SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I11P119

Volume 12 Issue 11, 228-239, November 2025

Original Article

Machine Learning-Aided Test Pattern Generation for VLSI Circuits: A Decision Tree Regressor Approach

Sima K Gonsai[1], Usha Mehta[2]

[1,2]*Department of Electronics and Communication, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India.*

[2]*Corresponding Author: usha.mehta@nirmauni.ac.in*

Received: 18 September 2025 Revised: 18 October 2025 Accepted: 17 November 2025 Published: 29 November 2025

Abstract *- As the demand for complex IC, high-performance computing increases, IC technology needs to evolve quickly. It relies on the technology node shrinking. Due to the highly complex nature of the chips, designing and testing circuits becomes an extremely challenging and crucial task. To ensure chip reliability, IC testing is performed. It is one of the most important, complex, time-consuming, and necessary tasks. To validate the chip, Testing ensures the integrity of the chip by verifying the functionality, timing, and interconnections. Machine learning empowers the world by enhancing automation, data-driven decision-making, and optimization across all domains, including healthcare, finance, security, chip design, and manufacturing. This work demonstrates the role of Machine Learning (ML) in Automatic Test Pattern Generation (ATPG) for digital VLSI circuits and contributes to an alternative approach for test data generation. Compared to the ATALANTA tool, the Decision Tree Regressor(DTR) model achieved the best score, with an 18.1% reduction in test pattern count while maintaining the same fault coverage. For almost all benchmark circuits, the DTR model provides either superior fault coverage compared to the ATALANTA tool or a reduced test pattern count at equivalent fault coverage.*

Keywords *- Automatic Test Pattern Generation, Decision Tree Regression, Fault Coverage, IC Testing, Machine Learning.*

1. Introduction

The primary goal of testing is to identify defects and eliminate defective and faulty integrated circuits reliably. As technology evolves, narrowing design geometries with technology node, growing circuit complexity, and faster clock speeds have greatly increased circuits' susceptibility to manufacturing defects. These features emphasise the significance of reliable fault modelling, detection, and testing processes. In particular, the continuous evolution of the manufacturing process has resulted in the introduction of unusual failure types, necessitating creative IC testing methodologies.

Despite noteworthy advancements in traditional ATPG algorithms, no direct attempt has been made to use machine learning techniques for test data generation. This gap highlights the need for a data-driven strategy to enhance fault prediction and test pattern generation efficiency.

Machine learning is a broad concept that has recently been applied across various domains, including image processing, speaker recognition, medical diagnostics, automation, and autonomous vehicles. The fundamental concept is to instruct a machine to perform an intellectual task, thereby substituting for human decision-making or an expensive and lengthy analysis procedure. A machine possesses inbuilt learning capabilities and can potentially be trained using a large volume of data as training examples. It is not specifically programmed; instead, it learns from the data during the training phase, as it can predict unseen input that was not present during the training phase.

Recent studies have commenced investigating ML methodologies for test vector generation and fault analysis work; nevertheless, initiatives focused on direct, end-to-end Automatic Test Pattern Generation are still relatively new. Data-driven ATPG approaches have been introduced to minimize backtracking and enhance test pattern search, resulting in significant reductions in runtime and increased test pattern quality on certain benchmark circuits. [1], More extensive reviews of Machine Learning in VLSI/EDA indicate that learning based methods for power estimation, fault prediction, and verification assistance are rapidly developing. They also highlight that ML is capable of capturing intricate, nonlinear relationships that standard heuristic ATPG tools are unable to model explicitly. [2] However, recent surveys of directed test-generation procedures demonstrate that the majority of standard ATPG tools still depend on heuristic and symbolic algorithms (e.g., D-algorithm and its variants, logic-based heuristics). Additionally, machine learning techniques are primarily employed for sub-problems, such as fault prediction, feature selection, and test pattern sorting, rather than a general ATPG workflow for standard benchmarks, like ISCAS. [3] Consequently, there is a prominent research gap in developing straightforward test generation methods that are evaluated on classical ISCAS benchmark circuits and compared with conventional tools. This work fills that gap by proposing and evaluating a decision-tree-regression-based approach for test pattern estimation, achieving comparable fault coverage.

In this work, a dataset is generated using the renowned academic tool ATALANTA. The ATALANTA tool was developed at Virginia Tech University. The necessary libraries, executable files, and documentation are available at GitHub. [4] ATALANTA is a tool for Automatic Test Pattern Generation (ATPG) that generates a list of test vectors and golden output to achieve high fault coverage. It is a free tool. The test vectors generated for the ISCAS85 benchmark circuits using the ATALANTA tool serve as input to the ML model in this work.

To address scalability concerns in EDA, a variety of Machine Learning (ML) and, recently, deep learning based techniques have been put forth, such as those in tests of silicon chips[5-8]. For example, designers can identify design hotspots [9] in a physical layout using a deep neural network. DNNs can also be used to assess routability in physical design. [10] In comparison to heuristic or analytical approaches, the power of the Deep Neural Network lies in its capacity to approximate functions [11-12], providing a quicker test turnaround. [13] Test engineers must translate digital circuit test challenges into "learning-based" problems in order to utilize the capacity of machine learning. Selecting relevant features from a design to feed into an ML model requires a thorough understanding of the chip test domain. As machine learning is a fully data-dependent and data-driven domain, sufficient care must be taken to avoid the fragility of the model due to an insufficient or imbalanced dataset. Specifically, in EDA life cycle, such situation may arise when poor predictive model is deployed. [14] This affects the production life cycle and timeframe of fabrication of IC is delayed. Therefore, it is essential that to accurately train and evaluate the model.

This paper examines the role of machine learning in digital circuit testing, highlighting the challenges and efficacy of its application. To highlight the effectiveness of the ML model in IC testing, the model required training and evaluation. Selecting a suitable input abstraction to allow a machine learning model to learn robust features is one challenge.

Testing is one of the most important and time-consuming aspects of digital VLSI chip design [14], as it determines the chip's overall fault coverage and yield. A proper test set is required to achieve high fault coverage; at the same time, a smaller number of test patterns is desirable to minimize test time.

The decision tree regression model performs well for chip testing, enabling the generation of test patterns for ISCAS85 Benchmark circuits. This work is carried out on the ISCAS85 benchmark circuits and a dataset of around 1000 digital combinational circuits. The circuits in the dataset have varying degrees of complexity in terms of the number of input and output ports, total number of gates, circuit level, and number of faults.

Choosing proper attributes to train a Decision tree regression model for chip testing is another issue that was handled. However, this study sheds light on whether machine learning ML), specifically Decision Tree regression, can aid ATPG for test set generation in digital combinational circuits. This work presents a new insight into the practical application of DTR in IC testing. The limitations of applying machine learning ML) in the IC testing domain are also revealed in this paper.

