# Machine-Learning Techniques for Predicting Post-CTS Worst Negative Slack

**G.M.G.Madhuri[1] and Dr.J.Selvakumar[2]**

1 Department of Electronics & Communication, College of Engineering & Technology, SRM Institute of Science and Technology, Kattankulathur - Chennai, India

2 Department of Electronics & Communication, College of Engineering & Technology, SRM Institute of Science and Technology, Kattankulathur - Chennai, India

## ABSTRACT

This paper focuses on predicting post clock tree synthesis (CTS) parameters using pre-CTS parameters. This paper a comparative analysis of three different prediction techniques: Multiple Regression, Random Forest, and Regression-based Decision Tree are conducted. Specifically, the prediction of post-CTS Worst Negative Slack is performed using a dataset consisting of 14 ISCAS'89 benchmark circuits. To evaluate the prediction models, the total dataset is randomly partitioned to 70% as the training set and the remaining 30% as the testing set. The proposed model is validated with two distinct technology nodes, namely TSMC 45nm and TSMC 90nm, to ensure its robustness The experimental findings demonstrate that mean square error for both technology nodes is found to be on the order of $10^{-5}$, indicating high accuracy in predicting post-CTS parameters. Additionally, the R2 score is calculated to be 0.998. Among the three prediction techniques, the results obtained using the Random Forest technique exhibit significant improvements.

**Keywords: Clock tree synthesis, worst negative slack, machine learning, supervised learning algorithms.**

## 1. Introduction

In the present scenario, the growth of CMOS technology has led to increasingly complex System-on-Chip (SoC) designs. In order to bring products to the market in a timely manner, the design time plays a critical role. Prolonged design time can significantly delay the release of integrated circuits (ICs), resulting in increased costs and a higher risk of failure. Therefore, it is imperative to explore methods to expedite the design process while maintaining high chip quality. The physical design (PD) process in integrated circuits encompasses various stages such as layout, placement, clock tree routing, and verification. The simulation and verification steps consume a considerable amount of time. When the layout is finalized as a GDS II file, timing closure becomes crucial. In order to expedite the backend design process, it is essential to accurately predict the final timing to identify and rectify potential errors at an early stage. Clock tree synthesis (CTS) plays a vital role in the physical design as clock networks consume a significant amount of power in full-chip designs[1-3] . The idea of CTS is to generate a well-designed clock tree structure that meets the design

specifications. A well-designed clock tree helps to mitigate major design issues like power consumption, routing challenges, and time-consuming timing closure. Timing violation occurs in the chip when clock is not propagated to all flops at right time. If paths don't meet time constraints, chip fails. A timing path typically begins at one flop and finishes at another, and traversing through combinational logic gates present between the two flops F1 and F2 as shown in Figure 1.

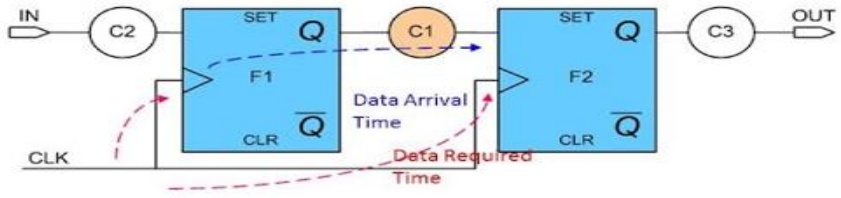


Fig.1. Slack

The clock signal is represented as Clk, When F1 is activated with the clock input, Q of F1 gives out its output. Before the needed time $T_{req}$, the data signal from Q of F1 must flow through the combinational logic C1 and verify the signal at the input pin D2 of F2. Slack may then be acquired by equation(1):

$$Slack=T_{req}-T_{arrival}(1)$$

Negative value of slack indicates timing violation .The path which has negative value for slack is termed as critical path.Worst Negative Slack (WNS) represents the most severe violation among all the negative slack values in a design. Critical paths, which have the greatest negative slack values, exert the most significant influence on circuit timing. Identifying the critical path is aided by WNS. Designers aim to reduce WNS to enhance overall circuit performance and ensure compliance with timing constraints. Achieving this often necessitates design iterations, including modifications to physical and logical parameters, optimization of location and routing, and overall design improvements to enhance timing properties. By accurately estimating parameters such as clock power, and worst negative slack, designers can make informed decisions to optimize designs, reduce iteration cycles, and improve overall circuit performance.

The process of identifying the optimal input parameters to meet design objectives becomes arduous due to the vast number of potential variables.To meet this requirement machine learning techniques have been explored as powerful tools for variety of applications indiverse domains. Machine learning techniques have gained prominence in solving complex problems by utilizing data and past experiences. Machine learning comprises three main components: tasks, models, and features. Common tasks include classification, regression, grouping, anomaly detection, and sorting, while popular models include neural networks, tree models, support vector machines, and linear models.

Several researchers have proposed applying machine learning (ML) algorithms to automatically estimate clock network metrics. For instance, Kahng et al. [4] used meta modelling tools to compute insertion delay and skew. Jeong et al. [5] utilized ML techniques to predict routing methodology and metrics such as power, performance, and area (PPA). In another study, Kahng et al. [6] proposed statistical learning and metamodeling techniques to forecast clock wire length and power. Additionally, Kahng

et al. [7] investigated the impact of non-uniform sinks and placement aspect ratios. Furthermore, Y. Kwon [8] employed artificial neural networks (ANNs) to predict transient clock power consumption.

