**FAULT DETECTION IN COMBINATIONAL CIRCUIT USING MACHINE LEARNING**

Bhuvanchand G1, Naga Chaitanya B1

*1Department of Electronics and Communication Engineering, SRM University–AP, Andhra Pradesh, India*

Email: bhuvanchand\_guntuku@srmap.edu.in

nagachaitanya\_b@srmap.edu.in

**Abstract:** Fault detection and rectification techniques play a crucial role in IC design as digital VLSI circuit complexity increases. For debugging and localizing various types of design defects, many CAD tools and formal methodologies are available. However, the problem of search space explosion with increasing number of logic gates remains the primary concern for IC designers. At the initial stage we have to find the test pattern for the Boolean network. For that we use the Atalanta software which includes the FAN Algorithm. FAN Algorithm is an Automatic test pattern Generation (ATPG), It will generate the test patterns. Using the Atalanta software we achieved a raw data, we converted the raw data into Excel format. Lately, machine learning and artificial intelligence models have been modified for maximum fault coverage through feature extraction and efficient fault detection. In this research, we worked with the machine learning Algorithms SVM and Random Forest. We achieved Higher Accuracy in Random Forest as compared to SVM.

**Keywords:** Automatic test pattern generation (ATPG); FAN Algorithm; SVM; Random Forest;

**INTRODUCTION**

Fault detection in a Boolean network is an important technique in many domains, including computer science, engineering, and biology. Boolean networks are mathematical models used to represent complicated systems with binary (on/off) states. These networks are made up of interconnected nodes that represent variables or components, and logical functions (Boolean functions) establish the interactions between these nodes. The fundamental goal of fault detection in a Boolean network is to detect abnormal or incorrect network behavior. There are many applications for the fault detection in Boolean networks. Here are some applications

1) Fault detection helps in the analysis of a Boolean network's behavior and performance. Identifying faulty nodes or components can provide information about the system's vulnerability, stability, and resilience.

2) Once a fault is detected, it becomes possible to take corrective measures. This may involve repairing or replacing faulty components to restore the proper functioning of the network.

3) Boolean networks are often used to model biological systems, such as gene regulatory networks. Fault detection in these networks can help in understanding the causes of genetic diseases, identifying malfunctioning genes, and designing interventions or treatments.

In this world full of electronics gadgets and integrated circuits ,fault detection plays a vital role to ensure that complexity to use electronic gadgets and integrated circuits decreases. As there is growth in the complexity of digital VLSI circuits,fault detection plays crucial role in the process of IC design. Presence of several faults can affect the performance of those gadgets. One of those faults is "STUCK AT FAULTS''.

STUCK AT FAULT is a type of fault in electronic circuits which makes a specific node or wire stuck at only one level either 0 or 1 irrespective of data transmitted. It leads to many malfunctions or false output. Detecting this stuck at faults is important for working of digital systems.Basically stuck at faults are mainly divided into 2 types [1]

1)stuck at 0(SA0):- This happens when a particular node or line continuously stuck at 0

2)stuck at 1(SA1):- This happens when a particular node or line continuously stuck at 1

We can detect these stuck-at-faults by using different types of algorithms for automatic test pattern generation (ATPG) and some Machine Learning methods. Detecting "stuck-at faults" using ATPG (Automatic Test Pattern Generation) involves the generation of test patterns that can identify whether certain wires in a digital circuit are stuck at logic '0' or '1'.

The primary objective of this project is to detect stuck faults in the given Boolean networks and explore innovative techniques and methodologies for identifying these faults in that circuits.

**RELATED WORKS:**

It's an important step in digital circuit testing and semiconductor fabrication. ATPG is the automatic development of test patterns or input vectors that are used to discover and diagnose flaws or problems in a digital circuit.

**Mathematical methods**

There are some methods to calculate the test patterns manually for the combinational circuits. They are:

1)Tabular Method

2)Boolean-Difference Method

3)Path sensitization Method

**Tabular method:**

For the Tabular method First we have to write all the possible outputs Before and after the Faults (stuck at-0 & stuck at-1) and compare the Both of the outputs where the output changes we can say that there is a fault in the circuit. Those vectors are called test vectors.

