Machine Learning-Based Hotspot Detection: Fallacies, Pitfalls and Marching Orders

Gaurav Rajavendra Reddy[1], Kareem Madkour[2] and Yiorgos Makris[1]

gaurav.reddy@utdallas.edu, kareem_madkour@mentor.com, yiorgos.makris@utdallas.edu
[1]Department of Electrical and Computer Engineering, The University of Texas at Dallas, USA
[2]Mentor, A Siemens Business, Egypt

***Abstract*—Extensive technology scaling has not only increased the complexity of Integrated Circuit (IC) fabrication but also multiplied the challenges in the Design For Manufacturability (DFM) space. Among these challenges, detection of design weak-points, popularly known as 'Lithographic Hotspots', has attracted substantial attention. Hotspots are certain patterns which exhibit a higher probability of causing defects due to complex design-process interactions. Identifying such patterns and fixing them in the design stage itself is imperative towards ensuring high yield. In the early days of hotspot detection, Pattern Matching (PM) based methods were proposed. While effective in identifying previously known patterns, these methods failed to identify Never-Seen-Before (NSB) hotspots. To address this drawback, Machine Learning (ML) based solutions were introduced. Over the last decade, we have witnessed a plethora of ML-based hotspot detection methods being developed, each slightly outperforming its predecessors in accuracy and false-alarm rates. In this paper, we critically analyze the ML-based hotspot detection literature and we highlight common misconceptions which are found therein. We also pinpoint the underlying reasons that have led to these misconceptions by dissecting the ICCAD-2012 benchmark dataset, which has largely guided the evolution of this area, and revealing its limitations. Furthermore, we propose an enhanced version of this benchmark dataset, which we deem more appropriate for accurately assessing hotspot detection methods. Finally, we offer our suggestions to improve the effectiveness of ML-based Hotspot Detection methods and demonstrate about 5X reduction in false-alarms in comparison to the state-of-the-art.**

I. Introduction

Constant scaling and reduction in feature sizes has made lithography more complex than ever before. Despite employing advanced Resolution Enhancement Techniques (RETs) such as Optical Proximity Correction (OPC), multi-patterning, phase shifted masks, etc., fabricating certain layout patterns with high fidelity remains challenging. Such layout patterns are commonly referred to as *design weak-points* or *hotspots* and stem from a broad variety of underlying root causes, making it impractical if not impossible to tune the process in order to ensure their manufacturability. As a result, over the last decade, *Lithographic Hotspot Detection* became an area of intense interest, with a variety of methods seeking to identify and fix the hotspots in a given layout well before tape-out [1], thereby avoiding defects during fabrication and ensuring high yield. Initial solutions relied on Pattern Matching (PM) based methods [2] [3]. Such methods were found to be fast and efficient in scanning through new layouts and identifying previously seen hotspots; however, they failed to identify Never-Seen-Before (NSB) hotspots. To address this drawback, Machine Learning (ML) based methods were proposed [4] [5]. Their basic idea is to use ML algorithms to 'learn' from a database of known hotspots and then make predictions on a new layout. These methods are believed to hold great promise in achieving high hotspot prediction rates (on both previously known, and NSB hotspots) while keeping false positives at a minimum.

Since their initial proposition, several flavors of ML-based hotspot detection methods have been developed. Most of them exhibited slight improvements in hotspot detection accuracy and/or reduction in false alarm rate over prior works. Such improvements were mostly obtained by using increasingly powerful ML algorithms [6] [7], advanced feature extraction methods [5] [8] and/or hybrid PM-ML approaches [9]. More recently, many researchers have proposed sophisticated hotspot detection methods using online learning [8], deep learning [10] [11][12][13], litho-aware learning [14], etc. Such methods show hotspot detection accuracy rates upwards of 98% with false alarm rates less than 0.5%, and claim that they can effectively detect NSB patterns while maintaining low false alarm rates. Despite their impeccable results, *we posit that most State-Of-the-Art (SOTA) methods are ineffective in reducing false alarms and are incapable of detecting Truly-Never-Seen-Before (TNSB) patterns*. Moreover, *our conjecture is that such shortcomings of the SOTA methods have not been apparent because most of them have been tested using the ICCAD-2012 benchmark dataset*.

The ICCAD-2012 benchmarks are derived from the ICCAD-2012 fuzzy pattern matching contest [15]. While these benchmarks were originally developed to evaluate fuzzy PM-based hotspot detection methods, over the years they have become the *de facto* standard in comparing various ML-based hotspot detection methods. However, the literature lacks a formal analysis of this dataset's characteristics and/or a justification of its appropriateness as an effective benchmark for evaluating ML-based hotspot detection methods. To the contrary, upon extensive analysis, we find that the test-set of this benchmark dataset contains neither TNSB patterns nor Hard-To-Classify (HTC) patterns, which would test the true resilience of ML-based hotspot detection methods against false-alarms. Therefore, the validity of the claims made by the

978-1-7281-2350-9/19/$31.00

various SOTA methods comes into question.