Specifically, three significant ML aspects in the context of VLSI chip testing are exhaustively examined in this work:

1.1. Data Set Preparations

A machine learning based models typically need a large data set to train the model accurately. The dataset comprises 1,000 digital combinational circuits and their corresponding test patterns, all of which are based on the ISCAS85 benchmark circuits. The ATALANTA tool was used to prepare the dataset for this research.

1.2. ML for Test Pattern Generation

A Decision tree regression model is applied to digital combinational circuits for the prediction of the corresponding test pattern set for various circuits.

1.3. Validate the Predictions

To validate the results (test pattern set) predicted by the DTR model, test patterns are applied to the FSIM fault simulator, and the fault coverage of the test pattern set is computed.

1.3.1. Paper Organization

In this paper, open challenges have been identified and potential research directions at the intersection of ML and VLSI testing. Some useful background information is provided in Section II. This section can be skipped if machine learning concepts and standard ML algorithms are known. In Section III, relevant machine learning ML) research has been investigated for circuit testing. In Section IV, the implementation of this machine learning ML) application is discussed. Results are shown and explored in Section V. Section VI concludes the work.

2. Brief on Machine Learning Techniques

Here is a summary of some useful background information. Readers who are already familiar with these ideas can go to the next Section.

2.1. Recent Machine Learning Methods

The machine learning field encompasses mathematical models that are trained on a dataset to make predictions on unseen data. Prediction of test patterns for different digital circuits is modeled as a regression machine learning technique. In this model, numerical or binary numbers are predicted rather than classified, where samples are classified into an appropriate class. Standard machine learning techniques, such as Decision Tree Regression (DTR) [15], Support Vector Regression (SVR) [16], Multiple Linear Regression [17], Neural Network [18], and Deep Neural Network [19], are well-suited for VLSI design and test pattern generation. Moreover, it is crucial to integrate feature extraction and domain knowledge to map the problem into a machine learning model effectively. In recent times, experts have investigated methods to organize "Artificial Neurons" into intricate, "Deep" structures; these Deep Neural Networks exhibit cutting-edge classification capabilities in every field, including Natural Language Processing (NLP), face recognition as identity, object detection, and video-image classification.

2.2. Decision Tree Regression

The advantages of DTR include their capacity to process datasets with mistakes and handle missing values, and their ability to deal with various kinds of input data, including text, numeric, and categorical data.

A decision tree can be used for solving classification as well as regression problems. Like a typical "tree", a decision tree has a root node, branches, and a leaf node. In solving many real-life problems, which are not necessarily linearly dependent on input variables, they are non-linear. In such cases, non-linear, hierarchical, rule-based methods are used increasingly. One of such methods is the regression decision tree technique. Upon observing the problem and its dataset, DTR will create a tree structure by establishing decision rules based on the queries. Firstly, some questions are asked to determine the root node, which is the primary node of the entire tree structure. As shown in Figure 1, the root node is the source of other nodes in a tree. A decision node is a child node of the root or a decision node. This node decides by testing an attribute, as in Figure 1, whether $X > \rho 1$or $\rho 1 > X > \rho 2$or $\rho 2 > X$. The decision node is also called an internal node. Then, by asking a few more questions, the entire tree structure will be completed, reaching the leaf nodes.

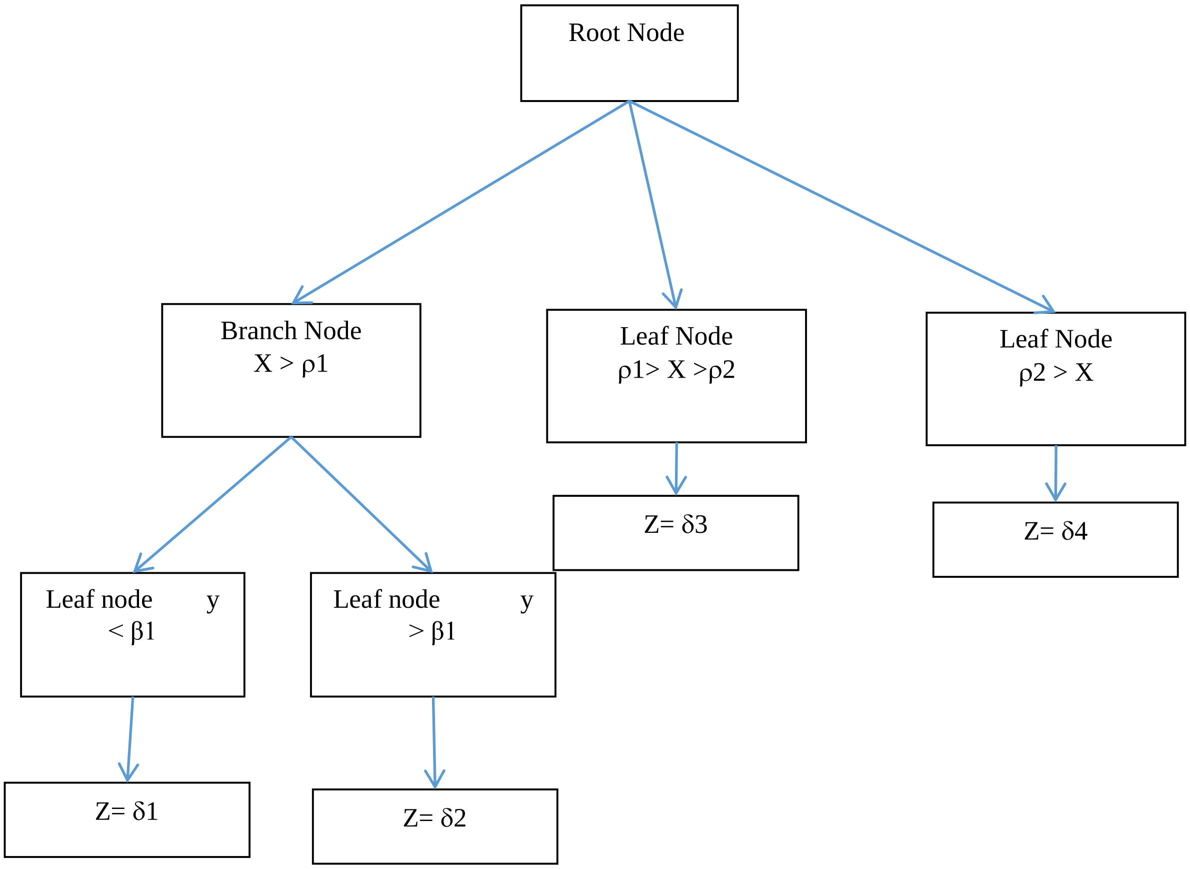


Fig. 1 Example of decision tree regression

A leaf node is also referred to as the terminal node, which is the node without any children. At this node, a certain decision criterion, as in Figure 1, Z=δ1 or Z=δ2 or Z= δ3 or Z= δ4 is used to find the target value. That is the last node of the tree structure.

