated using pre-characterized capacitance formulas [1], [6]. The mapping between cross-section patterns and the corresponding capacitance formulas is performed using pattern matching operations. Once all capacitances are calculated, they are multiplied by the corresponding projection length to get the total capacitance values.

TABLE 3. A comparison among different parasitic capacitance extraction methods including our work.

		Karsilayan et al. [13]	Tsai et al. [16]	Zhang et al. [8] and [9].	Li and Shi [19].	Abouelyazid et al. [6]	This work
Systematic process variations	Metal width variations	No	Yes (Supported)	No	No	Yes (Supported)	Yes (Supported)
	Metal Thickness Variations	Yes (Supported)	No	No	No	No	Yes (Supported)
	Loading effects	No	No	No	No	No	Yes (Supported)
	Trapezoidal variations	No	No	No	No	No	Yes (Supported)
Modeling Method of variations	Sensitivity models	lithography simulator	Not applicable	Not applicable	Embedded inside the model	Embedded inside the model	
Pattern coverage	It has a limited pattern coverage because their models cover a limited number of layout patterns.				High pattern coverage as the proposed models can handle patterns with arbitrary distributed polygons. However, they cannot handle metal polygons that are vertically overlapped.	High pattern coverage as the proposed models can handle patterns with arbitrary distributed polygons.	
Type of supported patterns	2D cross-section patterns of three parallel wires.	3D patterns of parallel wires.	3D patterns of isolated wires.	2D cross-section patterns of parallel wires.	2D cross-section patterns of arbitrary distributed metal polygons (i.e., wires).		
Pattern matching mechanism	Geometry-based	Geometry-based	Geometry-based	Geometry-based	Layer-based	Layer-based	
Pattern mismatches	High potential pattern mismatches as the pattern matching operations are performed based on the geometrical structures.				No pattern mismatches as the pattern matching operations are performed based on the layer names. Therefore, each set of metal layers are handled using a single compact model.		
Multi-dielectric environment	Yes	Yes	No	Yes	Yes	Yes	
Modeling methods	Curve fitted formulas.	Lookup tables	Analytical formulas	Neural networks	Neural networks	Neural networks Support vector regressions	
Validation and testing	Not Tested on real designs.				Only tested on 28nm process node.	Tested on 28nm, 14nm, and 7nm process nodes.	
Overhead on the actual parasitic capacitance extraction runtime	It doubles the number of computations as sensitivity formulas are computed separately.	Running a lithography simulator adds a lot of overhead on runtimes.	It is fast as it uses analytical formulas with small number of variables.	It adds extra runtime due to running a classifier to assign each layout pattern to a capacitance model.	It is slower than existing commercial rule-based tool (i.e., Calibre PEX [25]) by 1.3X when being tested on 28nm process node.	It is 2.5X faster than existing commercial rule-based tool (i.e., Calibre PEX [25]) when being tested on 28nm, 14nm, and 7nm process nodes.	

Figure 1 provides an illustrative example of extracting a certain metal polygon using the 2.5D extraction approach. The figure shows a layout structure of three metal layers including metal1, metal2, and metal3 layers. The target metal polygon is the middle metal2 polygon. There are four cross-sections for the target metal polygon. Three cross-sections are in the z - y plane, and one cross-section is in the z - x plane. Cross-section1 (C_1) and cross-section2 (C_2) are identical, and each contains five capacitance components. Cross-section3 (C_3) contains eight capacitance components. Cross-section4 (C_4) contains

six capacitance components. The fringing and lateral capacitances are calculated for each cross-section using corresponding capacitance formulas. Then, each capacitance component is multiplied by the corresponding projection length (i.e., L_1 to L_4) [6].

As for plate capacitances, they are calculated in one cross-section, either z - x or z - y cross-sections, and multiplied by the corresponding projection length. This is done to avoid duplicate calculations of the same plate capacitance [6].

The rule-based extraction method has two main steps: 1- a pre-characterization (i.e., calibration) step, as shown in

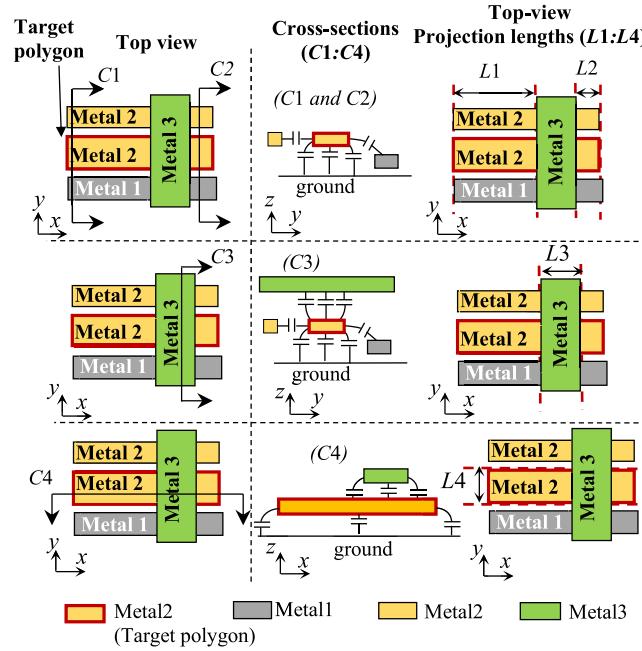


FIGURE 1. An illustrative example of extracting a metal polygon using the 2.5D parasitic capacitance extraction method [6].

Figure 2 (a), and 2- a layout parasitic capacitance extraction step, as shown in Figure 2 (b). The pre-characterization (i.e., calibration) step is responsible for generating a pre-characterized library of capacitance and sensitivity formulas, where each process technology node has a different pre-characterized library. On the other hand, the layout parasitic extraction step is responsible for analyzing layouts and calculating corresponding parasitic elements using the corresponding pre-characterized library.

1) THE PRE-CHARACTERIZATION STEP

In this step, a pre-characterized library of capacitance and sensitivity formulas are generated for a certain process technology node. The pre-characterization process starts with generating many 2D and 3D layout patterns based on the corresponding technology specifications. The structures of those patterns are pre-characterized. Then, a field-solver tool is used to extract reference parasitic capacitance values for each layout pattern. The reference capacitance numbers are either formatted in lookup tables or passed to a curve fitting tool. The curve fitting tool generates a capacitance formula for each capacitance component as below:

$$C = f(p_1, p_2, \dots), \quad (1)$$

where C represents a certain capacitance component, $f(p_1, p_2, \dots)$ represents the curve fitted capacitance formula, whereas p represents a certain geometrical parameter (e.g., width or spacing). Moreover, sensitivity formulas are generated to measure the impact of systematic process variations on each capacitance component, where each capacitance component is calculated using a single capacitance

formula and multiple sensitivity formulas as below:

$$C = f(p_1, p_2, \dots) + \sum_{i=0}^n \Delta S_i \cdot \frac{\partial C}{\partial S_i}, \quad (2)$$

where S represents a certain variation parameter (e.g., a metal thickness variation), $\partial C / \partial S$ represents a sensitivity formula that measures a capacitance sensitivity to a certain variation parameter, whereas n represents the number of systematic process variation parameters.

Eventually, the generated capacitance and sensitivity formulas are stored in a pre-characterized library in order to be later used by parasitic capacitance extraction tools [6], [27].

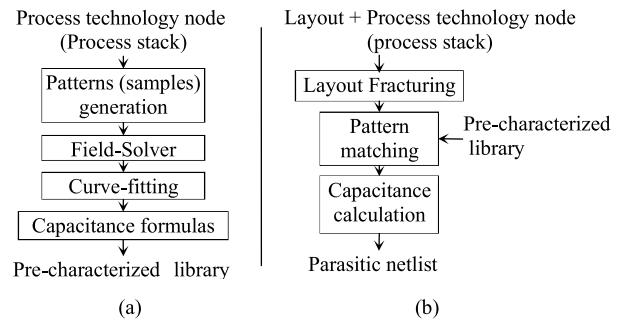


FIGURE 2. Rule-based parasitic capacitance extraction steps including (a) pre-characterization and (b) layout parasitic extraction steps [6].

2) LAYOUT PARASITIC CAPACITANCE EXTRACTION STEP

The layout parasitic capacitance extraction step is responsible for extracting parasitic capacitances of a given layout and writing the extracted parasitic elements into a parasitic netlist. The extraction flow starts with analyzing and measuring the geometries of a layout. After that, layout geometries are fractured into 2D cross-section patterns as shown in Figure 1. Then, a pattern matching operation is performed to match each 2D cross-section pattern with corresponding pre-characterized capacitance and sensitivity formulas. Eventually, the measured geometries are passed to the obtained pre-characterized formulas to calculate the corresponding capacitance values. Once all parasitic capacitances are extracted, a parasitic netlist is generated to be later used by circuit simulators to perform post-layout simulations [6], [27].

B. SYSTEMATIC PROCESS VARIATIONS

As process technology nodes scale down, the dimensions of metal wires continue to shrink, and the difficulty of controlling the variations of interconnect geometries and device parameters significantly increased [28]. There are two types of variations including random and systematic variations. Random variations represent the unpredictable and stochastic variations that cannot be associated with specific conditions or layout patterns. They might change from time to time and from location to another. The random variations are usually modeled using statistical models as in [29]–[32].

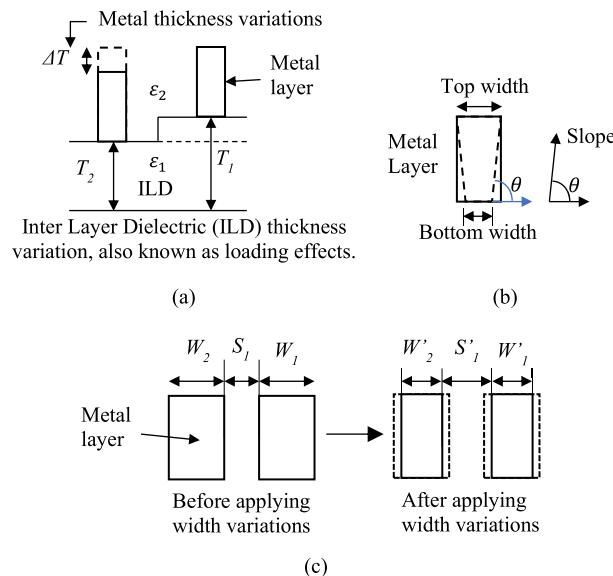


FIGURE 3. Examples of systematic process variations showing (a) metal thickness variation and loading effects, (b) trapezoidal variations, and (c) metal width variations.

On the other hand, systematic variations represent the predictable and deterministic variations that are associated with specific process conditions (e.g., chemical mechanical polishing) and layout patterns. In advanced process technology nodes, the impact of systematic variations on parasitic capacitances increases because systematic variations represent higher percentages of interconnect and device dimensions [12], [13], [33]. The main systematic process variations include metal thickness variations, inter layer dielectric thickness variations (i.e., loading effects), metal width variations (e.g., etching), and trapezoidal variations of metal layers as shown in Figure 3. Figure 3 (a) shows examples of metal thickness variations and inter layer dielectric thickness variations (i.e., loading effects). The loading effects mainly impact the thickness of inter layer dielectrics and the elevation of the corresponding upper metal layers, whereas the metal thickness variations mainly impact the top thickness of the corresponding metal layer. Figure 3 (b) shows an example of trapezoidal variations in metal layers, where the sidewall slope of a certain metal layer changes. Figure 3 (c) shows an example of a metal width variation that impacts the width of metals and the separation between them.