In this work, we present our views on the capabilities and limitations of ML-based methods and we contrast them to the communal opinion. We highlight common *fallacies* pervasively found in the literature, we discuss the *pitfalls* that led to these fallacies, and we offer our *marching orders* towards improving effectiveness of ML-based hotspot detection methods and accuracy of their evaluation. Specifically, the main contributions of this paper include:

1) A detailed description of three key misconceptions regarding ML-based hotspot detection:
   a) **Fallacy 1:** Hotspot detection on the ICCAD-2012 benchmark dataset is difficult.
   b) **Fallacy 2:** ML-based hotspot detection methods can detect NSB hotspot patterns.
   c) **Fallacy 3:** The SOTA methods can effectively prevent false alarms.
2) A detailed explanation of the underlying pitfalls, stemming from the use of the ICCAD-2012 benchmark dataset:
   a) **Pitfall 1:** The dataset does not contain TNSB patterns, hence using it to evaluate hotspot detection rate is misleading.
   b) **Pitfall 2:** The dataset does not contain HTC patterns, hence using it to evaluate false alarm rate is misleading.
3) Our suggestions for accurate evaluation and improvement of ML-based hotspot detection methods:
   a) **Marching Order 1:** Introduction of an enhanced version of the ICCAD-2012 benchmark dataset which does not suffer from the aforementioned limitations.
   b) **Marching Order 2:** Proposition of Early Design Space Exploration (EDSE) as a plausible direction for achieving detection of TNSB hotspot patterns.
   c) **Marching Order 3:** Proposition of synthetic training set enhancement as a plausible direction for root cause learning and false alarm reduction.

The rest of the paper is organized as follows: First, in Section II, we review the commonly used ICCAD-2012 benchmark dataset, we discuss its limitations, and we introduce a derivative thereof, which we use to support the claims made in this paper and to facilitate more accurate evaluation of hotspot detection methods. Then, in Sections III, IV, and V, we discuss the three main fallacies reflected in the ML-based hotspot detection literature. Within each section, we briefly present the claims of previous methods, we provide a detailed qualitative reasoning for our objections, and we quantitatively support them with experimental results. Moreover, where relevant, we also pinpoint the underlying pitfalls leading to these fallacies and we provide marching orders toward development of more effective hotspot detection methods and more accurate evaluation strategies. Conclusions are drawn in Section VI.

TABLE I: ICCAD-2012 benchmark statistics.

	Training Dataset		Testing Dataset	
	Hotspots #	**Non-Hotspots #**	**Hotspots #**	**Non-Hotspots #**
Benchmark1	99	340	226	319
Benchmark2	174	5285	498	4146
Benchmark3	909	4643	1808	3541
Benchmark4	95	4452	177	3386
Benchmark5	26	2716	41	2111

TABLE II: Proposed ICCAD-2019 benchmark statistics.

	Hotspots #	**Non-Hotspots #**
Training dataset	467	17758
Testing dataset - I	1001	14621
Testing dataset - II	64310	65523

II. Current & Proposed Benchmarks

The ICCAD-2012 dataset is the most widely used benchmark suite for evaluating ML-based hotspot detection methods. Specifically, they are used to test two aspects of such methods: (1) their ability to detect NSB patterns, and (2) their ability to keep false alarms at a minimum. However, as demonstrated in later sections of this paper, these benchmarks lack the types of patterns necessary to test ML-based methods on these two criteria. Therefore, to fill the void created by the shortcomings of the ICCAD-2012 benchmarks and to determine the true state of ML-based hotspot detection, we propose an improved ICCAD-2019 version of these benchmarks.

Statistics on the data distribution of the ICCAD-2012 benchmarks and the proposed ICCAD-2019 benchmarks are provided in Tables I and II, respectively. The ICCAD-2012 dataset consists of five benchmarks, each consisting of a prescribed Training Dataset and a Testing Dataset. These datasets are comprised of patterns from 2 different Product Design Kits (PDKs). The vast majority of these patterns are obtained from a 28nm PDK, while less than 3% are taken from a 32nm PDK.

The proposed ICCAD-2019 benchmarks consists of one training dataset and two different testing datasets, all of them coming from the same 28nm PDK used in the ICCAD-2012 benchmarks. For the purpose of uniformity and considering that a small number of 32nm patterns do not play a significant role in evaluating ML-based methods, we opted to omit them in the new dataset. The 'Training dataset' and the 'Testing dataset - I' include a subset of patterns from the ICCAD-2012 benchmarks. The evident reduction in the number of hotspots is due to the more recent lithographic models used in simulating this dataset, which identify only the more severe defects causing regions to be hotspots. On the other hand, the 'Testing dataset - II' consists of patterns not found in the ICCAD-2012 benchmarks. These new patterns are prepared to ensure that they can effectively test the claims of the contemporary ML-based hotspot detection methods. While the Testing dataset - I is focused towards ratifying the claims of detecting NSB hotspots (as discussed in Section IV), the Testing dataset - II focuses on evaluating robustness against false alarms (as discussed in Section V). The proposed ICCAD-2019 benchmarks are available at [16].

TABLE III: Test results from the ICCAD-2012 dataset.

Benchmark	Hotspots #	Non-Hotspots #	DAC'17 [12]		TCAD'18 [11]		SMACD'18 [13]		Simple ML-based method	
			Accuracy	False Alarm	Accuracy	False Alarm	Accuracy	False Alarm	Accuracy	False Alarm
Benchmark1	226	319	N/A	N/A	N/A	N/A	100%	0.00%	99.56%	0.00%
Benchmark2	498	4146	99.60%	0.17%	99.40%	0.10%	99.80%	0.17%	98.80%	0.00%
Benchmark3	1808	3541	98.06%	0.48%	98.29%	0.28%	99.90%	0.08%	97.84%	0.14%
Benchmark4	177	3386	96.61%	0.03%	95.48%	0.03%	99.80%	0.06%	93.79%	0.00%
Benchmark5	41	2111	97.56%	0.24%	97.56%	0.00%	95.12%	0.05%	100.00%	0.19%
Average			97.96%	0.23%	97.68%	0.10%	98.92%	0.07%	98.00%	0.07%

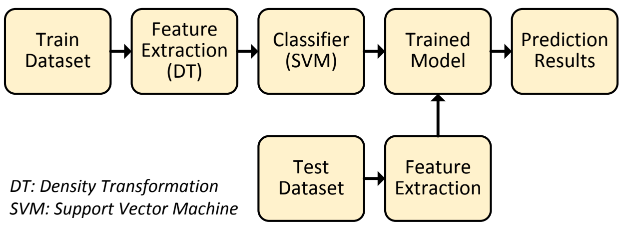

Fig. 1: A simple ML-based Hotspot detection method.