By recursively dividing the entire feature space into multiple spaces and then predicting continuous values at every leaf node based on weighted average target values. However, Data set is represented as D = (xi, yi), i=1…….N. Where, Xi is the input feature. Yi is the target variable, and N is the total samples in the dataset.

The feature space is divided into M disjoint regions via a decision tree R1, R2, …, RM.

The predicted result, y, for an input x that belongs to region RM, is calculated as the average of the target values in the same region.

$$\mathrm{y(x)} = \frac{1}{|R_m|}\sum_{\mathrm{x_i \in R_m}}^{n} \mathrm{y_i} \tag{1}$$

Here $|R_m|$ is the total samples in region $R_{m.}$

Performance Measurement Metrics for Regression Model.

To verify how well the regression model fits the dataset during training and also generalized well for unseen data, it is crucial to assess the regression model through some performance parameters. The widely used parameter for evaluation is explained in this section.

Mean squared error is determined by taking the difference between actual output yi and predicted output ŷi, for each sample, and then taking the square of the difference. Calculating the average over the entire n data samples will produce MSE, MSE-Mean mean-squared error. A low mean squared error is desirable to have a well-trained model. Also, observe that it is not very close to zero, which may lead to over-fitting of the model [20].

$$Mean\ Square\ Error = \frac{1}{\mathrm{n}}\sum_{K=0}^{n}\left(y^{k}{}_{act} - y^{k}{}_{pre}\right)^{2} \tag{2}$$

2.2.1. K-FOLD CROSS-VALIDATION

To generalize the machine learning model well for predicting unseen data, it is sometimes desirable to perform K-fold cross-validation. The entire dataset is divided into k parts. Among these k-parts, (k-1) parts are used for training the ML model, and the kth part is used to test the model. This process is performed k times, each time one part from the sequence number 1 to k is used as a test set, and the remaining total (k-1) parts are used to train the model. An important feature of K-fold cross-validation is that each part will be the test set once. Due to k-times training, the predicted result error will be averaged.

The error of cross-validation is calculated as,

$$ECV = \frac{1}{\mathrm{k}}\sum_{i=1}^{k} E_i \tag{3}$$

However, K-fold cross-validation is discussed as a study of a basic machine learning technique, but it was not implemented here in the proposed work. Instead, the train and test data were split using a train-test split of an 80%-20% ratio, which is considered a sufficient method for the present workflow. Moreover, in future work, it may be incorporated to improve the performance of the model.

Machine Learning (ML) techniques have demonstrated tremendous potential for improving fault detection in VLSI chip testing. Various ML techniques can efficiently evaluate vast amounts of test data and predict patterns that detect faults in chips. This work investigates how ML aids ATPG in VLSI testing.

First, the above methods evaluate the ML models using standard machine learning measures like accuracy, F1 score, or mean square error. However, all these measures are inadequate and fail to capture the practical significance of IC testing.

Test-specific metrics might have a poor correlation with standard ML measures. So, the ML model test pattern predictions result is taken and applied to the standard tool FSIM, a fault simulator, to measure fault coverage. Fault coverage is the most significant parameter in IC testing. It is defined as the ratio of the number of detected faults to the total number of collapsed faults present in the circuit. Collapsed faults are the reduced fault set or fault dictionary for any digital circuit, achieved by removing equivalent faults.

$$Fault\ Coverage\ (\%) = \frac{Number\ of\ detected\ faults}{Total\ number\ of\ collapsed\ faults}\ X\ 100 \tag{4}$$

2.3. Data Set Preparation

For the successful implementation of an ML model for testing, a large volume of quality, industry-based, balanced data set is necessary. Most industrial test data are not accessible as open source for everyone. In order to train the model correctly, ISCAS-85 benchmark circuits [20] are used. There are only 10 circuits in this benchmark combinational circuit set.

ATALANTA is a tool for Automatic Test Pattern Generation (ATPG) that generates a list of test vectors and golden output to achieve high fault coverage. ATALANTA tool is based on the FAN algorithm and is designed only for combinational circuits. The stuck-at-0 and stuck-at-1 faults are detected using test vectors generated. The output file with the name "\example_circuit_name>.test" is generated, which includes fault-free responses and test vectors. ATALANTA produces test vectors for all stuck-at faults in the example combinational circuit. These generated test vectors of ISCAS85 benchmark circuits using the ATALANTA tool are used as an input to the ML model in this work.

Initially, a total of 200 digital combinational circuits are constructed in Verilog. They are converted into .bench files and applied to the ATALANTA simulator tool [21]. Using the ATALANTA tool, the test pattern set and fault coverage are obtained and used as a dataset to train the model. Initial experiment results indicate that to train the Decision Tree Regression (DTR) model effectively, the data set needs to be further enriched.

So, around 1000 digital combinational circuits with a similar order of interconnections, number of gates, level of circuits, and complexity are developed and applied to the ATALANTA tool. After preprocessing, the enhanced data set is applied to the machine learning model to train the model effectively. As per Table 1, the parameters of DTR are set. The maximum tree depth in most research is set to 10, the random state is set to 42 by default, and the splitting strategy is selected as the best. To ensure accurate DTR training, circuits are divided into groups based on the number of inputs and outputs in each circuit. The number of circuits in each group of the ISCAS-85 circuit set is shown in Figure 2; a Total of 175 circuits are constructed in groups of c7552 and c6288. Similarly, other groups have 150, 105, 104, 70, 61, and 6 circuits as shown in Figure 2. Detailed steps for implementing the proposed work are discussed in the next section.