**Boolean-Difference Method :**

The set of test vectors can be Detect the Fault Xi /0 means Stuck at 0 is given by

The set of test vectors can be Detect the Fault Xi /1 means Stuck at 1 is given by[2]

**Path sensitization Method:**

Path sensitization is based on the assumption that a gate's failure mechanism causes its inputs or outputs to be stuck at one or zero [3].

They are three steps for this approach

1. Fault Excitation / Fault sensitization
2. Fault propagation
3. Backtracking / Line justification.

By using the above steps we have to find the test pattern generations. Using this method, some Scientists introduce some Automatic test pattern generation (ATPG) Algorithms. Here are some Algorithms mentioned below.

**ATPG ALGORITHMS:**

**D-Algorithm**: The calculus and D-cube-based ATPG algorithms were defined by Roth's D-Algorithm. It is a uni-path directed algorithm. This algorithm is described in Fig. 1.

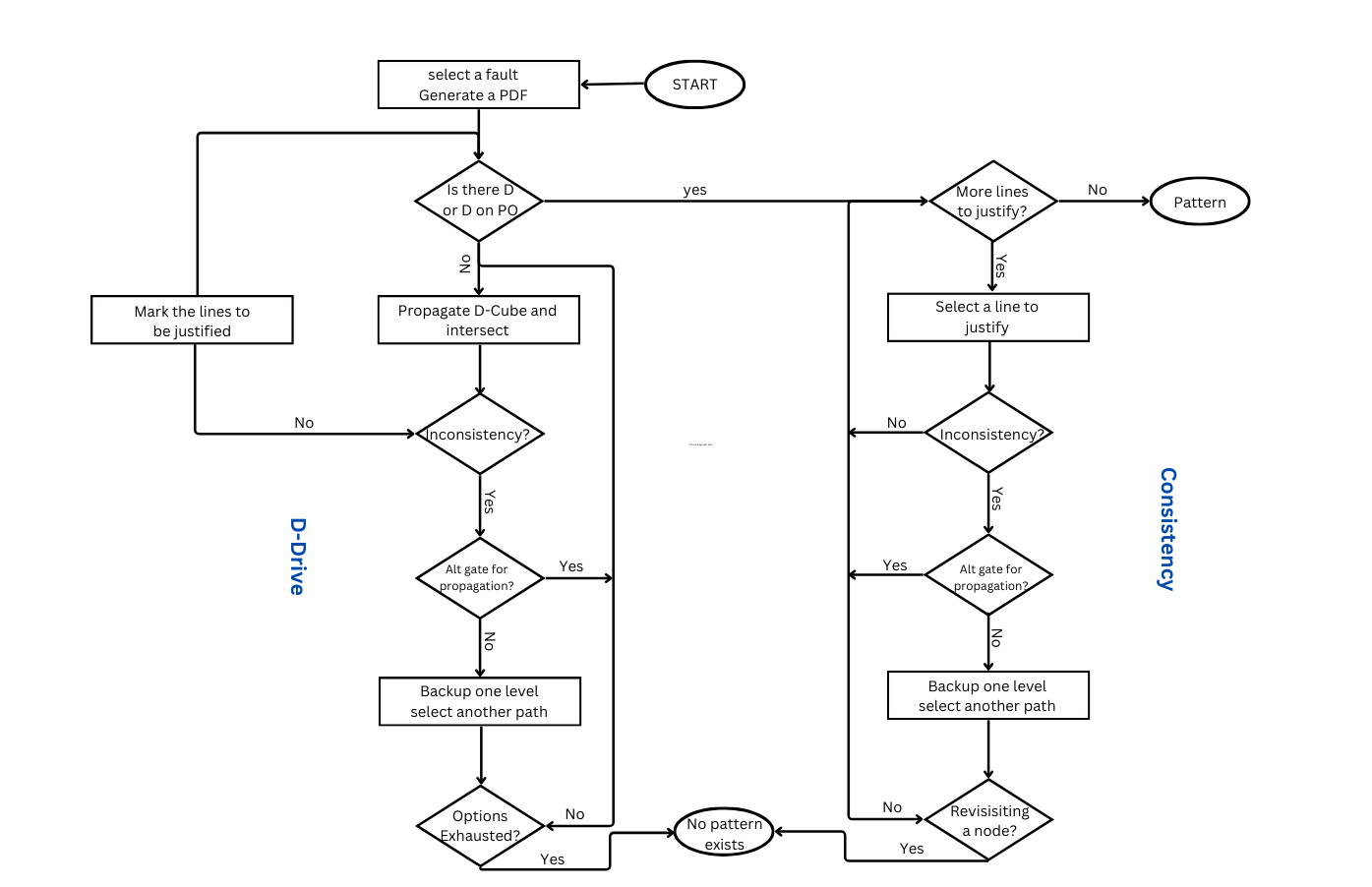


Fig. 1: D- ALGORITHM

**PODEM Algorithm**: (***Path-Oriented Decision-Making):***

To improve reliability, IBM added error correction and translation (ECAT) to their DRAM in the late 1970s. The lack of direction in the search caused the D-algorithm's inability to produce tests for these circuits. Instead of expanding the binary decision tree around every circuit signal, PODEM does so around the Primary Inputs (PIs).As a result, the tree's size drops from 2n to 2num\_PIs. D-CUBES and D-ALGORITHM tended to cross over even after the D-frontier vanished. A subroutine to check for the existence of the D-frontier was introduced by PODEM[4]. After introducing objectives, PODEM discovered that selecting which PIs to set was crucial to effectively achieving objectives. As seen in Fig. 2, back tracing was utilised to determine a PI assignment given an initial goal.