III. Fallacy 1: Hotspot detection on the ICCAD-2012 benchmarks is difficult

Soon after the introduction of the ICCAD-2012 dataset, fuzzy pattern matching methods were proposed [3], achieving acceptable hotspot hit rates but high false alarm rates. Thereafter, many methods including the use of Adaboost classifiers [7], hybrid PM-based and ML-based solutions [9], online learning [8], wire-distance based feature extraction [17], litho-aware learning [14], etc., have shown improved results. More recently, it has been suggested that deep learning is necessary to obtain the best classification results. In [12], authors posit that traditional feature extraction methods, such as density transform [3], suffer from spatial information loss due to their 1-dimensional nature. Therefore, they proposed the use of *feature tensors*, which retain spatial relationships between features, along with biased learning and batch biased learning [11]. In [10], imbalance-aware deep learning has been proposed to address the issue of disproportionate cardinality of hotspots and non-hotspots in the training datasets. Several other sophisticated methods have also been proposed in recent years, achieving some of the best results on the ICCAD-2012 dataset.

Unlike the impression given by the SOTA, we conjecture that hotspot detection on the ICCAD-2012 dataset is actually not as hard as it has been portrayed. Sophisticated methods to address the issue of dataset imbalance, feature extraction methods, deep learning models, etc., are not required to obtain high accuracy and low false alarm rates. In order to demonstrate this, we implement a very simple ML-based flow, as shown in Figure 1. We use density transform [3], one of the simplest feature extraction methods, along with an out-of-the-box Support Vector Machine (SVM) from [18]. We select the hyper parameters through cross-validation, and we set the class weights such that misclassifications on the minority class (i.e., hotspots) are penalized more in comparison to the majority class. In Table III, we compare the results from our simple flow against the results from three SOTA methods[1]. As shown through this comparison, a simple ML-based method provides similar results as the sophisticated deep learning approaches. The source code for this flow is available at [16]. The formulas used in this analysis are:

$$accuracy = \frac{hotspot\_hits}{total\_hotspots}$$

$$false\_alarms = \frac{false\_positives}{total\_nonhotspots}$$

We clarify that, through this analysis, we do not imply that ML-based hotspot detection is an easy problem. To the contrary, we believe that it is not. We also acknowledge that all previously proposed hotspot detection methods have made unique and important contributions which are necessary to improve the overall quality of hotspot detection. However, as we discuss in the next two sections, we posit that the ICCAD-2012 benchmarks used to evaluate them are not effective in accurately reflecting and contrasting their capabilities. As a result, the true benefits of using deep learning and other sophisticated methods, as opposed to simple ML-flows, remain yet to be ascertained.

IV. Fallacy 2: ML-based hotspot detection methods can detect TNSB hotspots

Authors of almost every ML-based hotspot detection paper claim that their methods can detect NSB hotspots. We note that, in such claims, the use of the term 'never-seen-before hotspot' has been rather liberal and general, resulting in some level of confusion. In fact, since the literature lacks a technical definition of this term, the reach of these claims remains open to the reader's interpretation. Confusion stems from the fact that two types of test patterns, namely those which are very similar to the training patterns and those which are totally different, can both be interpreted as NSB patterns. Let us consider, for example, the patterns shown in Figure 2a as the training set of a classifier and the patterns shown in Figures 2b-e as the test set. We can observe that test hotspots $Te1$ and $Te2$ are very similar to the training hotspots, whereas test hotspots $Te3$ and $Te4$ look entirely different. While there is a vast difference in the 'similarity' of these patterns to the training set, technically, all of them can be labeled as NSB hotspots, banking on the fact that they are only 'similar' and not 'same' to the patterns in the training set. However, the complexity of

[1] Benchmark 1 was not considered in [12] and [11]. Hence, its results are not available.