Table 1. Parameters of DTR model

Parameter	Description	Value
criterion	Function to measure the quality of a split	Squared error
splitter	Strategy used to choose the split at each node	best
Max depth	Maximum depth of the tree	10
Min samples split	Minimum number of samples to split an internal node.	2
Min samples leaf	Minimum number of samples required at a leaf node.	1
Random state	Random seed for reproducibility.	42
Max features	Number of features to consider when looking for the best split.	Auto

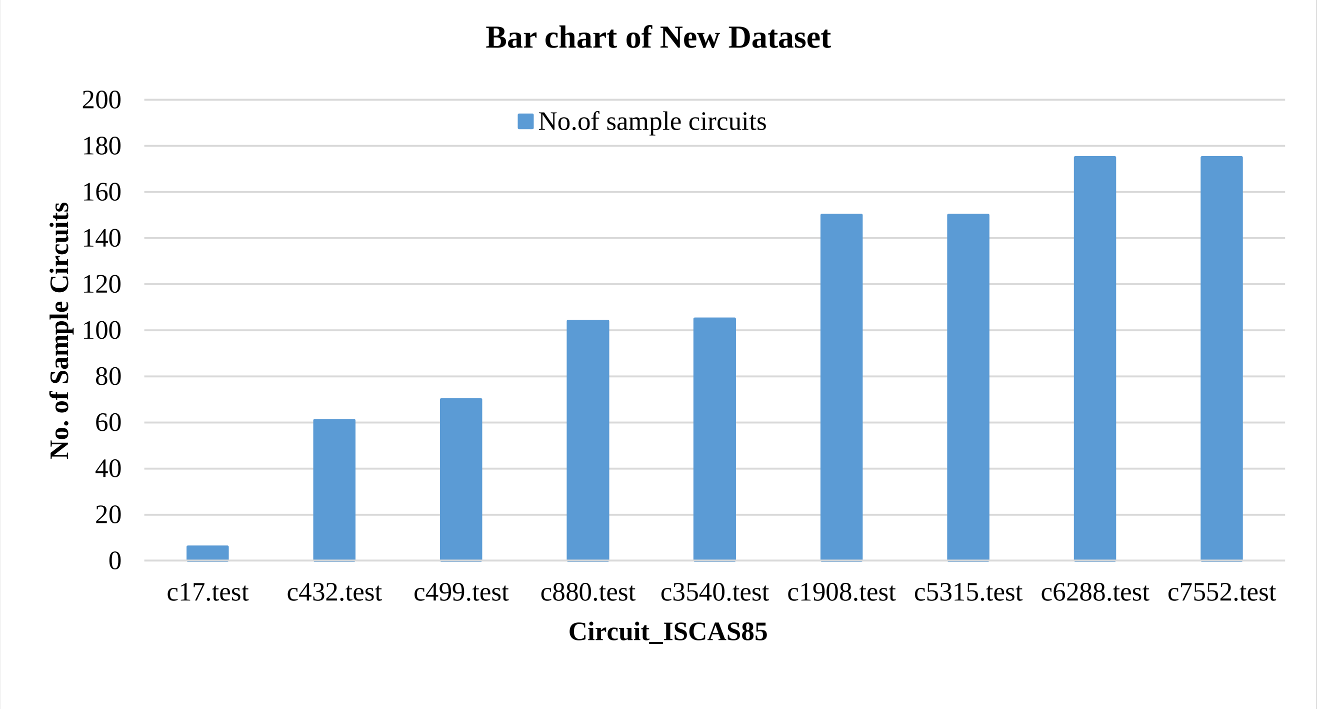


Fig. 2 Circuit groups for the dataset

3. Methodology

Recently, researchers have also found Naïve Bayes and Support Vector Machine useful for fault diagnosis and functional verification. Hence, Machine learning enables engineers to solve issues at different abstraction level in the Chip design life cycle [22-25]. An unsupervised machine learning technique, such as Bayes' theorem, is used for scan chain assessment [26].

The Decision Tree Regressor, a machine learning model, is selected as it provides clear interpretability. Additionally, the hierarchical decision rules of DTR facilitate the mapping of the relationship between circuit input parameters and fault coverage. In Multiple Linear Regression, the ML model can only visualize linear dependencies between input features and output variables, while DTR can map nonlinear dependencies.