Since systematic variations are pattern dependent, parasitic capacitance extraction tools usually model their effects using sensitivity formulas as in [13]–[15]. Such a modeling approach has three main problems that impact the extraction accuracy: 1- it neglects the cross-dependency impact among different variation parameters on parasitic capacitances, 2- it uses a limited number of patterns and parameters to model the impact of systematic variations on parasitic capacitances, and 3- it has a high potential of pattern mismatches similar to the case of capacitance calculations (i.e., formulas). Moreover, the current handling of systematic variations introduces extra computational runtime, where each capacitance component

is calculated using a single capacitance formula and multiple sensitivity formulas as shown in (2).

IV. COMPACT MODELS FOR PARASITIC CAPACITANCES CONSIDERING SYSTEMATIC VARIATIONS

A novel modeling methodology for interconnect parasitic capacitance extraction is developed for rule-based extraction methods. The proposed methodology uses machine learning methods to create compact models that predict parasitic coupling capacitances between metal polygons in 2D cross-section layout patterns. Unlike existing models, the compact models handle patterns with arbitrary distributed polygons, consider connected polygons (i.e., polygons that hold the same potential), reduce pattern mismatches, increase pattern coverage, and consider systematic process variations. The compact models are technology-dependent, where each process technology node has a pre-characterized set of compact models. The proposed compact models enabled the extraction of more complicated and multi-dimensional layout patterns. Moreover, each compact model can replace hundreds to thousands of existing capacitance and sensitivity formulas. Therefore, the compact models managed to provide a lower computational runtime, significant reduction in pattern mismatches, and significant accuracy improvements.

The implementation process of the parasitic capacitance compact models consists of five main steps as follows: 1- identify the main characteristics of input patterns, 2- obtain training patterns, 3- generate reference parasitic capacitance numbers of training patterns, 4- extract features of cross-section patterns, and 5- train machine learning models. Figure 4 shows the implementation process of interconnect parasitic capacitance compact models.

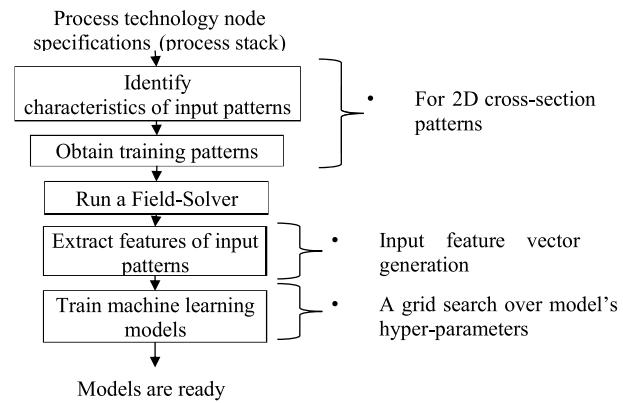


FIGURE 4. The process of implementing 2D cross-section machine learning compact models for rule-based extraction methods.

A. IDENTIFY INPUT PATTERNS CHARACTERISTICS

To create a compact model, we need to study several factors that identify the main characteristics of input patterns. The factors include: the surrounding multi-dielectrics, the window size of a cross-section pattern, the number of metal layers in a pattern window, the number of metal polygons in each layer, and systematic process variations.

1) SURROUNDING MULTI-DIELECTRICS

Each process technology node (i.e., process stack) consists of multiple metal layers that are placed vertically and surrounded by dielectrics. Each metal layer has its own geometrical specifications such as minimum width, minimum spacing, thickness, elevation, and corresponding systematic process variation parameters. The metal layers are separated by dielectric structures. The dielectrics can be planar or conformal. Each dielectric has certain specifications such as a dielectric constant and thickness. Figure 5 shows an example of a typical process technology node stack (i.e., process stack) with multi-dielectric environment. The surrounding dielectrics have a direct impact on coupling capacitances between metal layers. So, they must be considered during parasitic capacitance extraction processes.

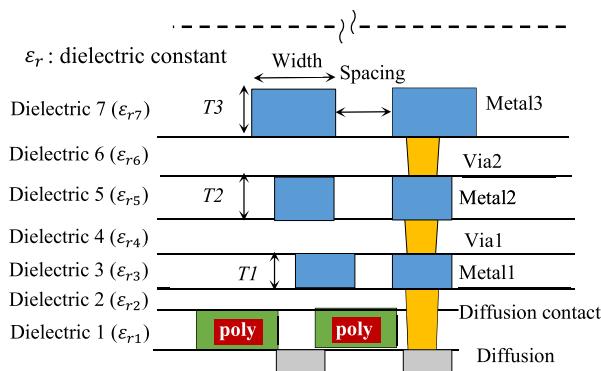


FIGURE 5. An example of a process technology node (i.e., process stack) with multi-dielectric environment [6].

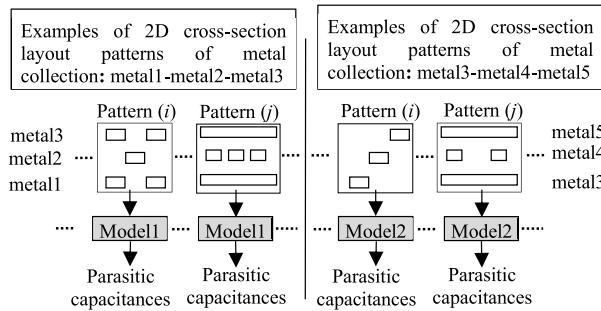


FIGURE 6. An example of machine learning compact models for cross-section patterns of two different metal collections.

However, including the surrounding dielectrics into the input parameters to our parasitic models would complicate the models, require more training patterns, increase pattern mismatches, and add more overhead on training and prediction runtimes. Therefore, to avoid such complications and generate effective parasitic models, each process technology node (i.e., process stack) must have its own set of parasitic capacitance compact models. Also, each pre-defined set of metal layers (i.e., metal collection), in a certain process technology node, must have a certain parasitic capacitance compact model as shown in Figure 6, for example,

metal1-metal2-metal3 collection has a compact model, whereas metal3-metal4-metal5 collection has another compact model. In other words, each process technology node would have a separate pre-characterized library of machine learning compact models.

2) WINDOW SIZE OF CROSS-SECTION PATTERNS

The window size of a 2D cross-section pattern represents the width of the pattern in the horizontal direction as shown in Figure 7. When the size of a pattern window increases, the number of polygons that overlap with the window increases. Hence, more coupling capacitance components are extracted. However, this would trigger the extraction of minor capacitance components that do not have any observable impact on the extraction accuracy. Moreover, extracting such minor capacitance components would significantly increase the extraction runtime without any considerable gain. As a result, the pattern window should only consider the coupling capacitances that impact the extraction accuracy.

As the separation between any two metal polygons increases, the coupling capacitance between them decreases as shown in Figure 8. Hence, any metal polygon would have an effective coupling distance (i.e., range), where any coupling capacitance to a polygon that is outside of this range is negligible.

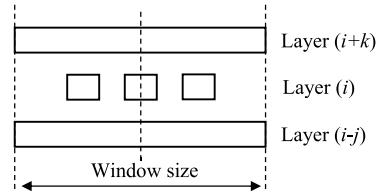


FIGURE 7. An example of a 2D cross-section pattern of a certain metal collection showing the corresponding window size.

A pattern window size is identified by using the maximum coupling range of a target metal layer. The maximum coupling range is the maximum distance where the lateral coupling capacitance between two polygons, which belong to the same target metal layer, represents 1% of their total capacitances. Therefore, all coupling capacitances to polygons that are outside of this range are ignored. For each metal layer, the maximum coupling range is calculated by constructing a 2D cross-section pattern of two adjacent polygons using minimum dimensions. The total and lateral coupling capacitances are calculated by a 2D field-solver. The separation (i.e., spacing) between the two polygons is increased until the lateral coupling capacitance between the two polygons is less than or equal to 1% of the total capacitance on one polygon. Figure 8 shows an example of calculating the maximum interaction range for metal3 layer in 28nm process node. The capacitance unit is in femtofarad (fF), whereas the separation unit is in micrometer (μm).

3) THE NUMBER OF METAL LAYERS IN A PATTERN

Each cross-section layout pattern consists of arbitrary distributed metal polygons that belong to the same or

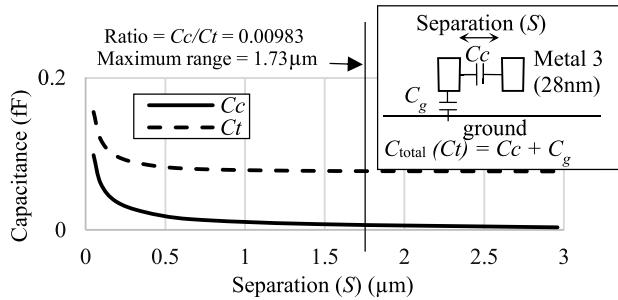


FIGURE 8. An example of calculating the maximum coupling range using metal3 layer with minimum dimensions in 28nm process technology node. The capacitance unit is in femofarad (fF), whereas the separation unit is in micrometer (μm).

different metal layers. Most of existing rule-based models handle cross-section layout patterns with one, two, and three metal layers [34]–[37]. This might be enough for high density layout designs; however, for low density designs, the capacitance models should consider more than three metal layers to provide a higher extraction accuracy.

The maximum number of layers in a pattern is identified by measuring the impact of adding multiple upper and lower metal layers on total and lateral capacitance of a target metal layer. The maximum number of upper metal layers (or lower layers) is identified by constructing multiple 2D cross-section patterns of two adjacent metal polygons. Each 2D cross-section pattern has a different numbers of upper metal layers (or lower) as shown in Figure 9 (a). The lateral capacitance, of a target metal polygon, is measured using a 2D field-solver, while adding more upper metal layers, until the impact of adding more upper metal layers on the lateral capacitance is negligible (< 1% difference in the lateral capacitance). It is worth mentioning that the patterns are constructed on a way that minimize the impact of intermediate upper metal layers and maximize the impact of the most upper metal layer on the lateral capacitance, where all intermediate upper metal layers are represented by a single polygon with minimum dimensions, whereas the most upper metal layer is represented by a plane. This process is applied on all metal layers on a process stack. Also, the same process is applied to the maximum number of lower metal layers.

Figure 9 shows an example of identifying the maximum number of upper metal layers using metal1 as a target layer in 28nm process node. Figure 9 (a) shows the constructed patterns, whereas Figure 9 (b) shows the lateral coupling capacitance values with increasing the number of upper metal layers. The results show that adding more than two upper layers has a minor impact (< 1% difference in the lateral capacitance) on the lateral capacitances. This process is tested on different process nodes including 28nm, 14nm, and 7nm nodes to identify the maximum number of upper and lower metal layers. The experiments show that adding more than two upper or lower metal layers has a minor impact on the lateral capacitance of a target layer. As a result, the maximum number of metal layers in a pattern is five, i.e., two upper layers, two lower layers, and one target layer.

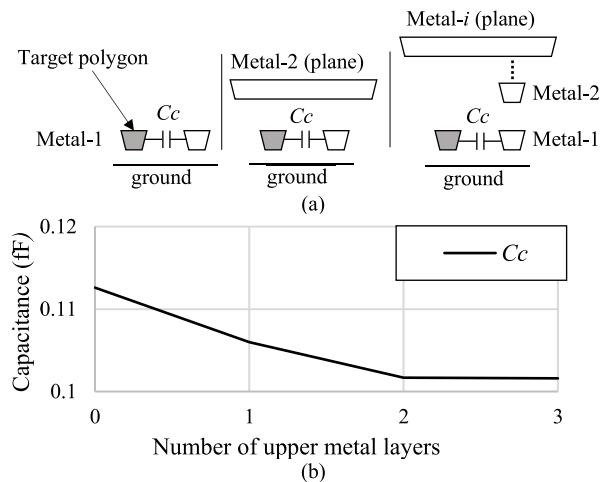


FIGURE 9. An example showing (a) 2D cross-section patterns that are created to identify the maximum number of upper metal layers for a target metal layer, and (b) the impact of adding upper metal layers on the lateral capacitance of a target metal layer. The results are generated using metal1 as a target layer in 28nm process node.