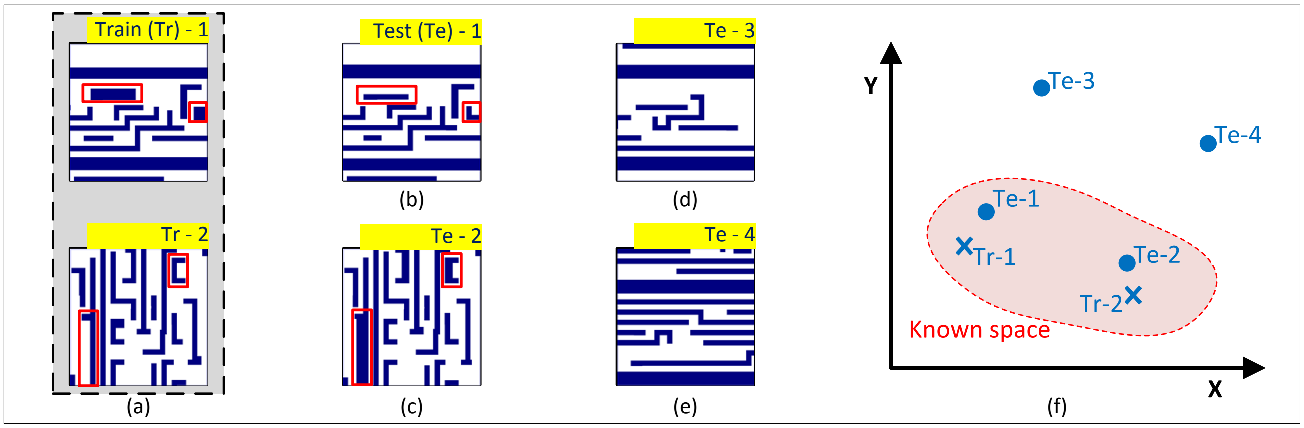

Fig. 2: An example to contrast Truly-Never-Seen-Before (TNSB) hotspots from previously seen hotspots.

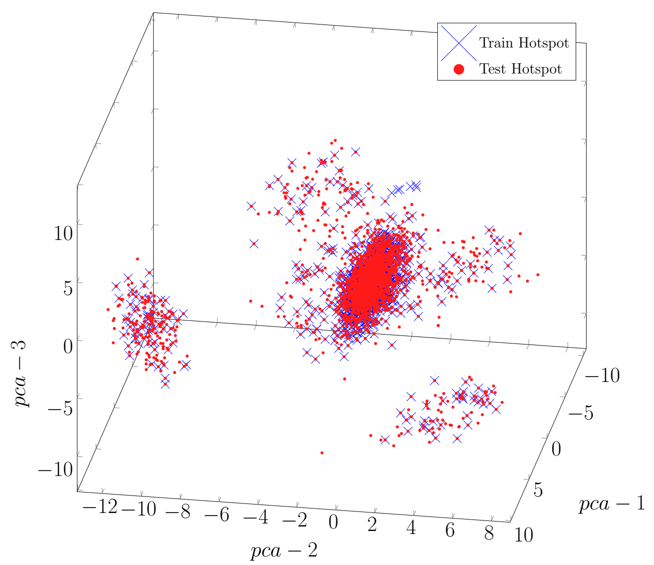

Fig. 3: Distribution of Testing hotspots w.r.t the Training hotspots in the ICCAD-2012 benchmarks.

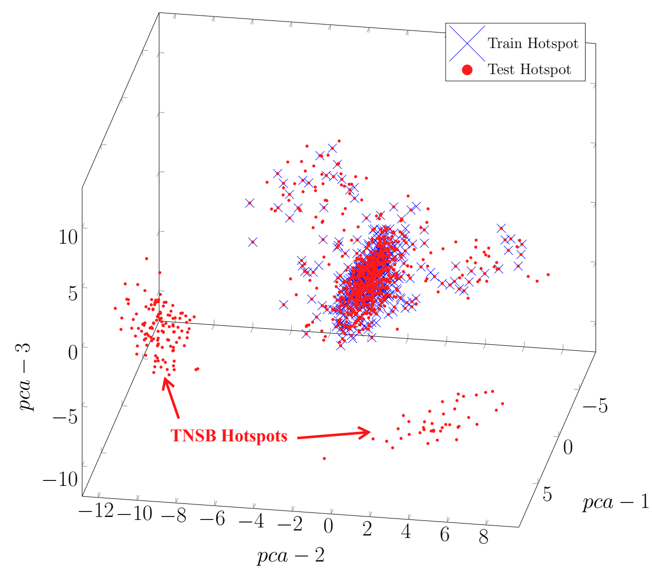

Fig. 4: Distribution of hotspots in the Testing dataset - I, w.r.t the Training hotspots in the ICCAD-2019 benchmarks.

detection varies drastically across these test patterns. Indeed, if we project these training and test patterns onto a hyper-dimensional space, their distribution would appear as shown in Figure 2f. Patterns $Te1$ and $Te2$ would be located very close to the training patterns, well within the known/learnt space, thereby, making them easier to detect. Whereas, for patterns $Te3$ and $Te4$, which are much farther away from the known space, any ML-entity would be making a random guess. Therefore, even though all test hotspots $Te1-4$ can be called NSB, in reality, only patterns $Te3$ and $Te4$ are TNSB.

Unfortunately, the ICCAD-2012 benchmarks do not contain any TNSB patterns. To demonstrate this, we use Principal Component Analysis (PCA) [19]. PCA is an effective tool to visualize the distribution of high-dimensional datasets in lower dimensions. We perform PCA on the training dataset of the ICCAD-2012 benchmarks and we project the test data onto the same space. The first three principal components are plotted as shown in Figure 3. For brevity, only the hotspots from both datasets are shown. Clearly, all of the testing set hotspots lie in very close proximity to the training set hotspots, thereby making them easy-to-detect, just like the patterns $Te1$ and $Te2$ from our previous example. However, TNSB patterns such as $Te3$ and $Te4$ are not found in this space. Oblivious to the pitfall introduced by the lack of such patterns in the ICCAD-2012 benchmarks, most previous methods which have shown high hotspot prediction rates have created a false perception that they can generally detect NSB hotspots, including TNSB hotspots.