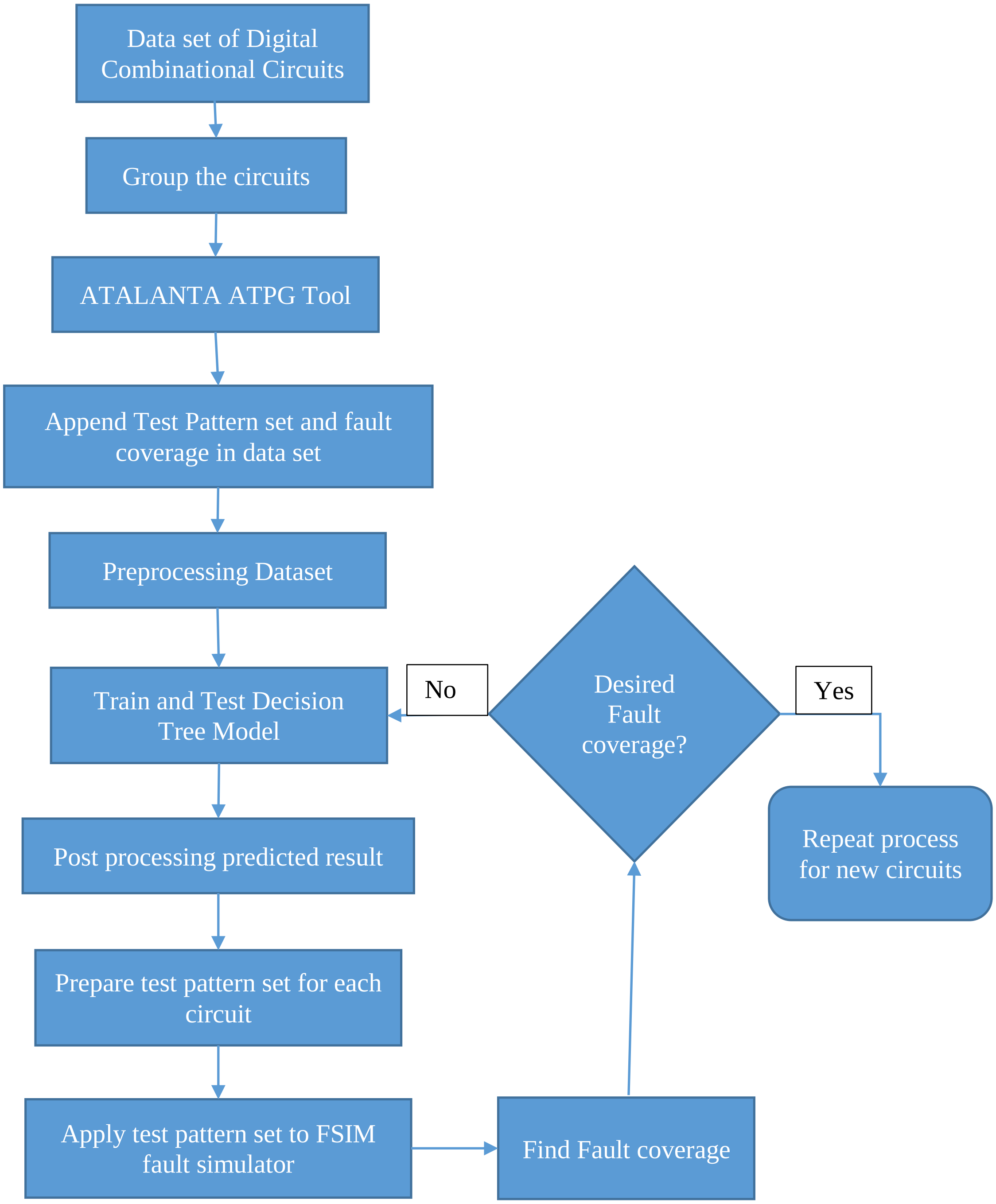

Data set of Digital Combinational Circuits
Group the circuits
ATALANTA ATPG Tool
Append Test Pattern set and fault coverage in data set
Preprocessing Dataset
Train and Test Decision Tree Model
Post processing predicted result
Prepare test pattern set for each circuit
Apply test pattern set to FSIM fault simulator
Find Fault coverage
Desired Fault coverage?
No
Yes
Repeat process for new circuits

Fig. 3 Overall flow of the proposed work

Table 1. Input features for decision tree regressor

Circuit	# Input	# Output	# Gates	# Faults	# OR	# AND	# NOT	# NAND	# NOR	# XOR
C432_3	36	7	160	524	0	7	40	116	19	18
C499_6	41	32	202	758	2	50	40	46	0	104
C880_1	60	26	383	942	29	145	63	148	61	0
C1908_14	33	25	880	1879	0	253	277	626	1	0
C3540_6	50	22	1669	3428	92	733	490	776	68	0
C5315_1	178	123	2307	5350	214	1033	581	1033	27	0
C6288_10	32	32	2416	7744	20	256	32	32	2108	0
C7552_11	207	108	3512	7550	244	1332	876	1882	54	0

Without requiring feature transformations. The Support Vector Regressor requires high tuning of its parameters. Compared to this model, DTR requires less tuning and handles mixed and non-normalized data efficiently. Moreover, for the digital circuit dataset, which includes both discrete and continuous attributes, the Decision Tree Regressor effectively captures feature interactions and nonlinear dependencies, leading to more accurate results.

Figure 3 shows the overall flow of our work. For any machine learning model, the data set is a very important and crucial part for the successful predictions of the result. To obtain an enriched dataset for this algorithm, around 1000 digital combinational circuits bench files are prepared and then applied to the ATALANTA fault simulator for gathering test pattern sets and fault coverage of each circuit.

Test patterns are available in a text file, which is converted into CSV format to facilitate the smooth application of the dataset to a machine learning model. Input features for the ML model are represented in Table 2. Circuit information like # input, # output, # gates, # AND, # OR, # XOR, # faults represents the number of inputs, the number of outputs, the total gates, the number of AND gates, OR gates, and the number of XOR gates. The total number of faults, respectively, is included as an input feature for each circuit, as shown in Table 2.

To properly process the dataset, it requires converting it into smaller fragments and then converting these fragments into numeric values. Each test pattern is fragmented into 3 to 4 parts and converted to decimal form. This preprocessed data set is now ready to train the model. The dataset is divided into per 20% and 80% ratio for testing and training the machine learning model. The decision tree regression model is trained and then tested for various digital combinational circuits.

The decision tree model will predict the set of test patterns for each circuit given for the validation set. The direct results from the ML model will be presented in test patterns for each circuit, in both fragmented and numeric forms. To retrieve the test patterns in binary form, post-processing of the predicted data is required. Predicted and processed results will contain test patterns in binary format, which can be exported as text files for each combinational digital circuit separately. Test pattern list file to be saved as '.tst' format, to apply it to FSIM, the fault simulator. Subsequently, identify and label the text file with the proper "circuitname.tst". Apply it to the FSIM fault simulator to obtain the actual fault coverage of the predicted list of test patterns of a particular circuit under test.

To compute the actual fault coverage of predictions by the Decision tree regression, the FSIM fault simulator is used. If the fault coverage value is lower than expectations, an additional iteration of training and testing the model is done. For this, the DTR model needs to be updated, or feature engineering should be performed. Initially, this experiment was carried out for 200 combinational circuits. Fault coverage was much lower than the expectations of around 90% target value. Therefore, the entire dataset was enhanced by increasing the sample size to 1000 digital circuits, inspired by the ISCAS-85 digital combinational circuits. All processing steps were performed again, including feature extraction, preparing a single .csv file to train and test the DTR model for all 1000 circuits, and then post-processing of the predicted results. In this iteration, the DTR model has performed well. Results of fault coverage are displayed and discussed in the next section.

4. Experimental Results and Discussion

This section represents the experimental result of the decision tree regression test pattern predictions. The machine learning model was implemented in Google Colaboratory (an online platform for executing Python code) on an Intel i5 processor. The model was applied to benchmark circuits, as well as to other circuits that exhibit characteristics similar to those of benchmark circuits. The ATALANTA tool is used for generating test patterns to compare actual results with predicted results. Predictions by the decision tree regression model of ISCAS85 and other complex digital circuits are post-processed. The data was converted back into binary form to calculate fault coverage and compared with the ATALANTA tool's fault coverage.

The FSIM fault simulator was applied to test the pattern set predicted by the decision tree model. FSIM fault simulator was used to check the fault coverage of the test patterns generated by the DTR model and the fault coverage achieved by ATALANTA Tool's test pattern set. Table 3 presents the experiment results, which show that for circuit c499_6, the predicted fault coverage difference is 0.132, indicating that DTR's fault coverage is higher than ATALANTA's by a small margin.

Similarly, highlighted digits in Table 3 demonstrate that c6288_10 and c7552_11 have almost the same fault coverage with 18.1% less test pattern count. Figure 4 displays the comparison graph of group 1 and group 2's circuit fault coverage. It is observed that the c499_6 circuit has 5.8% (test pattern count is increased by 3), and the fault coverage is increased by a small factor. Similarly, Figure 5 shows the comparison of the number of test patterns counted by the ATALANTA tool and the DTR model.

The table represents the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Normalized Root Mean Squared Error (NRMSE). C7552_11 has the smallest NRMSE, and C1908_14 has the highest NRMSE value from the entire set. In VLSI testing, fault coverage of the circuit under test with the total number of test vectors is a key performance parameter, and MSE, RMSE, and NRMSE are of secondary importance.

It can be observed that both ATALANTA's total test pattern count and DTR are nearly the same. The test pattern count difference range in Figure 5 is +6 to -8. So, overall, it can be observed that with nearly the same order of test pattern count, it gives the same order or comparable fault coverage for the circuit under testing of the DTR model.