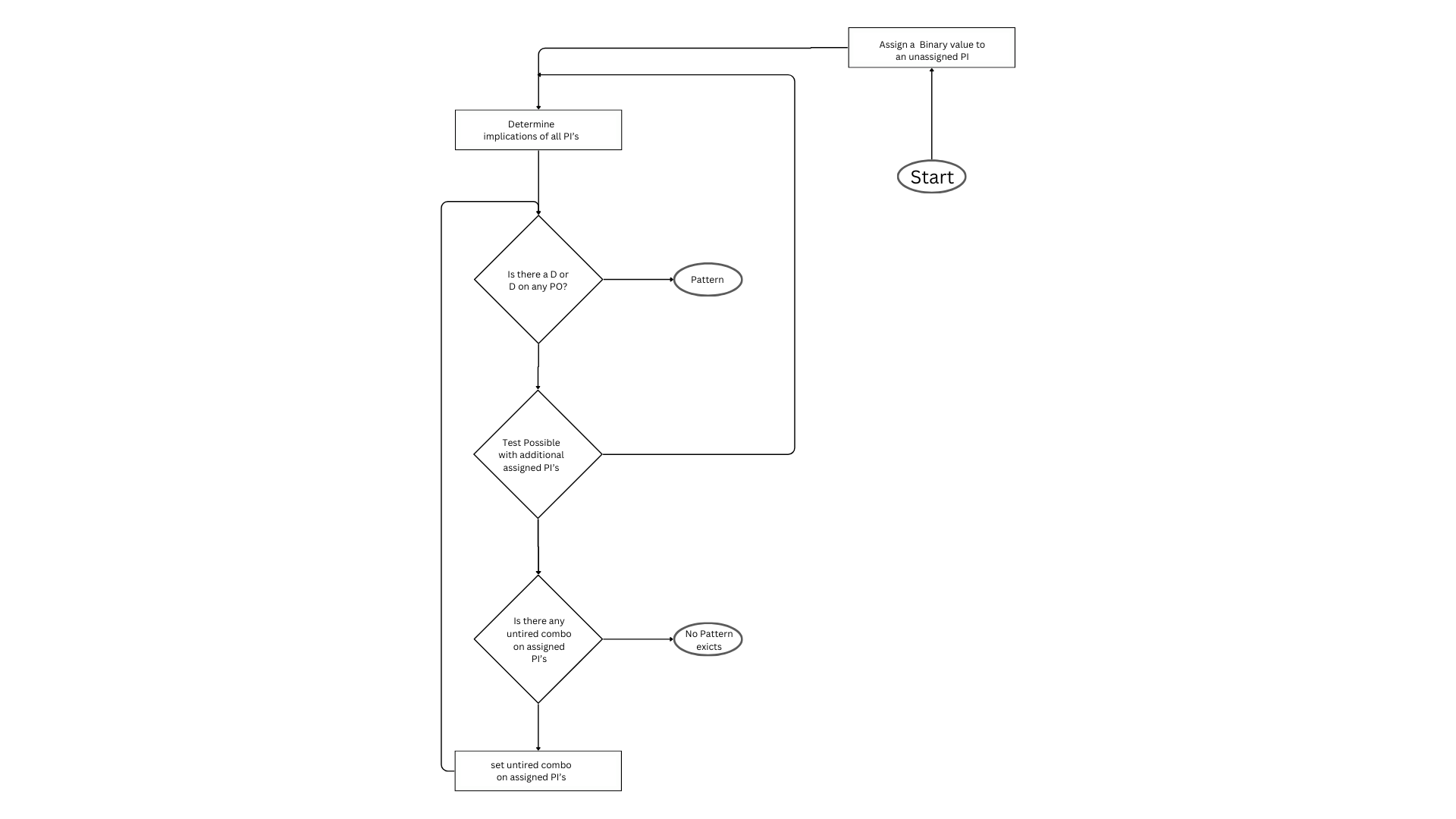


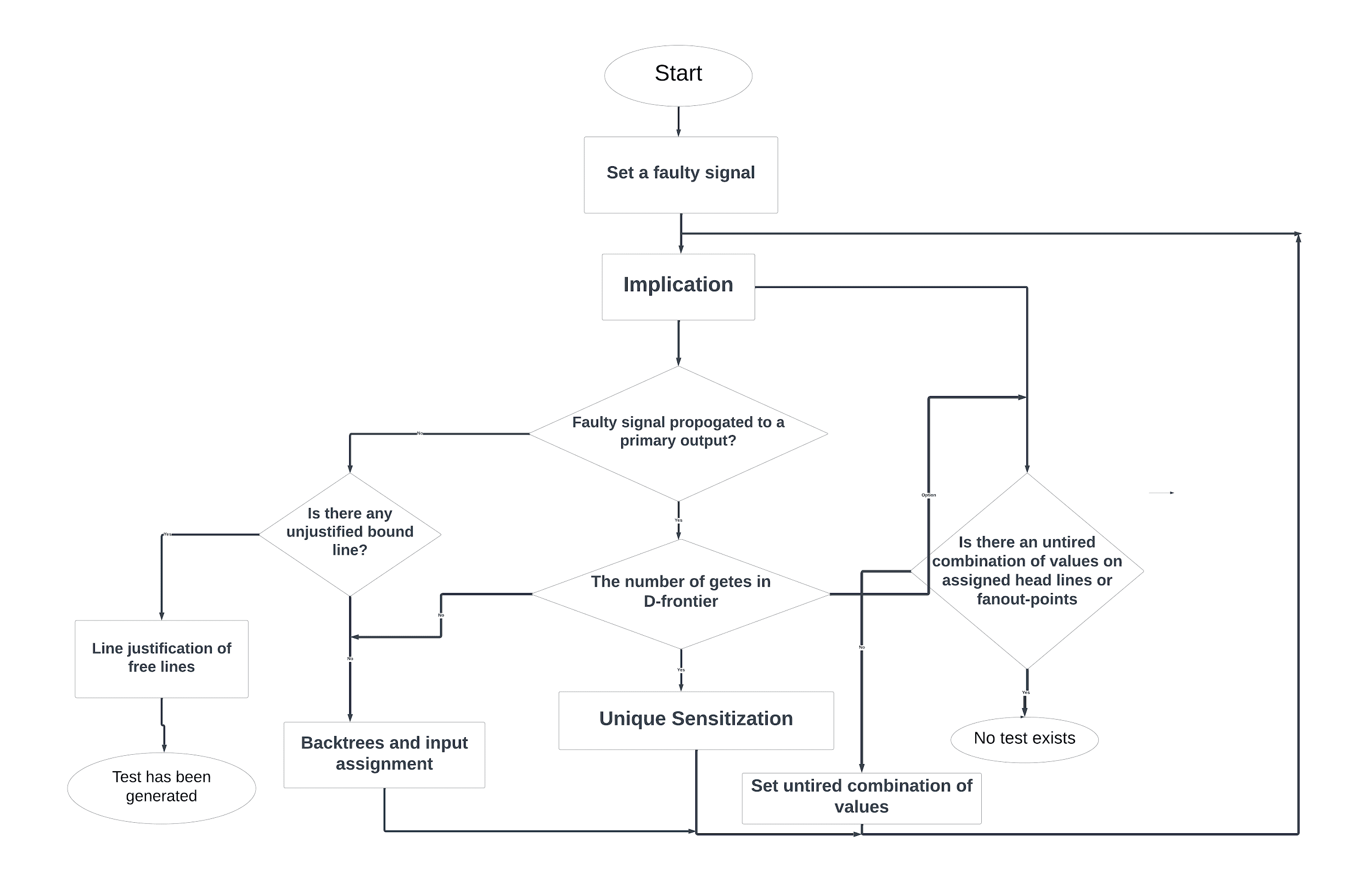
Fig. 2: PODEM ALGORITHM

**FAN Algorithm**: (A Fast Test generation system)

A number of innovative ideas were presented by Fujiwara and Shimono to further restrict the ATPG search space and speed up back tracing. The FAN out algorithm outperforms the PODEM algorithm in terms of efficiency and speed, according to experimental results. In the backtrace of

Fig 3: FAN Algorithm

PODEM, an objective is defined by a pair of an objective value and an objective line [5]. An objective which will be used in the multiple backtrace in FAN. It is defined by a triple

******Atalanta:**

For combinational circuits that are stuck at faults, Atalanta is a modified version of the ATPG (Automatic test pattern generation) tool and fault simulator [6-7]. It uses the parallel pattern single fault propagation technique for fault simulation and the FAN algorithm for test pattern generation.

**MACHINE LEARNING ALGORITHMS**

Machine learning is a computer technique where computers learn to make decisions or predictions from data without being explicitly programmed. It is like teaching a computer to recognize patterns and make sense of information on its own. By analyzing examples and finding patterns in the data, machine learning enables computers to improve their performance over time and give predictions which are very accurate or decisions on new or unseen data for the trained model.