The proposed ICCAD-2019 benchmarks, on the other hand, do contain TNSB hotspots. To visualize them, we once again perform PCA on the Training dataset and project the Testing dataset - I onto the same space. As shown in Figure 4, this

TABLE IV: Test results from the proposed ICCAD-2019 benchmarks (Testing dataset - I).

Benchmark	Hotspots #	Non-Hotspots #	DAC'17 [12]		TCAD'18 [11]	
			Accuracy	False Alarm	Accuracy	False Alarm
Complete test set including TNSB hotspots	1001	14621	67.83%	2.00%	77.02%	4.10%
TNSB hotspots only	164	0	14.02%	Not applicable	1.22%	Not applicable

test set contains two groups of hotspots which are significantly far away from all training set hotspots. These hotspots share the characteristics of patterns $Te3$ and $Te4$ from our earlier example, and therefore, can be regarded as TNSB hotspots.

To determine whether the SOTA methods can indeed detect TNSB hotspots, we evaluated their performance on the proposed ICCAD-2019 benchmarks. We used the source code from [20], trained the models using the proposed Training dataset and tested them using the Testing Dataset - I. The results are shown in Table IV[2]. We observe that both of the SOTA deep learning methods, which have demonstrated accuracy rates higher than 99% on the ICCAD-2012 benchmarks, now show a significant reduction in accuracy, mainly due to the presence of TNSB hotspots in the test set. Furthermore, to accurately verify their claims of being able to detect never-seen-before hotspots, we tested their performance on just the TNSB patterns. As shown in the table, their accuracy rates on TNSB patterns drop below 15%. In fact, we expect the accuracy rates of [12] to be even lower because [11], which is an extension of [12] and regarded as a superior method, shows much lower prediction rates. In turn, this implies that even the 14% accuracy rate could be due to spurious/random predictions.

This analysis corroborates our objection to the claim that ML-based methods are effective in detecting TNSB hotspots. This fallacy has largely gone unnoticed because the widely used ICCAD-2012 benchmarks do not contain any TNSB hotspots. By introducing the Testing dataset - I of the proposed ICCAD-2019 benchmarks, we challenge the community to reevaluate and improve their methods in order to achieve high accuracy rates on TNSB hotspots.

We clarify that, through this analysis, we do not imply that TNSB hotspot detection is impossible. We are, however, of the opinion that contemporary ML-based methods alone do not possess such abilities because their performance is capped by the quality of training datasets [22]. We believe that augmenting these training datasets through Early Design Space Exploration (EDSE) tools [23] holds the potential to turn TNSB hotspot detection into a reality.

EDSE involves generating random, but realistic, layout patterns using just the basic design rules from the PDK. Commercial CAD tools performing this task are already available [24]. The underlying rationale is that the random pattern generation process could potentially generate TNSB patterns. Such patterns could be identified, vetted using lithographic simulations and, then, used to enhance the training datasets for ML-based hotspot detection methods. Through this process, previously unknown areas of the hyper-dimensional space could be transformed into known/learnt areas and, therefore, could potentially make better predictions on TNSB patterns. In-depth analysis on EDSE is outside the scope of this paper. We direct the interested reader to [23] for further details.

[2][13] is not included in this analysis because the 'complete and working' source code [21] of this method has not yet been made available to us by the time of submission.

V. Fallacy 3: The SOTA Methods Can Effectively Prevent False Alarms

ML-based hotspot detection methods were proposed as an improvement over PM-based methods both in terms of NSB hotspot identification and in terms of false alarm reduction. While we already objected to the former claim, we do believe that the latter is true and that ML-based methods have the potential to achieve tremendous false alarm reduction. However, in our opinion, most SOTA methods are not effective in keeping false alarms in check. We make such a claim, despite their impeccable results on the ICCAD-2012 benchmarks, because we posit that they have been tested on an Easy-To-Classify (ETC) dataset. We consider the ICCAD-2012 dataset as ETC not only because we have demonstrated that it can be classified effectively using simple ML-based flows, but also because of its sparsely distributed test datasets. By sparse distribution we imply that the hotspots and non-hotspots in the test dataset are located far away from each other in the hyper-dimensional space. As depicted in Figure 5a, this creates an ETC scenario where, despite the ML entity learning large areas surrounding known hotspots as 'hotspot regions', it does not make many false predictions; this is because the test hotspots are always located in close proximity of known hotspots, while the non-hotspots are located farther away.

In practice, however, most layout pattern databases are dense in nature, i.e., the hotspots and non-hotspots in the test dataset lie in close proximity to each other. Such a data distribution arises from the fact that many patterns which look very similar to each other, yet have minor differences between them, can be found in the same technology node/PDK. Authors of [25] have performed an interesting study, wherein they compared the patterns found on test-chips, typically used by foundries, against the patterns found in product designs. They discovered that some of the patterns from product designs were topologically similar to the patterns seen in test-chips, but product designs had many more variations of the same. Applying the findings of this study to hotspot detection, we can consider a situation where a hotspot pattern, as shown in Figure 5c, could be used while training a model. Then, during the life-time of the node, this model may be tested with many patterns which look very similar to the original pattern, but have minor variations in them. Examples of such variants are shown in Figure 5d. As noted in [10], even minor nm-level variation could mean the difference between