In summary, Decision Tree Regression outperformed ATALANTA due to its ability to: a) Learns complex relationships between features of circuits and test pattern sets from data. Also, this model is adaptive for various circuit architectures. It reduces the algorithmic bias associated with the heuristic ATPG. It helps in improving the modeling of nonlinear interactions.

Thus, enhanced efficiency of the Decision Tree Regression (DTR) model over the ATALANTA tool for various ISCAS benchmark circuits is due to the data-driven, adaptive characteristics of DTR and the constraints of heuristic-based fault simulation in the ATALANTA tool.

5. Challenges While using Machine Learning in Testing

There are various steps in IC testing, like scan chain diagnosis, fault isolation, test point insertion [27], and test data generation, where a machine learning approach could help or can be an aid to standard methods. But still it is under study.

Or the research phase. For investigating the scope of ML in testing, a few researchers are applying ML methods at various stages of IC testing. The easy accessibility of adequate data that is enriching in quality and volume is crucial for the effectiveness of ML-based methods. Unfortunately, Standard databases for IC testing using machine learning methods have not yet been developed. Thus, the unavailability of a standardized data set is the major hurdle towards switching to ML-based/driven CAD tools. A few bottleneck issues are listed here:

- Majorly, Industrial data regarding testing, failure, or diagnosis is not accessible publicly. Such data would be the proper source to train the ML model.
- For any ML model, a large volume of data is required to train it. Generating synthetic data through simulation is a very lengthy process.
- A large benchmark circuit set (in the order of 1000s of circuits), in terms of variety in size, number of gates, types of faults, level of circuits, and variety of structures in interconnections, is needed to train the model.
- Feature extractions from the large volume and types of digital circuits are required.

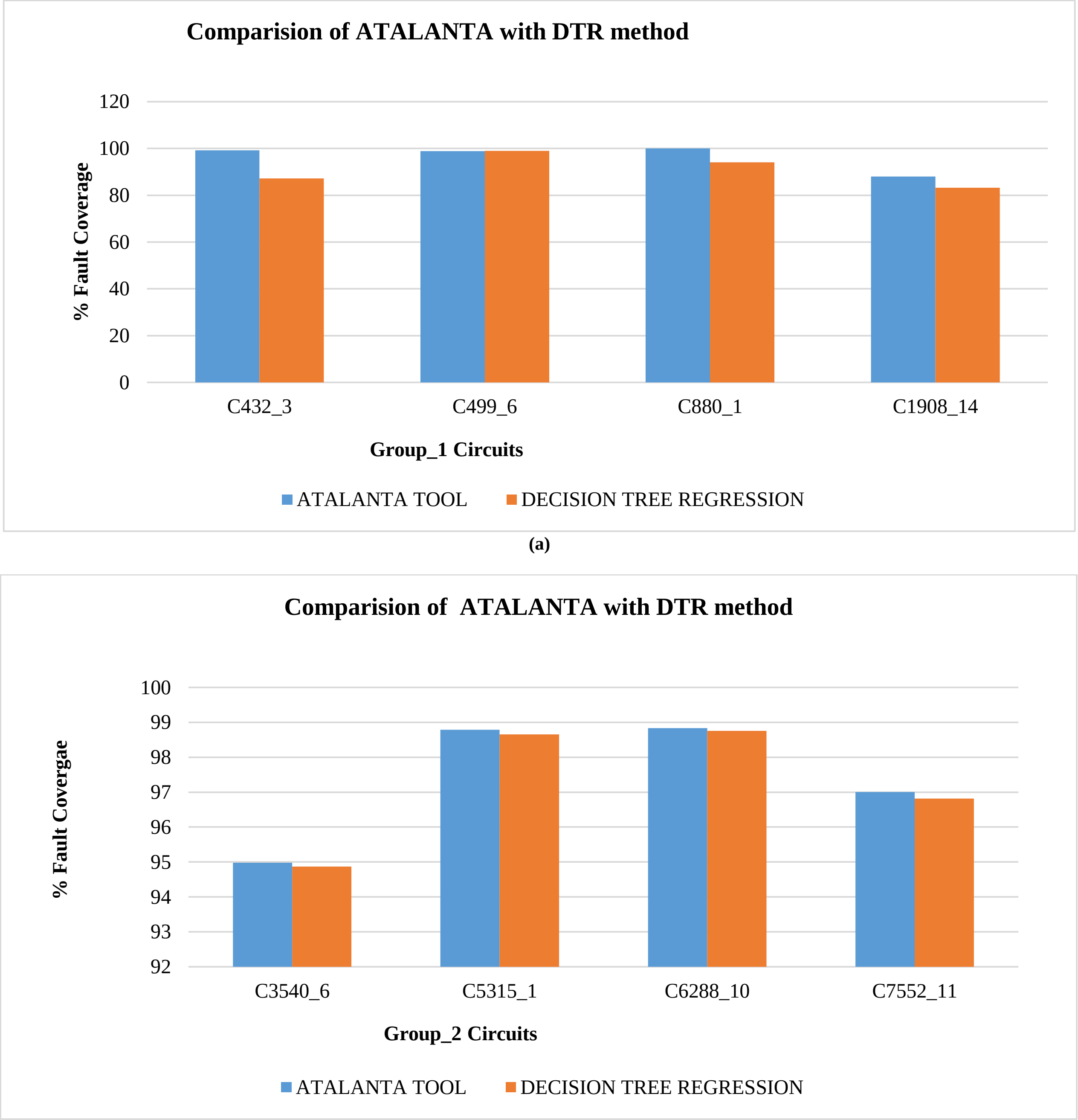
Thus, a considerable amount of time and effort is to be invested in generating synthetic data, preprocessing it, and doing the feature engineering on it.

Table 3. Experiment result of decision tree regressor

Circuit	# Gates	# Faults	# Test Patterns	% Fault Coverage	# Test Patterns	% Fault Coverage	Δ%Fault Coverage	Δ %Test Patterns
			ATALANTA [29]		DTR [FSIM]		Comparison	
C499_6	202	758	51	98.813	54	**98.945**	**+0.132**	+5.8
C880_1	383	942	57	100	**49**	94.055	-5.945	-14
C1908_14	880	1879	102	87.972	**102**	83.236	-4.736	0
C3540_6	1669	3428	161	94.982	**161**	94.866	-0.116	0
C5315_1	2307	5350	115	98.785	118	98.654	-0.131	+2.6
C6288_10	2416	7744	33	98.838	**27**	98.750	**-0.088**	**-18.1**
C7552_11	3512	7550	220	97.007	**217**	96.821	**-0.186**	**-1.4**