4) MAXIMUM NUMBER OF POLYGONS IN A PATTERN

Each pattern may contain multiple polygons across different metal layers. It is not necessarily for all polygons to have considerable coupling capacitances to target polygons, where some of the capacitances are considerable and impact the extraction accuracy, whereas other capacitances may be minor and do not impact the extraction accuracy. As a result, surrounding polygons that only impact the parasitic extraction accuracy, of target metal polygons, should be considered by the corresponding model.

The maximum possible number of polygons in a pattern is identified for each metal layer separately, where each metal layer in a pattern may have a different maximum number of polygons. In other words, the maximum number of polygons is identified for a target metal layer and surrounding (i.e., secondary) metal layers in a cross-section pattern. As for a target metal layer, the maximum number of polygons is identified by constructing 2D cross-section patterns of 3, 5, and 7 adjacent polygons as shown in Figure 10 (a). The lateral capacitance between the middle and right polygons is measured in each case, by using a 2D field-solver, until the impact of adding more adjacent polygons on the lateral capacitance is negligible (< 1% difference in the lateral capacitance). This process is applied on all metal layers in a process stack.

Figure 10 shows an example of identifying the maximum number of target metal polygons using metal1 as a target layer in 28nm process node. Figure 10 (a) shows the constructed patterns, whereas Figure 10 (b) shows the lateral coupling capacitance values with increasing the number of adjacent polygons. This process is tested on different process nodes including 28nm, 14nm, and 7nm nodes. The experiments show that the appropriate maximum number of polygons for a target metal layer is 5.

As for upper and lower (i.e., secondary) metal layers, the maximum numbers of polygons are calculated by

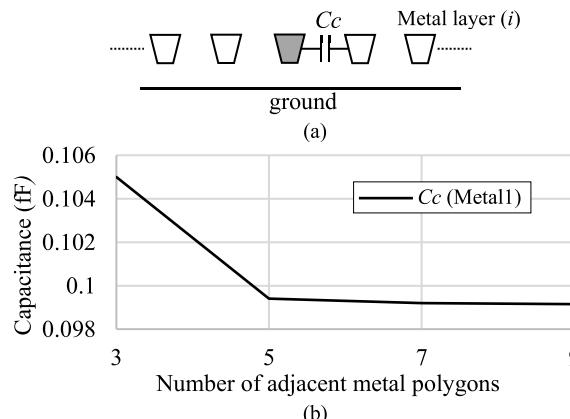


FIGURE 10. An example of (a) 2D cross-section patterns that are used to identify the maximum number of target metal polygons in an input pattern, and (b) the impact of adding more adjacent polygons on a same layer lateral capacitance. The results are generated using metal1 as a target layer in 28nm process node.

constructing 2D cross-section patterns of two metal layers (i.e., the target and secondary layers) as shown in Figure 11 (a). The target metal layer contains one polygon at the middle, whereas the secondary metal layer has a varying number of polygons (from 2 to 7). All polygons are constructed using the corresponding minimum dimensions. The total capacitance on the middle target polygon is measured, using a 2D field-solver, in each case until the impact of adding more secondary layer polygons on the total capacitance is negligible (< 1% difference in the total capacitance). This process is applied on all metal layers in a process stack.

Figure 11 shows an example of identifying the maximum number of secondary metal layer polygons using metal1 as a target layer and metal2 as a secondary layer in 28nm process node. Figure 11 (a) shows the constructed pattern, whereas Figure 11 (b) shows the total capacitance values with increasing the number of secondary metal layer polygons. This process is tested on different process nodes including 28nm, 14nm, and 7nm nodes. The experiments show that the appropriate maximum number of polygons for a secondary metal layer is 4.

Eventually, the maximum number of polygons in a target metal layer is 5, whereas the maximum number of polygons in each secondary metal layer is 4. For example, the maximum number of polygons in metal1-metal2-metal3 cross-section pattern is 13, where metal1 may contain up to 4 polygons, metal2 may contain up to 5 polygons, and metal3 may contain up to 4 polygons.

5) SYSTEMATIC PROCESS VARIATIONS

Systematic process variations may have a major impact on parasitic capacitances in advanced process technology nodes. They do not only impact parasitic capacitances of associated polygons, but they also may impact parasitic capacitances of surrounding polygons [5], [12], [38]. Therefore, parasitic models must consider systematic process variations along with input patterns in order to improve the accuracy of

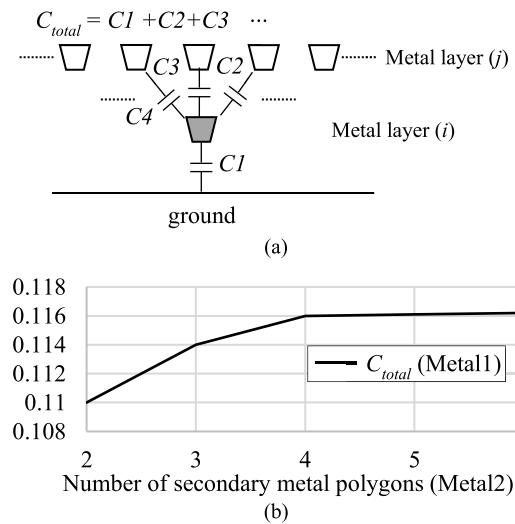


FIGURE 11. An example of (a) 2D cross-section patterns that are used to identify the maximum number of secondary metal polygons in an input pattern, and (b) the impact of adding more secondary metal polygons on a target layer total capacitance. The results are generated using metal1 as a target layer and metal2 as a secondary layer in 28nm process node.

parasitic capacitance extraction processes. In other words, the inputs to a parasitic model should be a 2D cross-section layout pattern along with the corresponding systematic process variations.

Systematic process variations are pattern dependent. They are provided by foundries in the form of lookup tables through a technology specifications file such as interconnect technology file (ITF) [12]. Therefore, systematic variations can be processed by parasitic extraction tools.

Figure 12 shows an example of metal width variations using metal1 layer with minimum dimensions in 28nm process node. Figure 12 (a) shows the impact of metal width variations on metal dimensions. The width variations impact both the width of metal wires and the separation between them, where increasing the width of metal wires decreases the separation between them. Figure 12 (b) shows the impact of width variations on lateral and total capacitances using metal1 layer with minimum dimensions in 28nm process node. The width variations may cause the lateral and total capacitances to change by more than 50%.

Figure 13 (a) shows an example of metal thickness variations using metal1 layer with minimum dimensions in 28nm process node. Figure 13 (b) shows the impact of metal thickness variations on lateral and total capacitances. The results show that the metal thickness variations may cause the lateral and total capacitances to change by more than 20%.

Figure 14 (a) shows an example of inter layer dielectric (ILD) thickness variations below metal1 layer with minimum dimensions in 28nm process node. Figure 14 (b) shows the impact of ILD thickness variations on the total capacitance. The results show that the ILD thickness variations may cause the total capacitances to change by more than 10%.

Figure 15 (a) shows an example of trapezoidal variations using metal1 layer with minimum dimensions in

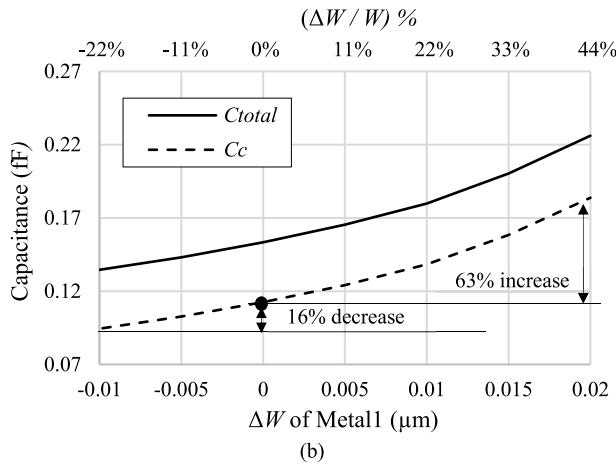
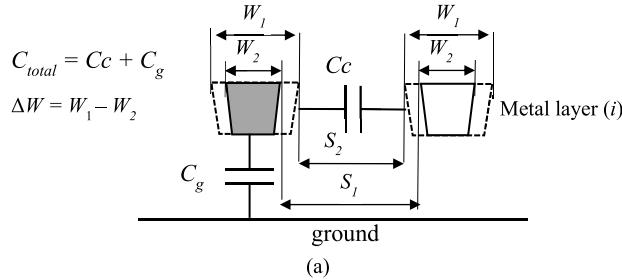


FIGURE 12. An example of (a) width variations in cross-section interconnect patterns, and (b) the impact of metal width variations on same layer lateral and total capacitances. The results are generated using metal1 in 28nm process node.

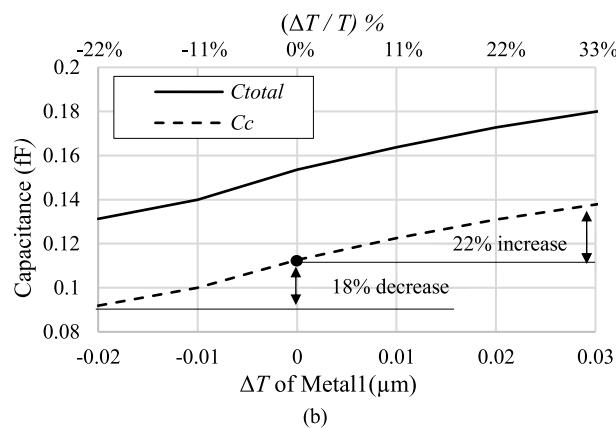
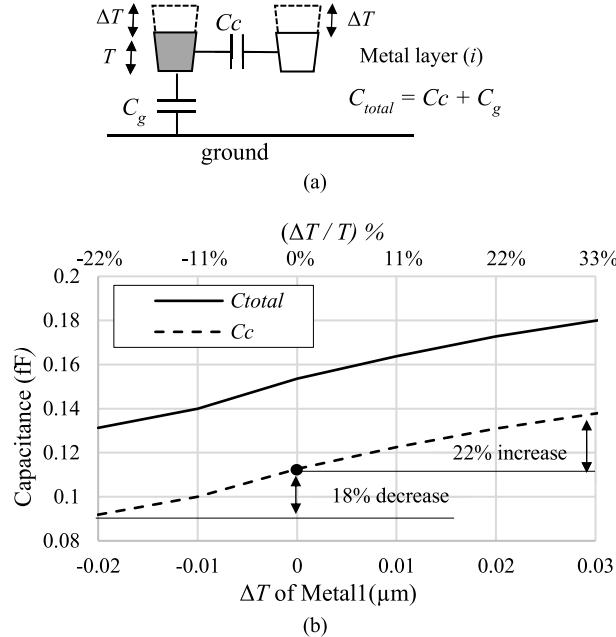


FIGURE 13. An example of (a) metal thickness variations in cross-section interconnect patterns, and (b) the impact of metal thickness variations on same layer lateral and total capacitances using metal1 with minimum dimensions in 28nm process technology node.