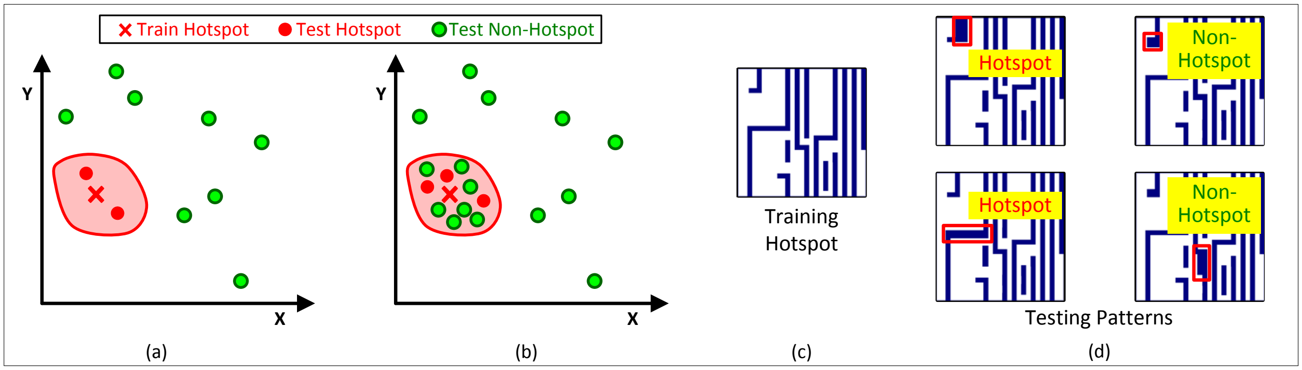

Fig. 5: Comparison of Sparse vs. Dense datasets.

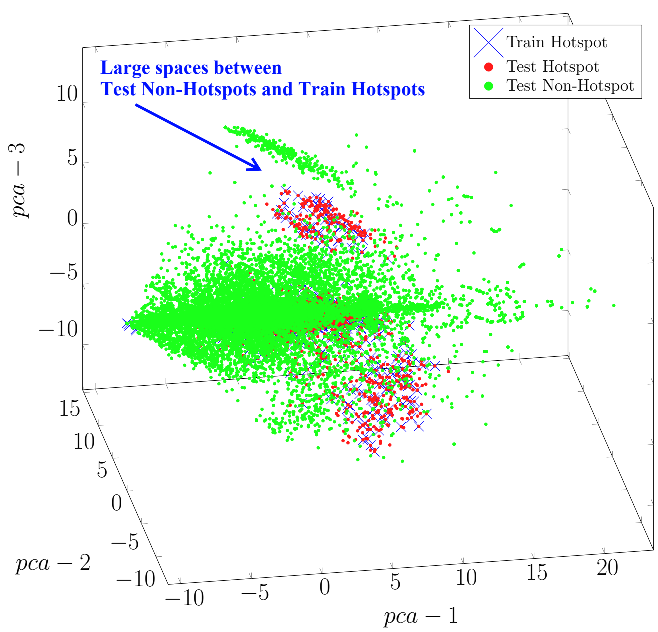

Fig. 6: Distribution of Testing patterns w.r.t the Training hotspots in the ICCAD-2012 benchmarks.

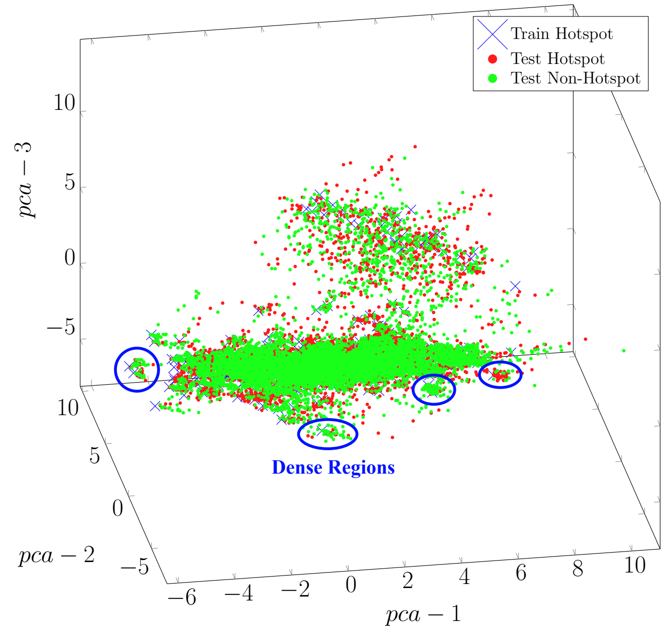

Fig. 7: Distribution of the HTC patterns w.r.t the Training hotspots in the ICCAD-2019 benchmarks.

a pattern becoming a hotspot or a non-hotspot. Therefore, many such variations of the original hotspot pattern could be true non-hotspots. To visualize such a scenario, if we project the patterns from Figures 5c-d onto a 2D space, as shown in Figure 5b, we observe that they form a dense cluster around the training hotspot. Due to their close proximity to the previously seen hotspot, the non-hotspots within this region also get predicted as hotspots, thereby giving rise to false alarms. Such patterns are the truly HTC patterns which can test the robustness of ML-based methods against false-alarms.