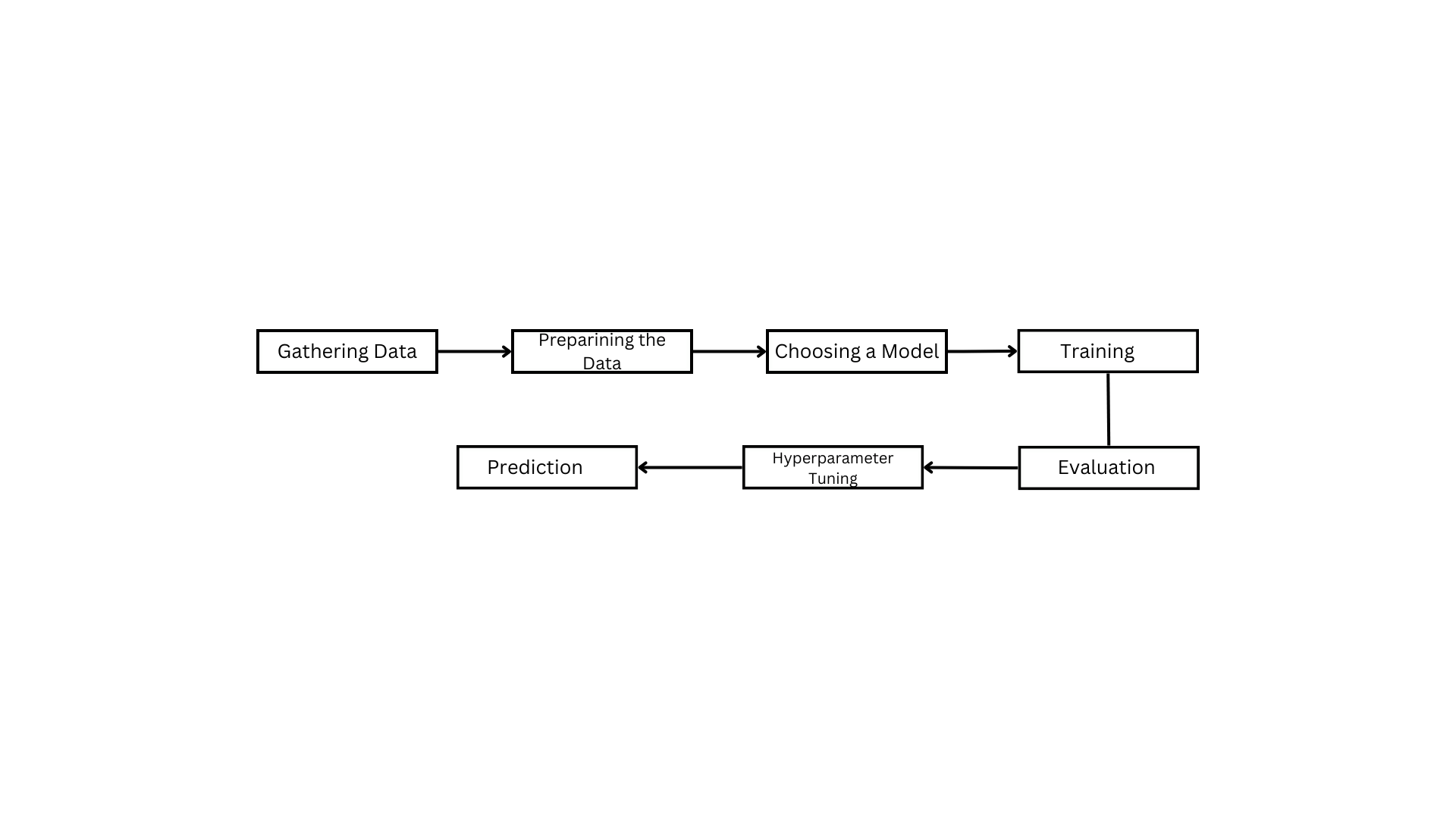
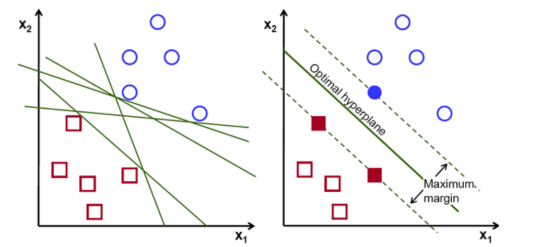


Fig. 4: Machine learning Basic Flow Chart

**Proposed Schemes:**

**Support Vector Machine (SVM)** :

Using supervised machine learning, SVM is a reliable algorithm for classification tasks. It works especially well for binary classification problems, such as separating spam emails from non-spam emails or benign tumors from malignant ones. In a high-dimensional feature space, SVM finds the hyperplane that best divides the data points of various classes. The margin, or distance, between the two nearest data points for each class is what it seeks to maximize[8–9]. The closest points—referred to as support vectors—are essential in establishing the decision boundary. By utilizing the kernel trick, SVM can handle data that is not linearly separable. By using this method, the data may become linearly separable in a higher-dimensional space that is mapped from the original feature space. Polynomial, sigmoid, and radial basis function (RBF) kernels are examples of common kernel functions [15].

Fig. 5: Support vector machine (SVM)

**Random Forest [10]:**

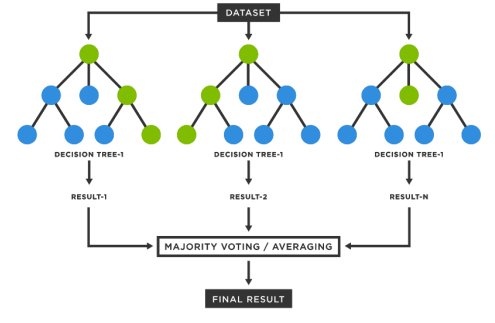
Popular machine learning algorithm Random Forest is a member of the supervised learning class. Its foundation is the idea of ensemble learning, which is the process of merging several classifiers to solve a challenging issue and enhance the model's functionality.It can be applied to ML problems involving both classification and regression.A Random Forest classifier uses multiple decision trees on different dataset subsets and averages them to increase the dataset's predictive accuracy.A forest with a larger proportion of trees produces better accuracy and avoids the overfitting issue[13].