Table 4. Experiment result with Mse, Rmse, and Nrmse

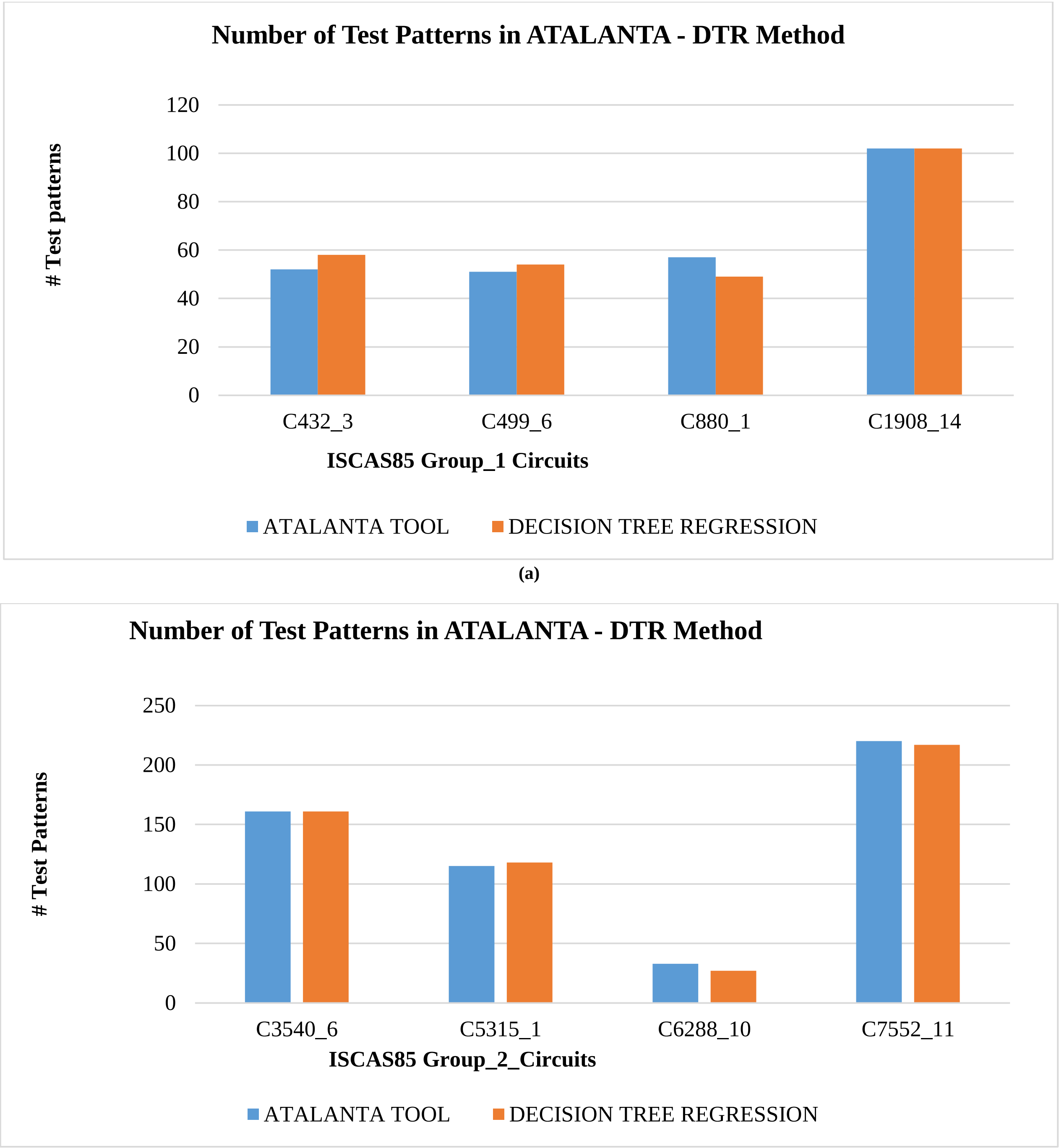
Circuit	# Gates	# Faults	# Test Patterns	% Fault Coverage	MSE	RMSE	NRMSE
C499_6	202	758	54	**98.945**	11947454086	109304	0.104
C880_1	383	942	**49**	94.055	3133524615	55977	0.054
C1908_14	880	1879	**102**	83.236	124096501655	352273	0.336
C3540_6	1669	3428	**161**	94.866	18418256875	135713	0.129
C5315_1	2307	5350	118	98.654	2111007	1452.93	0.090
C6288_10	2416	7744	**27**	98.750	11734	108.32	0.055
C7552_11	3512	7550	**217**	96.821	3042063704	55154.90	0.053



(a)

(b)

Fig. 4 (a) Comparison of group 1 circuits' % FC, and (b) Comparison of group 2 circuits' % FC.



(a)

(b)

Fig. 5 (a)Comparison of group 1 circuits' Test Pattern set, and (b)Comparison of group 2 circuits' Test Pattern set.

6. Conclusion

The size of the test set is a very crucial factor in testing the **c**hip after production.[28-29]. Pattern counts and test expenses rise tremendously with the circuits' growing size and complexity [30-31]. Hence, applying machine learning to solve the test pattern generation and to achieve comparable fault coverage is a recent development in the field of chip testing.

In this proposed work, a new machine learning-based Test set prediction is implemented, which is able to target high fault coverage. Given a digital combinational circuit, a Machine learning model will predict the test set for it. The Decision Tree Regressor model is powerful enough to generate a test set for complex circuits up to thousands of gates.

This method is an alternative approach to test set generation for detectable faults and performs equivalent to the ATALANTA tool [32] with the same number of test pattern sets. Experimental results on benchmark and other similar circuits have shown that the test set predicted has the ability to maintain high fault coverage in the c499 circuit group and a lesser test vector count for five circuit examples, c880, c6288, and c7552, in Table 3 by the proposed technique.

The ML-based methods can be scaled well for larger circuit designs, also. A comparison of fault coverage by the standard tool (ATALANTA) shows that, particularly for the stuck-at fault model, a significantly equivalent test set, and fault coverage of identical range is obtained, saving the test cost compared to expensive EDA tools.

Limitations and Future Work

Although the proposed Decision Tree Regressor (DTR) model provides encouraging outcomes in the prediction of test vectors for benchmark circuits, some limitations persist. This work is confined to a specific regression model and a dataset of around 1000 circuits, possibly limiting the generalization of predictions to larger or more intricate circuit designs.

As future work towards test vector predictions for digital circuits, ensemble techniques like Random Forest or Gradient Boosting Regressor can be applied to improve accuracy and robustness. Furthermore, the model's effectiveness can differ depending on data imbalance and feature selection sensitivity.

Moreover, integrating bigger and diversified benchmark datasets like ISCAS89 with feature optimization approaches may improve the model's scalability and suitability for real-world VLSI testing contexts.