However, the ICCAD-2012 benchmarks are devoid of HTC patterns, thereby giving the SOTA ML-based methods a false perception of achieving extremely low false-alarm rates. On the other hand, the Testing Dataset - II of the proposed ICCAD-2019 benchmarks solely consists of HTC patterns. To contrast the data distribution in the two datasets, we perform PCA on their training datasets and project their respective test datasets (Testing dataset - II in the case of the proposed ICCAD-2019 benchmarks) onto the same space. Since the Testing dataset - II from the ICCAD-2019 benchmarks is approximately 8X larger than the ICCAD-2012 test set, to ensure fairness in the comparison, we uniformly down-sample it and plot the same number of data-points as found in the ICCAD-2012 test set. For the sake of brevity, and to remain aligned with the illustrations in Figures 5a and b, only the training hotspots, testing hotspots and testing non-hotspots are plotted. The PCA plot of the ICCAD-2012 benchmarks is shown in Figure 6, whereas the plot corresponding to the proposed ICCAD-2019 benchmarks is shown in Figure 7. By comparing the two plots, we can clearly observe that the training hotspots in the ICCAD-2019 benchmarks are surrounded by dense clusters of both test hotspots and test

TABLE V: Test results from the proposed ICCAD-2019 benchmarks (Testing dataset - II).

Benchmark			**DAC'17 [12]**		**TCAD'18 [11]**		**VTS'18 [26] (With DB Enhancement)**	
	Hotspots #	**Non-Hotspots #**	**Accuracy**	**False Alarm**	**Accuracy**	**False Alarm**	**Accuracy**	**False Alarm**
Part-1	3860	4449	98.83%	96.18%	97.90%	94.56%	87.62%	18.63%
Part-2	8469	7825	99.44%	98.19%	99.57%	97.52%	90.03%	22.90%
Part-3	7552	9212	99.45%	98.52%	99.85%	98.83%	88.14%	18.17%
Part-4	8429	7102	99.71%	99.03%	99.69%	98.21%	90.66%	21.32%
Part-5	8361	7944	99.59%	97.70%	99.96%	97.48%	89.97%	19.45%
Part-6	9561	10363	99.64%	98.64%	99.69%	97.95%	87.15%	22.18%
Part-7	9027	9204	99.68%	98.60%	99.72%	98.74%	89.39%	19.32%
Part-8	7232	8034	98.92%	95.79%	98.99%	94.97%	88.33%	21.46%
Part-9	1819	1390	99.72%	97.77%	99.50%	99.06%	89.33%	26.83%
Average Values			99.44%	97.82%	99.43%	97.48%	88.96%	21.14%

non-hotspots, precisely the scenario illustrated in Figure 5b.

To determine whether the SOTA methods can indeed prevent false alarms, we evaluated them using the proposed ICCAD-2019 benchmarks. We used the source code from [20], trained the models using the proposed Training dataset and tested them using the Testing Dataset - II [3]. The results are shown in Table V. We observe that the same methods which demonstrated false-alarm rates below 0.5% on the ICCAD-2012 benchmarks now show average false alarm rates of about 97%. Therefore, these results confirm our claim that the SOTA methods are not effective in preventing false alarms.

Despite observing such high false alarm rates from recent, deep learning-based hotspot detection methods, we believe that ML-based methods hold the potential to prevent false alarms. Since the performance of ML algorithms often depends on the quality of the training dataset [22], greater emphasis must be placed on improving the information theoretic content of the training datasets. If many variants of known hotspots are used during training, ML algorithms can indeed learn the fine differences between such patterns and make much more robust predictions. However, obtaining many variations of a known hotspot is challenging as they may not be found in a single design, in a small set of designs, or even in many designs obtained from the same standard-cell libraries. They may be found by constantly mining product designs throughout the lifetime of the technology node, but that defeats the purpose of generating hotspot detection models. To resolve this stalemate, authors of [26] proposed the use of synthetic pattern generation to actively generate many variants of known hotspots, vet them through lithographic simulations and use them to enhance the training datasets. They have demonstrated that synthetic database enhancement can indeed achieve significant reduction in false alarms.

To verify whether database enhancement could indeed reduce false alarms, we used the source code and implemented the hotspot detection flow proposed in [26]. We used the training dataset from the proposed ICCAD-2019 benchmarks as a baseline, enhanced it using synthetic patterns, trained a hotspot detection model and tested it using the Testing dataset - II. The results from this experiment are also shown in Table V. We observe that database enhancement provides about 5X reduction in false alarms in comparison to the SOTA deep learning methods. Based on these results, we would like to steer the community's efforts away from relying on increasingly powerful ML algorithms and towards methods for synthetically or otherwise enhancing the underlying training dataset. In our opinion, it is extremely difficult to prevent false alarms without using such enhancement and, through the introduction of the proposed ICCAD-2019 benchmark dataset, we challenge the community to reevaluate and improve their methods in order to achieve low false alarm rates on HTC patterns.

[3]To ensure ease of sharing while abiding by the file size restrictions of data hosting sites, the large Testing Dataset-II has been divided into 9 parts and each part is tested individually. However, merging them and testing all at once provides the exact same results.

VI. Conclusion

Detection of lithographic hotspots during the design stage of ICs is an indispensable necessity towards achieving high yield in contemporary technology nodes. Despite the extensive efforts of the community over the last decade, solutions that achieve high detection rates on TNSB patterns and low false alarm rates on HTC patterns still remain elusive, partly due to the truly challenging nature of the problem and partly due the three fallacies discussed in this paper. We elucidated the pitfalls resulting in these fallacies, which stem mainly from the inherent limitations of the popularly used ICCAD-2012 benchmark dataset, and we proposed an updated ICCAD-2019 version which alleviates these limitations. We showed that ML-based hotspot detection is indeed an effective solution but its performance can be significantly improved by augmenting it with other novel methods, and we demonstrated about 5X reduction in false alarms in comparison to the SOTA. Thereby, we aspire to revitalize the efforts of the hotspot detection community and to steer them towards incorporating solutions such as EDSE and training-set enhancement, which address the issues at the heart of the problem, namely detection of TNSB and HTC patterns, respectively.