Fig.6 Random Forest

**outcomes:**

**Precision**:

In machine learning, precision is the measure of a model's positive prediction accuracy. Out of all the instances that the model predicted as positive, it calculates the percentage of accurate positive predictions.

**Recall:**

Recall quantifies the model's capacity to accurately identify every positive instance among the entire set of real positive instances found in the dataset.

**F1-Score**:

Through machine learning, the F1 score aggregates recall and precision into a single number. It considers both false positives and false negatives, giving a fair assessment of a model's performance. The F1 score is a number between 0 and 1, where 0 is the lowest possible score and 1 is the best possible score (ideal precision and recall).

**Accuracy**:

One of the most important evaluation metrics for evaluating a classification model's performance is accuracy. It calculates the percentage of accurate predictions the model produced relative to the total number of predictions.

**Confusion matrix**:

A common table used to assess a classification model's performance is the confusion matrix. We are able to examine the quantity of true positives, true negatives, false positives, and false negatives by using the comprehensive breakdown that it offers between the model's predictions and the actual class labels.

****  **Actual values**

**1 0**

|  |  |
| --- | --- |
| True Positive  (TP) | False Positive  (FP) |
| False Negative  (FN) | True Negative  (TP) |

**1**

**0**

**EXPERIMENTAL STUDY AND RESULT ANALYSIS**

**Atalanta Results:**

In this section, the experimental results are shown for finding the stuck-at-0 and stuck-at-1 faults. Implementation of the suggested technique uses C17 combinational circuit from the ISCAS’85 benchmark. From ISCAS’85, C17 has been transferred to Atlanta Software. The Atalanta Software generating number test patterns for each fault recognizes all stuck-at-0 and struck-at-1 faults.

* Circuit type: c17
* Inputs and outputs:7
* Faults:22
* Test vectors:54
* Fault coverage:100

**Machine Learning Results:**

Table 2 describes the accuracy levels attained by the random forest and SVM algorithms. The percentage of accurate classifications a trained machine learning model makes, or the ratio of accurate predictions to all other predictions combined, is used to determine how accurate the model is. According to table 2, for the provided C17 benchmark circuit, Random Forest provides better accuracy than svm model.

**Table 2: Accuracy of fault detection using SVM and Random Forest Algorithms**

| **Model Name** | **Accuracy** |
| --- | --- |
| SVM | 0.30 |
| Random Forest | 0.82 |

**SVM Results:**

Tables 3 and 4 provide examples of the SVM and Random Forest Model's parameters. Important performance metrics like precision, F-score, and recall are used to assess the efficacy of classification models like random forest and SVM models. Tables 3 and 4 contain the parameter values based on various Test Vectors. Model’s accuracy is shown in percentage. Recall quantifies the proportion of pertinent data points that the model correctly identified. F1 Score takes into account recall and precision. It is the precision and recall harmonic mean (average). Precision (p) and recall (r) should be balanced in the system for the best F1 Score.

**Table 3 : SVM Model Results**

| **SVM MODEL** | **Recall** | **Precision** | **F1-score** |
| --- | --- | --- | --- |
| 00 | 1.0 | 0.27 | 0.42 |
| 01 | 0.00 | 0.00 | 0.00 |
| 10 | 0.33 | 0.75 | 0.46 |
| 11 | 0.00 | 0.00 | 0.00 |

** Random Forest Results:**

**Table 4: Random Forest Model Results**

|  |  |  |  |
| --- | --- | --- | --- |
| **Random Forest MODEL** | **Recall** | **Precision** | **F1-score** |
| 00 | 0.75 | 0.75 | 0.75 |
| 01 | 0.80 | 0.80 | 0.80 |
| 10 | 0.83 | 0.62 | 0.71 |
| 11 | 0.83 | 0.94 | 0.88 |