28nm process node. Figure 15 (b) shows the impact of trapezoidal variations (i.e., sidewall slope) on the lateral and total capacitances. The results show that the trapezoidal variations

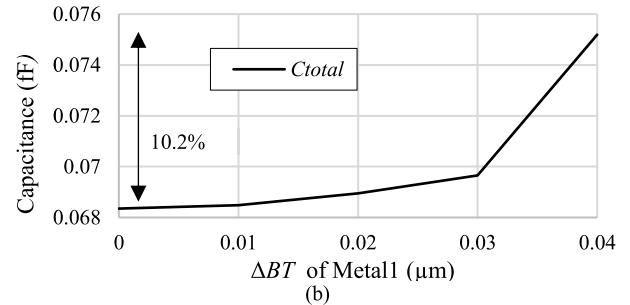
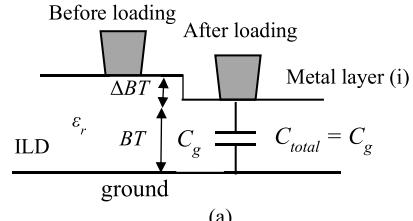


FIGURE 14. An example of (a) ILD thickness variations in cross-section interconnect patterns, and (b) the impact of ILD thickness variations on total capacitances using metal1 with minimum dimensions in 28nm process technology node.

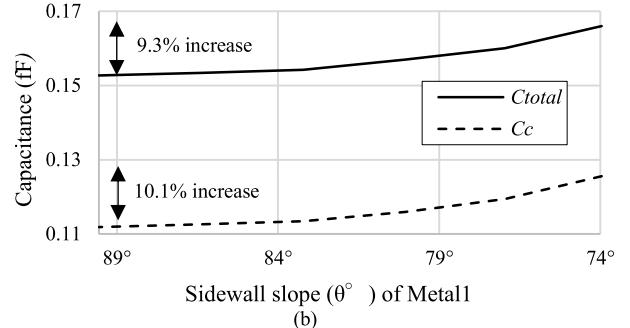
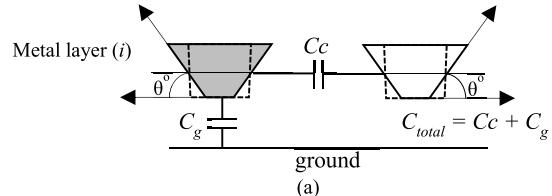


FIGURE 15. An example of (a) trapezoidal variations in cross-section interconnect patterns, and (b) the impact of trapezoidal variations on total capacitances. The results are generated using metal1 with minimum dimensions in 28nm process technology node.

may cause the total and lateral capacitances to change by more than 9%. Table 4 summarizes all required characteristics of input patterns.

Eventually, the maximum number of models for a process stack with N metal layers is given by:

$$\text{Number of models} = (^N C_k + ^N C_{k-1} \cdots + ^N C_1), \quad (3)$$

where C is the combination function, k is the maximum number of layers in a pattern of a certain layer collection. Usually, the number of models in a process stack ranges from tens to few hundreds, whereas the corresponding number

TABLE 4. A summary of input pattern characteristics.

Characteristic	The way of handling a certain characteristic
Multi-dielectric stacks	Multi-dielectric environments are handled by using a different pre-characterized library for each process stack.
The size of cross-section pattern window (in the horizontal x-direction)	The pattern size, in x-direction, equals to the maximum coupling range of the corresponding target metal layer.
The maximum number of metal layers in a pattern	5 metal layers
The maximum number of metal polygons	For a target metal layer: 5 polygons For a secondary metal layer: 4 polygons
Systematic process variations	Inputs patterns of parasitic models should include systematic variations.

of traditional rule-based formulas is in the range of many thousands.

B. GENERATE 2D CROSS-SECTION PATTERNS

Once all input pattern characteristics are identified, they are used to generate input and training patterns for parasitic models. The training patterns are obtained from several real designs in order to increase the pattern coverage and make sure that training patterns reflect real design topologies. The generation process of training patterns starts with selecting several real designs, for example, ring oscillator (RO), static read access memory (SRAM), and digital to analog converter (DAC) layout designs. Then, the geometries and dimensions of all selected designs are modified by applying the corresponding systematic process variations. After that, the modified layouts are fractured into 2D cross-section patterns taking into considerations the corresponding characteristics of input patterns. In addition, more patterns are generated randomly for each metal collections covering the ranges from 1X to 10X of minimum dimensions. Eventually, the obtained 2D cross-section patterns are used as training patterns to machine learning models. The total number of obtained cross-section patterns for each model is 30K patterns, where each model handles patterns of a certain metal layer collection (e.g., metal1-metal2-metal3).

C. FIELD-SOLVER EXECUTION

Once all training patterns are obtained, their parasitic capacitances are extracted using Raphael2D, a 2D field-solver tool [39]. The extracted parasitic capacitances are used as reference numbers to train our machine learning models.

D. VERTEX-BASED FEATURE REPRESENTATION

Parasitic capacitance models require three main inputs to predict parasitic capacitances efficiently. The inputs are: 1- pattern's geometries, 2- corresponding systematic process variations, and 3- required capacitance components. The three inputs are represented by a single input feature vector that is passed to the corresponding machine learning model.

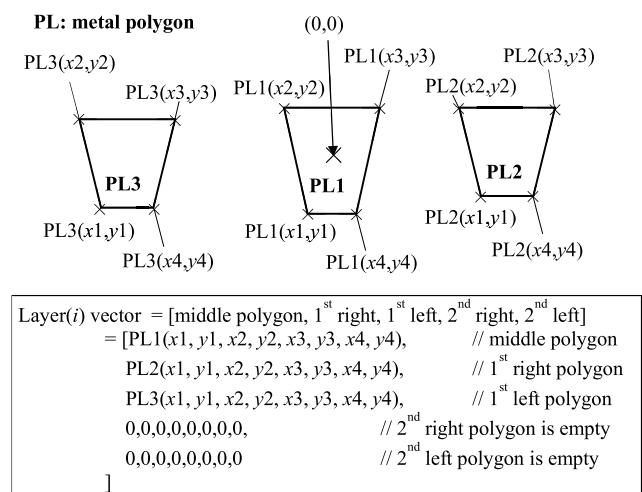


FIGURE 16. An example showing the novel vertex-based pattern representation using three polygons of the same metal layer in a cross-section pattern.

The pattern's geometries and systematic process variations are represented together by using a novel vertex-based feature representation. In vertex-based representation, each metal layer in a pattern is represented by a vector of polygons. The number of polygons of each metal layer in a pattern is shown in Table 4, where the maximum number of polygons of a target metal layer is 5, whereas the maximum number of polygons of a secondary metal layer is 4. Each metal polygon in a vector is represented by the polygon's vertices, where each vertex is measured from the center of the corresponding pattern. In other words, each polygon is represented by 8 displacement parameters including (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , and (x_4, y_4) as shown in Figure 16. As a result, each polygon is represented by 8 values (vertices), and the vector size of each layer is estimated by $(8 \times \text{maximum number of polygons in a metal layer})$. It is worth mentioning that the vertices of empty polygons are represented by zeros as shown in Figure 16.

Such a vertex-based representation considers metal thickness variations, loading effects, wire width variations, and trapezoidal variations of all polygons in a pattern simultaneously. In other words, it includes systematic process variations during capacitance calculations. Therefore, there is no need to invoke traditional sensitivity formulas or any special modeling to handle systematic process variations. Also, such a representation considers the cross-dependency impact of different variation parameters on parasitic capacitances. This resulted in fewer computations, better performance, and more accurate parasitic extraction results.

The next required input parameter by parasitic models is the required capacitance component, which informs the model about the capacitance components to be extracted. The required capacitance components are identified by including the geometries of aggressor and victim polygons to the input vector of parasitic models. Therefore, the input feature vector is represented by three internal vectors. The first vector contains geometries of all polygons, the second vector

contains geometries of aggressor polygons, whereas the third vector contains geometries of victim polygons as shown in Figure 17. The three vectors have the same size. The novel vertex-based pattern representation is used to represent the polygons in the three vectors. The size of an internal vector is estimated by:

$$\begin{aligned} \text{internal vector size} = & 8 \times (4 \times \text{number of secondary layers} \\ & + 5 \times \text{number of target layers}), \end{aligned} \quad (4)$$

whereas the input feature vector size of is estimated by:

$$\text{input feature vector size} = 3 \times \text{internal vector size}, \quad (5)$$

for example, the input vector size of a pattern with one target metal layer is 120, where the maximum number of polygons of a target metal layer is 5, each polygon is represented by 8 parameters (i.e., vertices), and there are three internal vectors with the same size (i.e., all polygons, aggressor polygons, and victim polygons). Table 5 shows the input vector sizes of different metal collections (i.e., models).

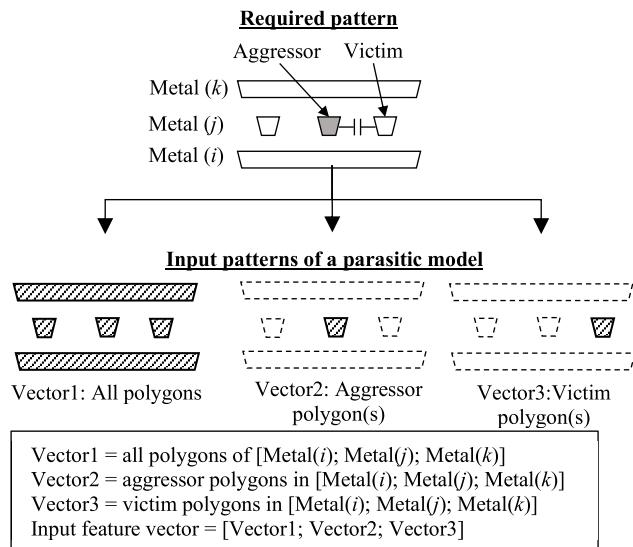


FIGURE 17. An example showing the input vector of a parasitic model.

E. TRAINING PARASITIC MODELS

Two different machine learning methods are used to create parasitic capacitance models including Neural Networks (NN) and Support Vector Regressions (SVR). The models are used to predict parasitic coupling capacitances between metal polygons in 2D cross-section patterns. For a certain process technology node, there is a model for each metal collection, where metal1-metal2-metal3 has a model, whereas metal2-metal4-metal5 has another model. The inputs of the models are the vertex-based representation of all polygons followed by aggressor and victim polygons as shown in Figure 18.

TABLE 5. Input vector sizes of several metal collections and models.

Collection	Input vector size
Metal1-Metal2-Metal3	Target Metal: Metal2 Secondary Metals: Metal1, Metal3 Target Metal polygons: 5 Secondary Metal polygons: 4 + 4 = 8 Number of Polygons: 8 + 5 = 13 Vertex-based representation: 13 × 8 = 104 Input vector size: 104 × 3 = 312
Metal2-Metal3-Metal5-Metal6	Target Metal: Metal3 Secondary Metals: Metal2, Metal5, Metal6 Target Metal polygons: 5 Secondary Metal polygons: 4 + 4 + 4 = 12 Number of Polygons: 12 + 5 = 17 Vertex-based representation: 17 × 8 = 136 Input vector size: 136 × 3 = 408
Metal3	Target Metal: Metal3 Target Metal polygons: 5 Number of Polygons: 5 Vertex-based representation: 5 × 8 = 40 Input vector size: 40 × 3 = 120
Metal1-Metal2	Target Metal: Metal1 Secondary Metals: Metal2 Target Metal polygons: 5 Secondary Metal polygons: 4 Number of Polygons: 4 + 5 = 9 Vertex-based representation: 9 × 8 = 72 Input vector size: 72 × 3 = 216

1) NEURAL-NETWORKS MODELS

A Neural Network (NN) model is implemented to predict parasitic capacitances in 2D cross-section patterns. There is a NN model for each metal collection in a certain process technology node. The architecture and hyper-parameters of NN models are obtained using a grid search algorithm. The purpose of applying a grid search algorithm is to obtain appropriate NN architectures. The NN architectures are obtained based on the number of metal layers in the corresponding metal collection. For example, a metal collection with five metal layers has a NN architecture, whereas a metal collection with four metal layers has another NN architecture.