Acknowledgment

This research has been partially supported by the Semiconductor Research Corporation (SRC) through task 2810.025.

References

[1] V. Dai, L. Capodieci, J. Yang, and N. Rodriguez, "Developing DRC Plus rules through 2D pattern extraction and clustering techniques," in *SPIE Design for Manufacturability through Design-Process Integration*, 2009, p. 727517.

[2] J. Yang, N. Rodriguez, O. Omedes, F. Gennari, Y.-C. Lai, and V. Mankad, “DRCPlus in a router: automatic elimination of lithography hotspots using 2D pattern detection and correction,” in *SPIE Design for Manufacturability through Design-Process Integration*, 2010, p. 76410Q.

[3] W.-Y. Wen, J.-C. Li, S.-Y. Lin, J.-Y. Chen, and S.-C. Chang, “A fuzzy-matching model with grid reduction for lithography hotspot detection,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 11, pp. 1671–1680, 2014.

[4] D. Ding, X. Wu, J. Ghosh, and D. Z. Pan, “Machine learning based lithographic hotspot detection with critical-feature extraction and classification,” in *IEEE International Conference on IC Design and Technology (ICICDT)*, 2009, pp. 219–222.

[5] D. Ding, J. A. Torres, and D. Z. Pan, “High performance lithography hotspot detection with successively refined pattern identifications and machine learning,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 11, pp. 1621–1634, 2011.

[6] D. Ding, B. Yu, J. Ghosh, and D. Z. Pan, “EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation,” in *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2012, pp. 263–270.

[7] T. Matsunawa, J.-R. Gao, B. Yu, and D. Z. Pan, “A new lithography hotspot detection framework based on AdaBoost classifier and simplified feature extraction,” in *SPIE Design-Process-Technology Co-optimization for Manufacturability*, 2015, p. 94270S.

[8] H. Zhang, B. Yu, and E. F. Young, “Enabling online learning in lithography hotspot detection with information-theoretic feature optimization,” in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, p. 47.

[9] K. Madkour, S. Mohamed, D. Tantawy, and M. Anis, “Hotspot detection using machine learning,” in *International Symposium on Quality Electronic Design (ISQED)*, 2016, pp. 405–409.

[10] H. Yang, L. Luo, J. Su, C. Lin, and B. Yu, “Imbalance aware lithography hotspot detection: a deep learning approach,” in *SPIE Design-Process-Technology Co-optimization for Manufacturability*, 2017, p. 1014807.

[11] H. Yang, J. Su, Y. Zou, Y. Ma, B. Yu, and E. F. Young, “Layout hotspot detection with feature tensor generation and deep biased learning,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 6, pp. 1175–1187, 2018.

[12] H. Yang, J. Su, Y. Zou, B. Yu, and E. F. Young, “Layout hotspot detection with feature tensor generation and deep biased learning,” in *Design Automation Conference (DAC)*, 2017, p. 62.

[13] V. Borisov and J. Scheible, “Lithography hotspots detection using deep learning,” in *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2018, pp. 145–148.

[14] J. W. Park, A. Torres, and X. Song, “Litho-aware machine learning for hotspot detection,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 7, pp. 1510–1514, 2018.

[15] J. A. Torres, “ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite,” in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2012, pp. 349–350.

[16] Source_Code and Proposed Benchmarks, “Machine Learning-Based Hotspot Detection: Fallacies, Pitfalls and Marching Orders,” https://github.com/gauravr1991/ICCAD2019Benchmarks, 2019.

[17] G. Kataoka, M. Inagi, S. Nagayama, and S. Wakabayashi, “Novel feature vectors considering distances between wires for lithography hotspot detection,” in *Euromicro Conference on Digital System Design (DSD)*, 2018, pp. 85–90.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[19] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[20] Source_Code of SOTA methods (DAC 2017, TCAD 2018), “Layout hotspot detection with feature tensor generation and deep biased learning,” https://github.com/phdyang007/dlhsd, 2017, 2018, [Online; accessed 1-March-2019].

[21] Source_Code of SOTA methods (SMACD 2018), “Lithography Hotspots Detection Using Deep Learning,” https://github.com/unnir/lithography_hotspot_detection, 2018, [Online; accessed 1-March-2019].

[22] C. Cortes, L. D. Jackel, and W.-P. Chiang, “Limits on learning machine accuracy imposed by data quality,” in *Advances in Neural Information Processing Systems*, 1995, pp. 239–246.

[23] H. Li, E. Zou, R. Lee, S. Hong, S. Liu, J. Wang, C. Du, R. Zhang, K. Madkour, H. Ali *et al.*, “Design space exploration for early identification of yield limiting patterns,” in *SPIE Design-Process-Technology Co-optimization for Manufacturability*, 2016, p. 97810W.

[24] Mentor Layout Schema Generator.

[25] V. Dai, E. K. C. Teoh, J. Xu, and B. Rangarajan, “Optimization of complex high-dimensional layout configurations for IC physical designs using graph search, data analytics, and machine learning,” in *SPIE Design-Process-Technology Co-optimization for Manufacturability*, 2017, p. 1014808.

[26] G. R. Reddy, C. Xanthopoulos, and Y. Makris, “Enhanced hotspot detection through synthetic pattern generation and design of experiments,” in *IEEE VLSI Test Symposium (VTS)*, 2018, pp. 1–6.