**Conclusion:**

In this work, an approach for detecting stuck at faults in VLSI circuits using Random forest Machine learning algorithm has been proposed. There are many mathematical models to find the Test Patterns of particular Benchmark Circuit and there are many ATPG algorithms like "D Algorithm", "FAN Algorithm", "PODEM Algorithm" In our work, we used ATALANTA software for generating test patterns of respective benchmarks . Then we Introduced machine learning algorithms namely "SUPPORT VECTOR MACHINE(SVM)" and "RANDOM FOREST". Using voting or averaging across several decision trees, Random Forest is an ensemble machine learning algorithm that reduces overfitting and increases accuracy in prediction-making. As we tried detecting test patterns in VLSI circuits using two Machine learning algorithms , The maximum fault coverage by Random forest delivers around 82% accuracy in detecting faults whereas SVM model has only 30% accuracy.

**Future work:**

In the future, an extension of this project could involve the development and implementation of advanced fault localization techniques in VLSI, with the goal of achieving precise fault location identification. This may use deep learning algorithms namely neural networks and different types of autoencoders like Stacked Sparse autoencoders (SSAE) for improving overall fault localization in integrated circuits. This may also increase the robustness of the system or integrated circuit.

**References:**

[1] Li, J. Chien-Mo. "Diagnosis of single stuck-at faults and multiple timing faults in scan chains." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 13.6 (2005): 708-718.

[2] Smith Jr, George W. "Fault detection test derivation using Boolean difference techniques." *Proceedings of the ACM annual conference-Volume 1*. 1972.

[3] Chiang, Albert CL, Irving S. Reed, and Anthony V. Banes. "Path sensitization, partial boolean difference, and automated fault diagnosis." *IEEE Transactions on Computers* 100.2 (1972): 189-195.

[4] Goel, Prabhakar, and Barry C. Rosales. "PODEM-X: An automatic test generation system for VLSI logic structures." Papers on Twenty-five years of electronic design automation. 1988. 412-420.

[5] Lai, K., and Parag K. Lala. "Multiple fault detection in fan-out free circuits using minimal single fault test set." IEEE transactions on computers 45.6 (1996): 763-765.

[6] Gaber, Lamya, Aziza I. Hussein, and Mohammed Moness. "Fault Detection based on Deep Learning for Digital VLSI Circuits." *Procedia Computer Science* 194 (2021): 122-131.

[7] Arvind Raghuraman, "ELEC-7250 Course Project:-o Generate a Single Test Vector to detect all/most number of faults in a given combinational circuit".

[8] T. -W. Kuan, J. -F. Wang, J. -C. Wang, P. -C. Lin and G. -H. Gu, "VLSI Design of an SVM Learning Core on Sequential Minimal Optimization Algorithm," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 4, pp. 673-683, April 2012, doi: 10.1109/TVLSI.2011.2107533.

[9] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, *13*(4), 18-28.

[10] Parmar, Aakash, Rakesh Katariya, and Vatsal Patel. "A review on random forest: An ensemble classifier." *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*. Springer International Publishing, 2019..

[11] Ferrari, F. "System-on-a-chip verification~ methodology and techniques." *IEEE Circuits and Devices Magazine* 18.6 (2002): 39-39.

[12] Gontara, Salah, Amine Boufaied, and Ouajdi Korbaa. "Fault localization algorithm in computer networks based on the boolean particle swarm optimization." *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019.

[13] Arrigoni, Viviana, Novella Bartolini, Annalisa Massini, and Federico Trombetti. "A Bayesian Approach to Network Monitoring for Progressive Failure Localization." *IEEE/ACM Transactions on Networking* (2022).

[14] Fornasini, Ettore, and Maria Elena Valcher. "Fault detection analysis of Boolean control networks." *IEEE Transactions on Automatic Control* 60.10 (2015): 2734-2739.

[15] Ettore, Fornasini, and Valcher Maria Elena. "Fault detection problems for Boolean networks and Boolean control networks." In *2015 34th Chinese Control Conference (CCC)*, pp. 1-8. IEEE, 2015..

[16] Thakar, Sarita. "On the generation of test patterns for combinational circuits." PhD diss., Virginia Tech, 1993.

[17] Goel. "An implicit enumeration algorithm to generate tests for combinational logic circuits." *IEEE transactions on Computers* 100, no. 3 (1981): 215-222.

.