The grid search algorithm is applied on fully connected neural networks. The search range of the grid search covers several parameters including the number of layers, number of neurons in each layer, activation functions, optimizer, batch size, learning rate, and initializations. Table 6 summarizes the search ranges of each parameter. The evaluation criteria of selecting a NN architecture are set based on the test set accuracy, where the grid search observes the accuracy of test sets across all architectures until a mean square error of 0.01% is achieved. Such a process is applied on 28nm, 16nm, and 7nm process nodes in order to obtain unified NN architectures for each metal collection model. Table 7 shows the obtained NN architectures for each input vector size.

As for hyper-parameters, the dataset is divided into 70% training data and 30% test data, validation set is 10%, the number of epochs is 1K, adaptive moment estimation (ADAM) optimizer is used, the learning rate is set to 1e-3,

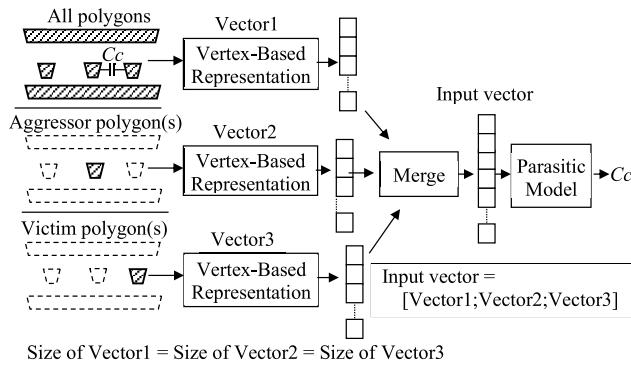


FIGURE 18. An example showing the flow of generating an input feature vector of a parasitic capacitance model.

TABLE 6. Search ranges of neural network architectures.

Parameter	Search range
Number of layers	From 1 to 7 with a step size of 1
Number of neurons in each layer	$n/7, n/6, n/5, n/4, n/3, n/2$, and n , where n is the input vector size.
Activation function	The rectified linear activation unit (RELU) and hyperbolic tangent function (tanh)
Optimizer	The adaptive moment estimation (ADAM) and stochastic gradient descent (SGD)
Batch size	500, 1000, and 1500
Learning rate	1e-2, 1e-3, and 1e-4

the batch size is set to 500, the cost function is set to a mean square error, and the batch normalization is applied. These parameters are obtained using a grid search.

2) SUPPORT VECTOR REGRESSIONS

Support vector regression (SVR) models are implemented to predict parasitic coupling capacitances of 2D cross-section patterns. There is a model for each metal collection in a certain process technology node. In order to obtain unified hyper parameters for all models, a grid search algorithm is applied across 28nm, 14nm, and 7nm process nodes. The search range of SVR models includes kernel, regularization parameter (C), gamma, and epsilon parameters. The search ranges of these parameters are listed in Table 8. The cost function is set to a mean square error. The evaluation criteria are set based on the test set accuracy, where the grid search observes the accuracy of test sets across different combinations of hyper-parameters until a mean square error of 0.01% is achieved. Table 9 shows the obtained SVR hyper-parameters for each input vector size.

V. EXPERIMENTAL RESULTS

The proposed modeling methodology was tested across three different process technology nodes including 28nm, 14nm, and 7nm process nodes. The testing covered several real designs for each node. The accuracy of the generated compact

TABLE 7. Neural network architectures of parasitic capacitance models.

Input vector	NN architecture
One metal layer Input vector size: 120	Number of hidden layers: 2. Number of neurons in hidden layers: 40 and 60, respectively. Activation functions: RELU and tanh, respectively. Cost function: mean square error. Initializations: He's normal [40] and Glorot's normal [41] initializations, respectively.
Two metal layers Input vector size: 216	Number of hidden layers: 2. Number of neurons in hidden layers: 72 and 108, respectively. Activation functions: RELU and tanh. Cost function: mean square error. Initializations: He's normal and Glorot's normal initializations, respectively.
Three metal layers Input vector size: 312	Number of hidden layers: 2. Number of neurons in hidden layers: 78 and 156, respectively. Activation functions: RELU and tanh, respectively. Cost function: mean square error. Initializations: He's normal and Glorot's normal initializations, respectively.
Four metal layers Input vector size: 408	Number of hidden layers: 3. Number of neurons in hidden layers: 68, 102, and 204, respectively. Activation functions: RELU, tanh, and tanh, respectively. Cost function: mean square error. Initializations: He's normal, Glorot's normal, and Glorot's normal initializations, respectively.
Five metal layers Input vector size: 504	Number of hidden layers: 3. Number of neurons in hidden layers: 84, 126, and 252, respectively. Activation functions: RELU, tanh, and tanh, respectively. Cost function: mean square error. Initializations: He's normal, Glorot's normal, and Glorot's normal initializations, respectively.

TABLE 8. Search ranges of support vector regression hyper-parameters.

Parameter	The search range
Kernel	Radial Basis Function (RBF) and polynomial.
Regularization parameter (C)	From 1 to 20 with a step size of 1
Epsilon	From 0.05 to 0.5 with a step size of 0.05
Gamma	From 0.1 to 1 with a step size of 0.1

models was measured relative to Raphael, 2D field-solver. Also, the accuracy and runtime of the generated NN and SVR compact models were compared against Calibre PEX cross-section models [25] and ratio-based cross-section models in [6] using sensitivity formulas of Calibre PEX to handle systematic process variations [25]. The relative error was measured for each capacitance component in a layout pattern

TABLE 9. SVR hyper-parameters of parasitic capacitance models.

Input vector of a model	SVR Hyper-parameters
One metal layer Input vector size: 120	Kernel: Radial Basis Function (RBF) Regularization parameter (C): 8 Epsilon: 0.1 Gamma: 0.3
Two metal layers Input vector size: 216	Kernel: Radial Basis Function (RBF) Regularization parameter (C): 9 Epsilon: 0.1 Gamma: 0.4
Three metal layers Input vector size: 312	Kernel: Radial Basis Function (RBF) Regularization parameter (C): 9 Epsilon: 0.1 Gamma: 0.4
Four metal layers Input vector size: 408	Kernel: Radial Basis Function (RBF) Regularization parameter (C): 10 Epsilon: 0.1 Gamma: 0.3
Five metal layers Input vector size: 504	Kernel: Radial Basis Function (RBF) Regularization parameter (C): 10 Epsilon: 0.1 Gamma: 0.3

using the below formula:

$$\text{Relative error} = (\text{predicted} - \text{reference})/\text{predicted}, \quad (6)$$

where the predicted value represents the capacitance value that is obtained from the model, whereas the reference value represents the corresponding capacitance value that was obtained from Raphael, 2D field-solver. Moreover, nonparametric statistical tests were performed to test the significant difference in performance (i.e., accuracy) between each two models.

For each process technology node, the proposed modeling methodology was used to generate NN and SVR models. The training data were obtained from real layouts including static read access memory (SRAM), digital to analog converter (DAC), and ring oscillator (RO) designs. Also, more training patterns were randomly generated covering the ranges from 1X to 10X of the minimum dimensions. The training and model's generation used Tensor flow libraries [42]. The training used Intel Xeon(R) E5-2680, 4CPU, 2.50GHz, and 16G of RAM. The errors of extracted capacitances were measured relative to Raphael, 2D field-solver.

A. TESTING REAL DESIGNS OF 28nm PROCESS NODE

The total number of generated models (either NN or SVR) is 130. The generated models cover 130 different metal collections each includes 1 to 5 different metal layers. Each model (i.e., NN or SVR model) was trained over 30K cross-section patterns, where 21K patterns (70%) were used for the training set, and 9K patterns (30%) were used for the test set. As for NN models, the total training runtime of all models is 19.3 hours. As for SVR models, the total training runtime of all models is 12.7 hours. The training (i.e., models generation) runtimes can significantly improve by multi-processing. It is worth mentioning that the models were generated only once for each process node. After that, the generated models are used numerous times by parasitic extraction tools.

TABLE 10. The accuracy and relative errors of test sets for NN and SVR models of 28nm process node.

	NN models	SVR models
Mean of relative errors	0.031%	0.052%
Standard deviation of relative errors	2.13%	2.35%
Outliers with relative error > 5%	1.45%	1.67%
Mean square error	0.00173	0.00216

Table 10 shows the test sets accuracy of NN and SVR models. The training and test sets accuracy comparison used four main criteria including 1) the mean of all relative errors, 2) the standard deviation of all relative errors, 3) the percentage of outliers that exceeds 5% relative error (i.e., the number of outliers to the total number of extracted capacitance components), and 4) the mean square error across all models. The accuracy results of test sets show that the NN and SVR models provide a high accuracy, where almost 98% of the extracted capacitances have relative errors below 5%.

As for testing the generated models on real design patterns of 28nm process node, the generated models were tested over cross-section patterns of three different test chips including dynamic read access memory (DRAM), static read access memory (SRAM), and voltage-controlled oscillator (VCO) designs that were not included during the training processes. The total numbers of extracted cross-section patterns in DRAM, SRAM, and VCO designs are 790K, 327K, and 953K patterns, respectively. The corresponding total number of capacitance components are 2.76M, 1.3M, and 4.2M capacitances, respectively. Therefore, the total number of extracted cross-section patterns across all designs is 2.07M patterns, and the total number of extracted capacitances across all designs is 8.26M.

Figure 19 shows histograms of relative errors covering all extracted capacitances across all designs using the rule-based extraction, ratio-based, proposed NN, and proposed SVR cross-section models. The accuracy comparisons show that the proposed NN and SVR models provide high accuracy results as compared to existing rule-based cross-section models and ratio-based models. The percentages of extracted capacitance components with relative errors below 5% using the rule-based, ratio-based, NN, SVR cross-section models are 75.24%, 92.29%, 98.5%, and 98.1%, respectively. The corresponding mean of relative errors are 2.61%, 0.973%, 0.071%, and 0.104%, respectively, while the corresponding standard deviation of relative errors (STDEV) are 6.8%, 4.8%, 2.31%, and 2.89%, respectively. On the other hand, most of the outliers, with more than 5% relative error, that were generated using the proposed NN and SVR models have very small capacitance values (<1e-4 fF).