Acknowledgment

For dataset preparation and result comparison, ATALANTA and FSIM (Version 1.2) fault simulator tools were used. The authors are thankful to Dong S. Ha and Virginia Polytechnic Institute and State University for this tool.

References

[1] Wenxing Li et al., “SmartATPG: Learning-based Automatic Test Pattern Generation with Graph Convolutional Network and Reinforcement Learning,” *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Yassine Attaoui et al., “Machine Learning in VLSI Design: A Comprehensive Review,” *Journal of Integrated Circuits and Systems*, vol. 19, no. 2, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Wenji Fang et al., “A Survey of Circuit Foundation Model: Foundation AI Models for VLSI Circuit Design and EDA,” *arXiv Preprint*, pp. 1-62, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] User's Guide for ATALANTA. Virginia Polytechnic & State University, The GitHub Website, 2025. [Online]. Available: https://github.com/hsluoyz/Atalanta

[5] Renjian Pan et al., “Black-Box Test-Coverage Analysis and Test-Cost Reduction Based on a Bayesian Network Model,” *2019 IEEE 37th VLSI Test Symposium (VTS), Monterey*, CA, USA, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Harshad Dhotre et al., “Machine Learning-based Prediction of Test Power,” *2019 IEEE European Test Symposium (ETS)*, Baden, Germany, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[7] Kai-Chieh Yang et al., “ATPG and Test Compression for Probabilistic Circuits,” *2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, Taiwan, pp. 1-4, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Spencer Millican et al., “Applying Neural Networks to Delay Fault Testing : Test Point Insertion and Random Circuit Training,” *2019 IEEE 28th Asian Test Symposium (ATS)*, Kolkata, India, pp. 13-135, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Hauyu Yang et al., “Layout Hotspot Detection with Feature Tensor Generation and Deep Biased Learning,” *Proceedings of the 54th Annual Design Automation Conference 2017*, Austin TX, USA, pp. 1-6, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[10] Aysa Fakheri Tabrizi et al., “Eh ? Predictor : A Deep Learning Framework to Identify Detailed Routing Short Violations from a Placed Netlist,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 6, pp. 1177-1190, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Kurt Hornik, Maxwell Stinchcombe, and Halbert White, “Multilayer Feedforward Networks are Universal Approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989. [CrossRef] [Google Scholar] [Publisher Link]

[12] Maithra Raghu et al., “On the Expressive Power of Deep Neural Networks,” *34th International Conference on Machine Learning*, vol. 70, pp. 2847-2854, 2017. [Google Scholar] [Publisher Link]

[13] Yuzhe Ma et al., “High Performance Graph Convolutional Networks with Applications in Testability Analysis,” *Proceedings of the 56th Annual Design Automation Conference 2019*, Las Vegas NV, USA, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] Kevin Eykholt et al., “Robust Physical-World Attacks on Deep Learning Visual Classification,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 1625-1634, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] R. Selvarasan, and G. Sudhagar, “An Overview on Strategies to Perform VLSI Testing Productively Utilizing BIST,” *International Journal of Advanced Networking and Applications*, vol. 16, no. 2, pp. 6342-6348, 2024. [Google Scholar] [Publisher Link]

[16] Jeffrey E. Nelson, Wing Chiu Tam, and R. D. Blanton, “Automatic Classification of Bridge Defects,” *2010 IEEE International Test Conference*, Austin, TX, USA, pp. 1-10, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[17] Zipeng Li et al., “Test-Cost Optimization in a Scan-Compression Architecture Using Support-Vector Regression,” *2017 IEEE 35th VLSI Test Symposium (VTS)*, Las Vegas, NV, USA, pp. 1-6, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[18] Kaiming Fu, Yulu Jin, and Zhousheng Chen, “Test Set Optimization by Machine Learning Algorithms,” *2020 IEEE International Conference on Big Data (Big Data)*, Atlanta, GA, USA, pp. 5673-5675, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] A. Karthikeyan, and N. Rajeswaran, “Design and Implementation of Multiple Fault Diagnosis on VLSI Circuits using Artificial Neural Networks,” *International Journal of Advances in Engineering & Technology*, vol. 3, no. 2, pp. 685-695, 2012. [Google Scholar] [Publisher Link]

[20] Animesh Basak Chowdhury et al., “Robust Deep Learning for IC Test Problems,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 1, pp. 183-195, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Hideo Fujiwara, “ISCAS ’ 85 Benchmarks : Special Session on ATPG and Fault Simulation,” *1985 IEEE International Symposium on Circuits and Systems*, Kyoto Hotel, Kyoto, Japan, pp. 1-7, 1985. [Google Scholar]

[22] M. Lipovský et al., “A New User-Friendly ATPG Platform for Digital Circuits,” *2016 IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, Kosice, Slovakia, pp. 1-4, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[23] Fangming Ye et al., “Board-Level Functional Fault Diagnosis using Multikernel Support Vector Machines and Incremental Learning,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 2, pp. 279-290, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[24] Manjari Pradhan et al., “Predicting X -Sensitivity of Circuit-Inputs on Test-Coverage: A Machine-Learning Approach,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 12, pp. 2343-2356, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[25] Shi Jin et al., “Efficient Board-Level Functional Fault Diagnosis With Missing Syndromes,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 6, pp. 985-998, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[26] Soumya Mittal, and R.D. Blanton, “LearnX : A Hybrid Deterministic-Statistical Defect Diagnosis Methodology,” *2019 IEEE European Test Symposium (ETS)*, Baden-Baden, Germany, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[27] Yu Huang et al., “Scan Chain Diagnosis based on Unsupervised Machine Learning,” *2017 IEEE 26th Asian Test Symposium (ATS)*, Taipei, Taiwan, pp. 225-203, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[28] Manjari Pradhan, and Bhargab B. Bhattacharya, “A Survey of Digital Circuit Testing in the Light of Machine Learning,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 1, pp. 1-18, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[29] Garima Thakur, Shruti Jain, and Harsh Sohal, “Measurement : Sensors “Current Issues and Emerging Techniques for VLSI Testing - A Review,” *Measurement: Sensors*, vol. 24, pp. 1-9, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[30] Stephan Eggersglüß et al., “On Optimization-based ATPG and its Application for Highly Compacted Test Sets,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 12, pp. 2104-2117, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[31] S. Biswas et al., Automatic Test System for Testing VLSI Circuits, 2017. [Google Scholar]

[32] H.K. Lee, and D.S. Ha, “*Atalanta: An Efficient ATPG for Combinational Circuits*,” Technical Report, Department of Electrical Engineering, Virginia Polytechnic Institute and State University, 1993. [Google Scholar]