As for runtime comparisons, the total runtimes of extracting (i.e., computing) all cross-sections (i.e., 2.07M patterns) using the rule-based, ratio-based, NN, and SVR cross-section models are 16.07, 19.27, 6.7, and 6.1 hours, respectively. Therefore, the corresponding runtimes relative to rule-based models are 1, 1.2, 0.417, and 0.38, respectively. The capacitance computations were done on a single CPU using Intel

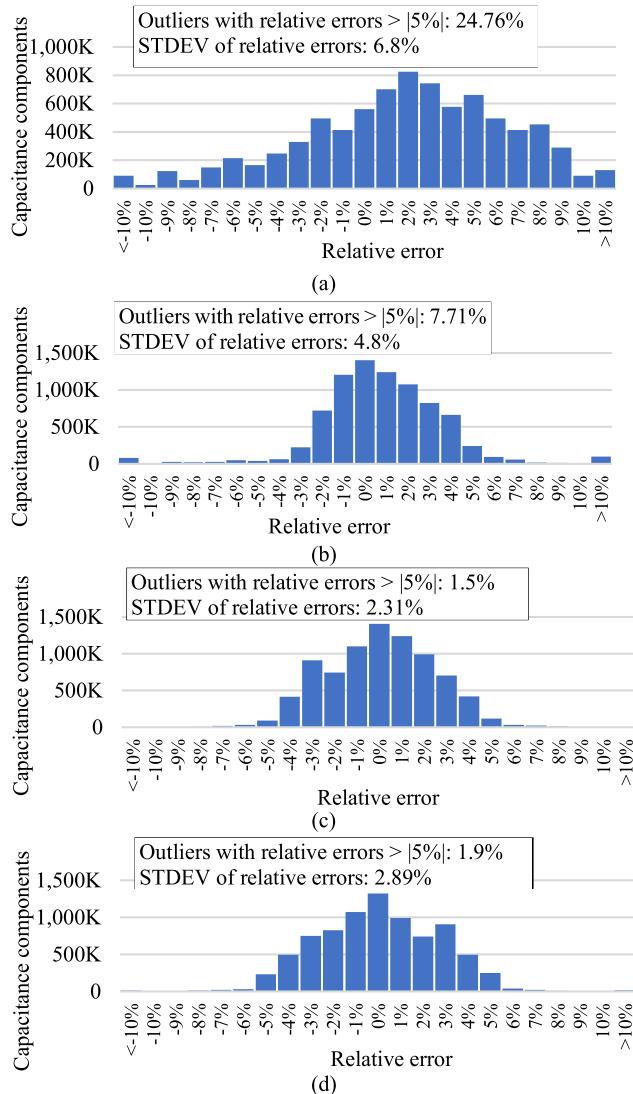


FIGURE 19. Relative error histograms of extracted capacitances for 2D cross-patterns of 28nm designs using the (a) existing rule-based, (b) ratio-based, (c) proposed NN, and (d) proposed SVR cross-section models.

Xeon(R) E5-2680, 2.50GHz, and 16G of RAM. As a result, the generated models (i.e., NN and SVR models) managed to achieve high accuracy results as compared to existing rule-based cross-section models and ratio-based models, in [6], with an average speed up of 2.5X.

B. TESTING REAL DESIGNS OF 14nm PROCESS NODE

The total number of generated models (either NN or SVR) is 175. The generated models cover 175 different metal collections each includes 1 to 5 different metal layers. Each model (i.e., NN or SVR model) was trained over 30K cross-section patterns, where 21K patterns (70%) were used for the training set, and 9K patterns (30%) were used for the test set. As for NN models, the total training runtime of all models is 21.7 hours. As for SVR models, the total training runtime of all models is 13.9 hours.

TABLE 11. The accuracy and relative errors of test sets for NN and SVR models of 14nm process node.

	NN models	SVR models
Mean of relative errors	0.042%	0.061%
Standard deviation of relative errors	2.21%	2.75%
Outliers with relative error > 5%	1.77%	1.97%
Mean square error	0.00237	0.00316

Table 11 shows the test sets accuracy of NN and SVR models. The results show that the NN and SVR models provide high accuracy values, where almost 98% of the extracted capacitances have relative errors below 5%.

As for testing the generated models on real design patterns of 14nm process node, the generated models were tested over cross-section patterns of three test chips including cache memory, DRAM, and VCO designs that were not included during the training processes. The total numbers of extracted cross-section patterns in cache memory, DRAM, and VCO designs are 630K, 915K, and 1.03M patterns, respectively. The corresponding total number of capacitance components are 2.8M, 4M, and 4.4M capacitances, respectively. Therefore, the total number of extracted cross-section patterns is 2.575M patterns, and the total number of extracted capacitances is 11.2M.

Figure 20 shows histograms of relative errors covering all extracted capacitances across all designs using the rule-based extraction, ratio-based, proposed NN, proposed SVR cross-section models. The accuracy comparisons show that the proposed NN and SVR models provide high accuracy results as compared to existing rule-based cross-section models and ratio-based models. The percentages of extracted capacitance components with relative errors below 5% using the rule-based, ratio-based, proposed NN, proposed SVR cross-section models are 74.3%, 87.4%, 98.03%, and 97.81%, respectively. The corresponding mean of relative errors are 2.36%, 1.81%, 0.107%, and 0.148%, respectively, while the corresponding standard deviation of relative errors (STDEV) are 6.94%, 5.82%, 2.58%, and 3.07%, respectively. On the other hand, most of the outliers, with more than 5% relative error, that were generated from the proposed NN and SVR models have very small capacitance values (<1e-4 fF).

As for runtime comparisons, the total runtimes of extracting (i.e., computing) all cross-section patterns (i.e., 2.575M patterns) using the rule-based, ratio-based, proposed NN, proposed SVR cross-section models are 20.03, 24.1, 8.32, and 7.84 hours, respectively. Therefore, the corresponding runtimes relative to rule-based models are 1, 1.203, 0.415, and 0.391, respectively. The capacitance computations were done on a single CPU using Intel Xeon(R) E5-2680, 2.50GHz, and 16G of RAM. As a result, the generated models (i.e., NN and SVR models) managed to achieve high accuracy results as compared to existing rule-based cross-section models and ratio-based models, in [6], with an average speed up of 2.45X.

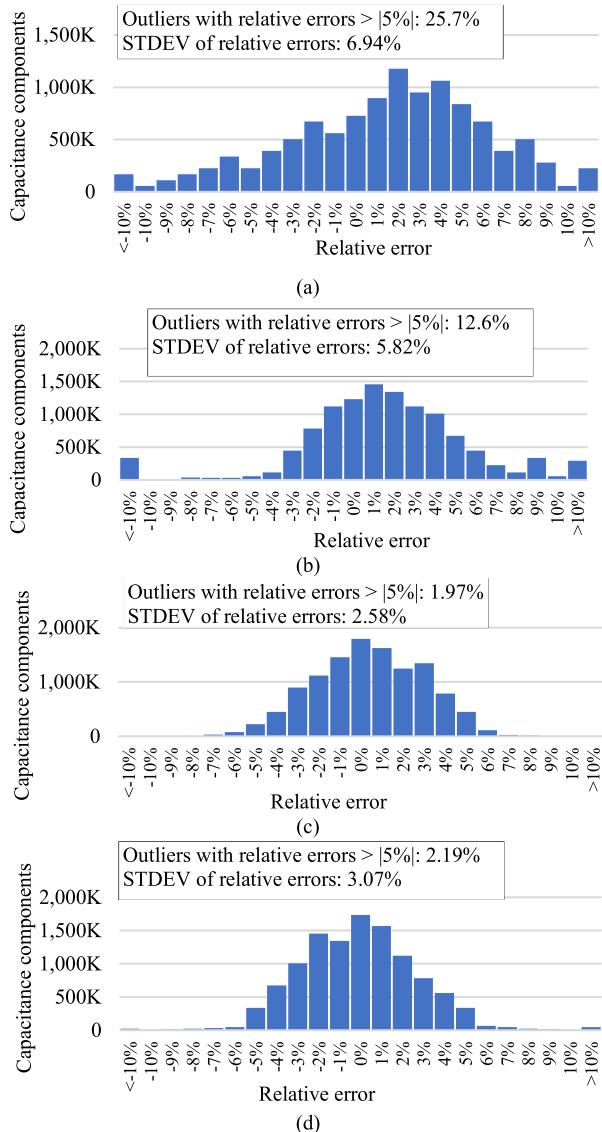


FIGURE 20. Relative error histograms of extracted capacitances for 2D cross-patterns of 14nm designs using the (a) existing rule-based, (b) ratio-based, (c) proposed NN, and (d) proposed SVR cross-section models.

C. TESTING REAL DESIGNS OF 7nm PROCESS NODE

The total number of generated models (either NN or SVR) is 231. The generated models cover 231 different metal collections each includes 1 to 5 different metal layers. Each model (i.e., NN or SVR model) was trained over 30K cross-section patterns, where 21K patterns (70%) were used for the training set, and 9K patterns (30%) were used for the test set. As for NN models, the total training runtime of all models is 23.01 hours. As for SVR models, the total training runtime of all models is 15.03 hours. The training runtimes can significantly improve by multi-processing. It is worth mentioning that the models were generated only once for each process node. After that, the generated models are used numerous times by parasitic extraction tools.

TABLE 12. The accuracy and relative errors of test sets for NN and SVR models of 7nm process node.

	NN models	SVR models
Mean of relative errors	0.065%	0.083%
Standard deviation of relative errors	3.02%	3.17%
Outliers with relative error > 5%	2.31%	2.6%
Mean square error	0.0052	0.0063

Table 12 shows the test sets accuracy of NN and SVR models. The results show that the NN and SVR models provide high accuracy values, where almost 97% of the extracted capacitances have relative errors below 5%.

As for testing the generated models on real design patterns of 7nm process node, the generated models were tested over cross-section patterns of two test chips including cache memory (CM) and VCO designs that were not included during the training processes. The total numbers of cross-section patterns of cache memory and VCO designs are 920K and 1.17M patterns, respectively. The corresponding total number of capacitance components are 4.1M and 5M capacitances, respectively. Therefore, the total number of extracted cross-section patterns is 2.09M patterns, and the total number of extracted capacitances is 9.1M.

Figure 21 shows histograms of relative errors covering all extracted capacitances across all designs using the rule-based extraction, ratio-based, proposed NN, proposed SVR cross-section models. The accuracy comparisons show that the proposed NN and SVR models provide high accuracy results as compared to existing rule-based cross-section models and ratio-based models. The percentages of extracted capacitance components with relative errors below 5% using rule-based, ratio-based, NN, and SVR cross-section models are 71.1%, 76.7%, 97.47%, and 96.8%, respectively. The corresponding mean of relative errors are 2.51%, 2.6%, 0.11%, and 0.15%, respectively, while the corresponding standard deviation of relative errors (STDEV) are 8.31%, 7.01%, 2.65%, and 3.29%, respectively. On the other hand, most of the outliers, with more than 5% relative error, that were generated from the proposed NN and SVR models have very small capacitance values (<1e-4 fF).

As for runtime comparisons, the total runtimes of extracting all cross-sections (i.e., 2.09M patterns) using the rule-based, ratio-based, proposed NN, proposed SVR cross-section models are 16.26, 19.46, 6.93, and 6.81 hours, respectively. Hence, the corresponding runtimes relative to rule-based models are 1, 1.197, 0.43, and 0.419, respectively. The capacitance computations are done on a single CPU using Intel Xeon(R) E5-2680, 2.50GHz, and 16G of RAM. As a result, the generated models (i.e., NN and SVR models) managed to achieve high accuracy results as compared to existing rule-based cross-section models and ratio-based models, in [6], with an average speed up of 2.35X.

As for the uncertainty of predictions, there are two types of uncertainties that include aleatoric and epistemic uncertainty. In interconnect parasitic capacitance extraction, the aleatoric uncertainty represents the irreducible part of the

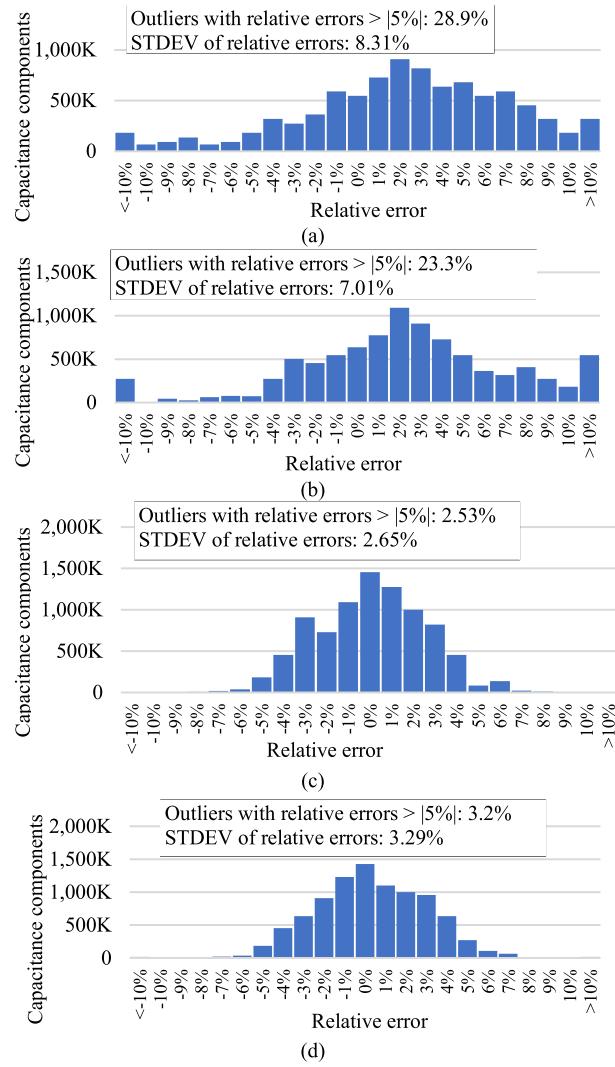


FIGURE 21. Relative error histograms of extracted capacitances for 2D cross-patterns of 7nm designs using the (a) existing rule-based, (b) ratio-based, (c) proposed NN, and (d) proposed SVR cross-section models.

uncertainty that might occur due to the random process variations [29]–[32]. On the other hand, the epistemic uncertainty represents the reducible part of the uncertainty. It occurs due to limited coverages of training patterns, for example, input cross-section patterns that violate the corresponding design rules may not be part of training patterns. The epistemic uncertainty can be reduced by increasing the coverage of training patterns.

D. STATISTICAL TESTS

Nonparametric statistical tests were performed to test the significant difference in performance (i.e., accuracy) between each two models. The Wilcoxon signed-ranks test [43] was selected because it is a nonparametric statistical test that is performed to test the significant difference between two models (i.e., paired comparisons). In our case, the null hypothesis indicates a lack of a significant difference between the two tested models. The null hypothesis will be rejected if the

TABLE 13. Accuracy comparisons in terms of mean square errors for rule-based, ratio-based, the proposed SVR, and the proposed NN models.

Dataset	Mean square error (MSE)			
	Commercial rule-based [25]	Ratio-based [6]	The proposed SVR	The proposed NN
RO (28nm)	0.00710	0.00231	0.00141	0.00173
DRAM(28nm)	0.00930	0.00213	0.00219	0.00201
SRAM(28nm)	0.00120	0.00160	0.00173	0.00233
DAC(28nm)	0.00760	0.00432	0.00201	0.00190
VCO(28nm)	0.00314	0.00470	0.00379	0.00211
RO(14nm)	0.00301	0.00217	0.00296	0.00237
DAC(14nm)	0.00430	0.00523	0.00327	0.00190
DRAM(14nm)	0.00810	0.00455	0.00316	0.00385
VCO(14nm)	0.01300	0.00710	0.00411	0.00310
RO(7nm)	0.05010	0.10810	0.00510	0.00387
SRAM(7nm)	0.05230	0.05030	0.00471	0.00490
CM(7nm)	0.08150	0.07711	0.00720	0.00590
VCO(7nm)	0.10300	0.13800	0.00811	0.00610

TABLE 14. Paired comparisons using Wilcoxon signed-rank test (two-tailed) to test the significant difference between each two models, where the mean square error, against a field-solver, is used as a performance metric.

Pairwise comparison		SNR	SPR	z-value	p-value	Significance
Model1	Model2					
The proposed NN	The proposed SVR	71	20	-1.7821	0.07508	No
The proposed NN	Commercial rule-based [25]	88	3	-2.9701	0.00298	Yes
The proposed NN	Ratio-based [6]	84	7	-2.6906	0.00714	Yes
The proposed SVR	Commercial rule-based [25]	86	5	-2.8304	0.00466	Yes
The proposed SVR	Ratio-based [6]	85	6	-2.7605	0.00578	Yes

SNR: Sum of negative ranks; SPR: Sum of positive ranks.

p-value is less than 0.05 (*p*-value < 0.05). The mean square error (MSE) was used as a performance metric to help in performing statistical tests. MSE values were obtained for the four extraction models over 13 datasets using Raphael, 2D field-solver, as a reference, as shown in Table 13.

Table 14 shows statistical comparisons using Wilcoxon signed-rank tests. The table shows the *p*-value and *z*-value for each paired comparison test. Also, the table shows the sum of positive ranks (SPR) and sum of negative ranks (SNR) for each paired comparison test. The comparisons show that there is no significant difference between the proposed NN and SVR models as the *p*-value is greater than 0.05. However, the results show significant differences (i.e., rejecting the

null hypothesis) between the proposed models and each compared extraction model as the p -values are less than 0.05.

VI. CONCLUSION AND FUTURE WORK

A novel modeling methodology for interconnect parasitic capacitances is developed for rule-based extraction tools using machine learning methods. The proposed methodology managed to overcome several problems in rule-based extraction tools such as handling systematic process variations, high pattern mismatches, and limited pattern coverages. The proposed methodology creates cross-section compact models for a certain process technology node. Such compact models predict the parasitic coupling capacitances between metal polygons on a given 2D cross-section layout pattern considering the impact of systematic process variations. The modeling methodology process starts with processing process stack specifications to identify the main characteristics of layout input patterns, such as pattern's size, the maximum number of metal layers in a pattern, handling multi-dielectric stacks, systematic process variations, and the maximum number of polygons in a pattern. The input of the compact models is a given cross-section pattern including the required capacitances and the corresponding systematic process variations. The patterns are represented by a novel vertex-based pattern representation that considers systematic process variations as a part of the geometrical characteristics of a given pattern. The compact models are implemented using two different machine learning methods: neural networks and support vector regression methods. The proposed methodology is tested over thirteen real designs of 28nm, 14nm, and 7nm process nodes with more than 6.7M interconnect patterns. The generated compact models are faster than traditional rule-based models by 2.5X. Also, they managed to achieve outstanding results as compared to field-solvers and rule-based cross-section models, where the average relative error of the generated models is $< 0.5\%$ and the standard deviation of relative errors is $<.31\%$.

As for a future work, the proposed models cannot predict parasitic capacitances of three-dimensional integrated circuits (3DIC) technologies, such as stacked-die 3DIC and monolithic 3DIC technologies. The 3DIC technologies aims to combine and integrate multiple systems on a single package. In stacked-die 3DIC technologies, multiple silicon wafers (or chips) are stacked vertically and connected together by using a through-silicon-via (TSV). The stacking may have many forms, such as a face to face or a face to back. In such cases, the capacitance coupling interactions among the interconnects across those chips need to be modeled. As for monolithic 3DIC technologies, the device layers and their corresponding devices are fabricated sequentially, and multiple devices with different elevations may exist. In such cases, there are many different metal and device layers that are vertically overlapped, and the parasitic capacitances among them need to be modeled correctly. Eventually, the proposed models need to be extended to support 3DIC technologies.

REFERENCES

- [1] W. Yu, M. Song, and M. Yang, "Advancements and challenges on parasitic extraction for advanced process technologies," in *Proc. 26th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2021, pp. 841–846.
- [2] *FinFETs, 16 nm and 14 nm Nodes, and Parasitic Extraction*. Accessed: Oct. 12, 2021. [Online]. Available: https://www5.cadence.com/2014-July-Newsletter_LP.html
- [3] J.-H.-C. Chen, T. E. Standaert, E. Alptekin, T. A. Spooner, and V. Paruchuri, "Interconnect performance and scaling strategy at 7 nm node," in *Proc. IEEE Int. Interconnect Technol. Conf.*, May 2014, pp. 93–96, doi: [10.1109/IITC.2014.6831843](https://doi.org/10.1109/IITC.2014.6831843).
- [4] A. Kurokawa, T. Sato, T. Kanamoto, and M. Hashimoto, "Interconnect modeling: A physical design perspective," *IEEE Trans. Electron Devices*, vol. 56, no. 9, pp. 1840–1851, Sep. 2009, doi: [10.1109/TED.2009.2026208](https://doi.org/10.1109/TED.2009.2026208).
- [5] C. Venkataiah, K. S. Prasad, and T. J. C. Prasad, "Effect of interconnect parasitic variations on circuit performance parameters," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Oct. 2016, pp. 1–4, doi: [10.1109/CESYS.2016.7889958](https://doi.org/10.1109/CESYS.2016.7889958).
- [6] M. S. Abouelyazid, S. Hammouda, and Y. Ismail, "Connectivity-based machine learning compact models for interconnect parasitic capacitances," in *Proc. ACM/IEEE 3rd Workshop Mach. Learn. CAD (MLCAD)*, Aug. 2021, pp. 1–6, doi: [10.1109/MLCAD52597.2021.9531300](https://doi.org/10.1109/MLCAD52597.2021.9531300).
- [7] W. Yu and X. Wang, *Advanced Field-Solver Techniques for RC Extraction of Integrated Circuits*. Berlin, Germany: Springer-Verlag, 2014, doi: [10.1007/978-3-642-54298-5](https://doi.org/10.1007/978-3-642-54298-5).
- [8] A. Zhang, W. Zhao, Y. Hu, J. He, Q. He, L. Song, H. Zhou, and Y. Wu, "A field-based parasitic capacitance model with 3-D terminal and terminal fringe components," in *Proc. 6th Asia Symp. Quality Electron. Design (ASQED)*, Aug. 2015, pp. 166–170, doi: [10.1109/ACQED.2015.7274028](https://doi.org/10.1109/ACQED.2015.7274028).
- [9] A. Zhang, W. Zhao, X. Zhu, W. Deng, J. He, A. Chen, and M. Chan, "Field-based parasitic capacitance models for 2D and 3D sub-45-nm interconnect," in *Proc. 4th Asia Symp. Quality Electron. Design (ASQED)*, Jul. 2012, pp. 110–116, doi: [10.1109/ACQED.2012.6320485](https://doi.org/10.1109/ACQED.2012.6320485).
- [10] *Parasitic Extraction Must Solve Advanced Node Issues—EDN*. Accessed: Oct. 12, 2021. [Online]. Available: <https://www.edn.com/parasitic-extraction-must-solve-advanced-node-issues/>
- [11] W. H. Kao, C.-Y. Lo, M. Basel, and R. Singh, "Parasitic extraction: Current state of the art and future trends," *Proc. IEEE*, vol. 89, no. 5, pp. 729–739, May 2001, doi: [10.1109/5.929651](https://doi.org/10.1109/5.929651).
- [12] X. Qi, A. Gyure, Y. Luo, S. C. Lo, M. Shahram, and K. Singhal, "Measurement and characterization of pattern dependent process variations of interconnect resistance, capacitance and inductance in nanometer technologies," in *Proc. 16th ACM Great Lakes Symp. VLSI (GLSVLSI)*, Philadelphia, PA, USA, 2006, p. 14, doi: [10.1145/1127908.1127914](https://doi.org/10.1145/1127908.1127914).
- [13] N. K. Karsilayan, J. Falbo, and D. Petranovic, "Efficient and accurate RIE modeling methodology for BEOL 2.5D parasitic extraction," in *Proc. IEEE 57th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2014, pp. 519–522, doi: [10.1109/MWSCAS.2014.6908466](https://doi.org/10.1109/MWSCAS.2014.6908466).
- [14] N. K.-H. Huang, "Implementation of algorithms to determine the capacitance sensitivity of interconnect parasitics in the Magic VLSI layout tool," M.S. thesis, Dept. Elect. Eng., Univ. British Columbia, 2009. Accessed: Nov. 29, 2017. [Online]. Available: <https://open.library.ubc.ca/collections/ubctheses/24/items/1.0067690>
- [15] N. Huang and A. Labun, "Extracting interconnect capacitance sensitivity to linewidth variation," in *Proc. Can. Conf. Electr. Comput. Eng.*, May 2009, pp. 185–189, doi: [10.1109/CCECE.2009.5090117](https://doi.org/10.1109/CCECE.2009.5090117).
- [16] C. Zhang and G. Sun, "Fabrication cost analysis for 2D, 2.5D, and 3D IC designs," in *Proc. IEEE Int. 3D Syst. Integr. Conf. (3DIC)*, Jan. 2012, pp. 1–4, doi: [10.1109/3DIC.2012.6263032](https://doi.org/10.1109/3DIC.2012.6263032).
- [17] S. Wong, A. El-Gamal, P. Griffin, Y. Nishi, F. Pease, and J. Plummer, "Monolithic 3D integrated circuits," in *Proc. Int. Symp. VLSI Technol. Syst. Appl. (VLSI-TSA)*, Apr. 2007, pp. 1–4, doi: [10.1109/VTSA.2007.378923](https://doi.org/10.1109/VTSA.2007.378923).
- [18] J. H. Lau, "Evolution, challenge, and outlook of TSV, 3D IC integration and 3D silicon integration," in *Proc. Int. Symp. Adv. Packag. Mater. (APM)*, Oct. 2011, pp. 462–488, doi: [10.1109/ISAPM.2011.6105753](https://doi.org/10.1109/ISAPM.2011.6105753).
- [19] K.-Y. Tsai, W.-J. Hsieh, Y.-C. Lu, B.-S. Chang, S.-W. Chien, and Y.-C. Lu, "A new method to improve accuracy of parasitics extraction considering sub-wavelength lithography effects," in *Proc. 15th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2010, pp. 651–656, doi: [10.1109/ASPDAC.2010.5419805](https://doi.org/10.1109/ASPDAC.2010.5419805).

- [20] W. Gong, W. Yu, Y. Lü, Q. Tang, Q. Zhou, and Y. Cai, "A parasitic extraction method of VLSI interconnects for pre-route timing analysis," in *Proc. Int. Conf. Commun., Circuits Syst. (ICCCAS)*, Jul. 2010, pp. 871–875, doi: [10.1109/ICCCAS.2010.5581853](https://doi.org/10.1109/ICCCAS.2010.5581853).
- [21] B. Cardoso, R. Martins, N. Lourenço, and N. Horta, "AIDA-PEx: Accurate parasitic extraction for layout-aware analog integrated circuit sizing," in *Proc. 11th Conf. Ph.D. Res. Microelectron. Electron. (PRIME)*, Jun. 2015, pp. 129–132, doi: [10.1109/PRIME.2015.7251351](https://doi.org/10.1109/PRIME.2015.7251351).
- [22] Z. Li and W. Shi, "Layout capacitance extraction using automatic pre-characterization and machine learning," in *Proc. 21st Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2020, pp. 457–464, doi: [10.1109/ISQED48828.2020.9136970](https://doi.org/10.1109/ISQED48828.2020.9136970).
- [23] S.-C. Wong, G.-Y. Lee, and D.-J. Ma, "Modeling of interconnect capacitance, delay, and crosstalk in VLSI," *IEEE Trans. Semicond. Manuf.*, vol. 13, no. 1, pp. 108–111, Feb. 2000, doi: [10.1109/66.827350](https://doi.org/10.1109/66.827350).
- [24] R. Kasai, T. Kanamoto, M. Imai, A. Kurokawa, and K. Hachiya, "Neural network-based 3D IC interconnect capacitance extraction," in *Proc. 2nd Int. Conf. Commun. Eng. Technol. (ICCET)*, Apr. 2019, pp. 168–172, doi: [10.1109/ICCET.2019.8726919](https://doi.org/10.1109/ICCET.2019.8726919).
- [25] Siemens Digital Industries Software. *Calibre xRC*. Accessed: Oct. 12, 2021. [Online]. Available: <https://eda.sw.siemens.com/en-US/ic/calibre-design/circuit-verification/xrc/>
- [26] Synopsys *StarRC—Golden Signoff Extraction*. Accessed: Oct. 12, 2021. [Online]. Available: <https://www.synopsys.com/implementation-and-signoff/signoff/starrc.html>
- [27] L. Lavagno, L. Scheffer, and G. Martin, *EDA for IC Implementation, Circuit Design, and Process Technology*. Boca Raton, FL, USA: CRC Press, 2006.
- [28] K. J. Kuhn, M. D. Giles, D. Becher, P. Kolar, A. Kornfeld, R. Kotlyar, S. T. Ma, A. Maheshwari, and S. Mudanai, "Process technology variation," *IEEE Trans. Electron Devices*, vol. 58, no. 8, pp. 2197–2208, Aug. 2011, doi: [10.1109/TED.2011.2121913](https://doi.org/10.1109/TED.2011.2121913).
- [29] Y.-S. Song, C.-Y. Chu, J. Jeon, U.-H. Kwon, K.-H. Lee, and S. Kim, "Accurate BEOL statistical modeling methodology with circuit-level multi-layer process variations," in *Proc. Int. Conf. Simulation Semiconductor Processes Devices (SISPAD)*, Sep. 2017, pp. 265–268, doi: [10.23919/SISPAD.2017.8085315](https://doi.org/10.23919/SISPAD.2017.8085315).
- [30] J.-H. Liu, J.-K. Zeng, A.-S. Hong, L. Chen, and C. C. P. Chen, "Process-variation statistical modeling for VLSI timing analysis," in *Proc. 9th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2008, pp. 730–733, doi: [10.1109/ISQED.2008.4479828](https://doi.org/10.1109/ISQED.2008.4479828).
- [31] V. Mehrotra, "Modeling the effects of systematic process variation on circuit performance," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2001.
- [32] L. Sun, W. Wong, H. H. Tseng, R. Shi, X. Jiang, J. Fan, and G. Wan, "A novel customized RC tightened corner modeling methodology using statistical SPICE simulation in advanced FinFET technology," in *Proc. 14th IEEE Int. Conf. Solid-State Integr. Circuit Technol. (ICSICT)*, Oct. 2018, pp. 1–4, doi: [10.1109/ICSICT.2018.8564960](https://doi.org/10.1109/ICSICT.2018.8564960).
- [33] S. Kiamehr, P. Weckx, M. Tahoori, B. Kaczer, H. Kukner, P. Raghavan, G. Groeseneken, and F. Catthoor, "The impact of process variation and stochastic aging in nanoscale VLSI," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2016, pp. CR-1-1-CR-1-6, doi: [10.1109/IRPS.2016.7574590](https://doi.org/10.1109/IRPS.2016.7574590).
- [34] R. Sharma, Nitin, V. K. Sehgal, and D. S. Chauhan, "Closed-form expressions for extraction of capacitances in multilayer VLSI interconnects," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2008, pp. 1–4. [Online]. Available: <https://ieeexplore.ieee.org/document/4766573>, doi: [10.1109/TENCON.2008.4766573](https://doi.org/10.1109/TENCON.2008.4766573).
- [35] S.-C. Wong, T. G.-Y. Lee, D.-J. Ma, and C.-J. Chao, "An empirical three-dimensional crossover capacitance model for multilevel interconnect VLSI circuits," *IEEE Trans. Semicond. Manuf.*, vol. 13, no. 2, pp. 219–227, May 2000, doi: [10.1109/66.843637](https://doi.org/10.1109/66.843637).
- [36] A. Kurokawa, "Second-order polynomial expressions for on-chip interconnect capacitance," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E88-A, no. 12, pp. 3453–3462, Dec. 2005, doi: [10.1093/ietfec/e88-a.12.3453](https://doi.org/10.1093/ietfec/e88-a.12.3453).
- [37] T. Sakurai and K. Tamaru, "Simple formulas for two- and three-dimensional capacitances," *IEEE Trans. Electron Devices*, vol. ED-30, no. 2, pp. 183–185, Feb. 1983, doi: [10.1109/T-ED.1983.21093](https://doi.org/10.1109/T-ED.1983.21093).
- [38] E. Demircan, "Effects of interconnect process variations on signal integrity," in *Proc. IEEE Int. SOC Conf.*, Sep. 2006, pp. 281–284, doi: [10.1109/SOCC.2006.283898](https://doi.org/10.1109/SOCC.2006.283898).
- [39] Synopsys. *Raphael—Technology Computer Aided Design (TCAD)*. Accessed: Oct. 12, 2021. [Online]. Available: <https://www.synopsys.com/silicon/tcad/interconnect-simulation/raphael.html>
- [40] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1026–1034, doi: [10.1109/ICCV.2015.123](https://doi.org/10.1109/ICCV.2015.123).
- [41] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, Mar. 2010, pp. 249–256. Accessed: Nov. 30, 2021. [Online]. Available: <https://proceedings.mlr.press/v9/glorot10a.html>
- [42] *TensorFlow*. Accessed: Mar. 25, 2021. [Online]. Available: <https://www.tensorflow.org/>
- [43] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945, doi: [10.2307/3001968](https://doi.org/10.2307/3001968).



MOHAMED SALEH ABOUELYAZID (Member, IEEE) received the B.Sc. and M.Sc. degrees in electronics engineering from the Faculty of Engineering, Cairo University, Egypt, in 2009 and 2013, respectively. He is currently pursuing the Ph.D. degree with the School of Science and Engineering, The American University in Cairo, Egypt. He has been working with Siemens EDA, Egypt, since 2009, where he is currently a Research and Development Staff Engineer in layout parasitic extraction team, Calibre design to silicon. He has a strong background in circuit verification and electronic design automation.



SHERIF HAMMOUDA received the B.Sc. and M.Sc. degrees in electrical engineering from Ain Shams University, Cairo, Egypt, in 1996 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Calgary, AB, Canada, in 2006. He has been with Siemens EDA, Cairo, since 1997, where he is also the Engineering Director managing research and development activities for Calibre design to silicon activities in Cairo. He holds a U.S. patent and coauthored a number of publications in various conferences focusing on EDA for IC Design.



YEHEIA ISMAIL (Fellow, IEEE) is currently the Director of the Nanoelectronics and Devices Center, The American University in Cairo and Zewail City. He was a Tenured Professor with Northwestern University, USA, from 2000 to 2011. He has published more than 450 papers in top refereed journals and conferences and many patents. He has coauthored seven books, including *On-Chip Inductance in High Speed Integrated Circuits*, *Handbook on Algorithms for VLSI Physical Design*, *Temperature-Aware Computer Architecture*, *Arbitrary Modeling of TSVs for 3D Integrated Circuits*, and *Circuit Design Techniques for Microscale Energy Harvesting Systems*. He has many patents in the area of high performance circuits and interconnect design and modeling. His work is some of the most highly cited in the VLSI area and is extensively used by industry. He has several awards, such as the USA National Science Foundation Career Award, the IEEE CAS Outstanding Author Award, the Best Teacher Award at Northwestern University, and many other best teaching awards and best paper awards. He is a Distinguished Lecturer of IEEE CASS. He was the Editor-in-Chief of the IEEE TRANSACTION ON VERY LARGE SCALE INTEGRATION (TVLSI) (2011–2015) and the Chair Elect of the IEEE VLSI Technical Committee. He is on the editorial board of the *Journal of Circuits, Systems, and Computers*, was on the editorial board of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: FUNDAMENTAL THEORY AND APPLICATIONS, and a Guest Editor for a special issue of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS on "On-Chip Inductance in High Speed Integrated Circuits." He has also chaired many conferences, such as GLSVLSI, IWSOC, and ISCAS. He was the Chief Scientist of the Innovation and Entrepreneurship Center, Ministry of Communications and Information Technology, Egypt.