Z. Qi [9] introduced a machine-learning technique for predicting congestion in routing using regression model. W. Cheng et al. [10] analyzed macro placement to forecast routing wire length after placement completion. D. Hyun [11] employed a regression-based machine learning approach. Y.C. Lu et al. [12] introduced a GAN-CTS method to estimate CTS results such as clock power and wire length. However, it requires placement and trial routing, which increases runtime and lacks the ability to assess transient clock power. Jain [13] proposed using an artificial neural network to predict post-CTS quality of results (QoR) based on pre-CTS QoR report. Various machine learning techniques have been applied to automate different stages of the design process and forecast clock network metrics [14]. This research paper aims to overcome the challenges in predicting the Worst Negative Slack (WNS) by exploring the potential of machine learning techniques post-clock tree synthesis (CTS) using pre-CTS parameters. The primary contributions of this study are as follows:

1. The paper presents a comprehensive comparative analysis of various machine-learning methodologies for predicting post-CTS WNS. Techniques such as multiple regression, random forest, and regression-based CART decision tree are evaluated to assess their effectiveness in capturing the complex relationships between pre-CTS parameters and post-CTS WNS.

2. To validate the proposed methodologies, a set of 14 benchmark circuits from the ISCAS'89 dataset is selected. These circuits represent diverse design characteristics and provide a realistic evaluation platform for assessing the predictive accuracy and generalization capabilities of the machine-learning models.

3. The predictive performance of the machine-learning models is assessed using metrics such as mean square error (MSE) and R2 score. By comparing the results obtained for different technology nodes (90nm and 45nm), the study provides insights into the models' robustness and accuracy across different design technologies.

This work offers a valuable contribution by thoroughly examining and contrasting various machine-learning methods in their ability to predict post-CTS WNS.By demonstrating the effectiveness of these techniques, the paper opens avenues for incorporating automated prediction models into the IC design flow, leading to improved design quality, reduced design time, and enhanced overall efficiency.

The paper is organized as follows: Section II presents an overview of supervised learning techniques. Section III delves into the proposed methodology. Section IV concentrates on the examination of results and subsequent discussions. Section V concludes the research work and discusses future avenues for exploration.

## 2. Prelımınarıes

Machine learning[15] has gained significant traction as a powerful approach in various fields. Its ability to extract meaningful features from data and apply specific learning methods to train models has proven valuable in solving complex problems. An overview of the different techniques used in machine learning is shown in Fig2. One prominent branch of machine learning is supervised learning. Supervised learning

encompasses the utilization of labeled datasets to train computers in accurate data classification or outcome prediction. The model's weights are iteratively adjusted as data is inputted, ensuring a suitable fit through the cross-validation technique.When fitting a prediction model to a collection of data, linear regression can be employed to provide a structural interpretation that facilitates hypothesis testing.

Training and evaluating the machine learning model involves using multiple datasets with the same design. However, if the data does not provide crucial information for the learning task, the effectiveness of the model may be compromised. Therefore, careful selection of relevant features is crucial for improving the quality of the model.

The linear regression model is defined in Eq(2) as follows:

$$y_i=\beta_0+x_1\,\beta_1+x_2\,\beta_2+\cdots+x_i\,\beta_i+e_i\,,\ i=1,2,\ldots,n \quad (2)$$

where, i= the index of the data points, and the dependent variable ,
$y_i$ = the covariates of $x_j$
$e_i$ = error term that is normally distributed and are estimated from the available data.

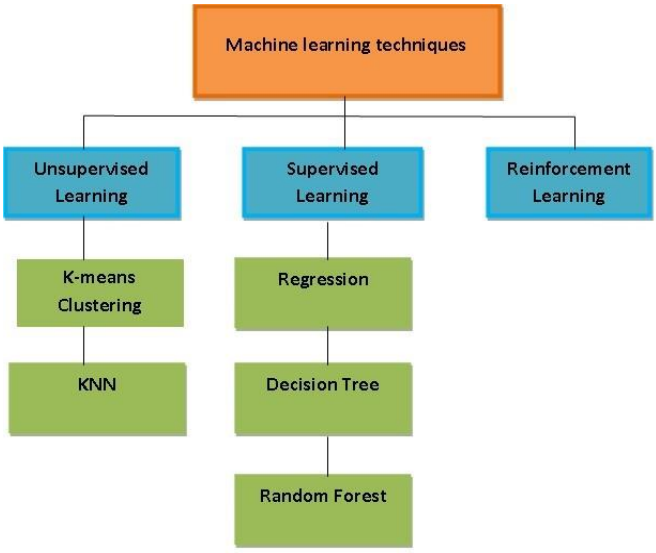


**Fig.2.** Machine learning techniques

The CART algorithm, alternatively referred to as a classification and regression tree, is widely used in machine learning for both classification and regression purposes. Its implementation involves creating a binary tree structure that progressively splits the input data using decision rules. This process continues until the tree produces leaf nodes that contain the predicted output values.The first step is to prepare the input data. Each data instance consists of a set of features (independent variables) and a target variable (dependent variable). The features could be numerical or categorical, and the target variable is typically categorical for classification or numerical for regression. CART constructs the decision tree in a top-down manner, starting from the root node. It selects the optimal criterion, either mean squared error for regression or Gini impurity for classification, to determine the best feature for splitting the data.

The feature that minimizes the impurity or error is chosen. Once a feature is selected, the data is partitioned into two subsets based on a threshold value for that feature. Instances with values greater than or equal to the threshold go to the left child node, while instances with values less than the threshold go to the right child node. The splitting process is applied recursively to each child node until certain stopping criteria are met. Some potential criteria for constructing the CART tree include setting a maximum depth, ensuring a certain number of instances in a node, or aiming for a minimum decrease in impurity or error. After constructing the CART tree, new instances can be classified (for classification problems) or predicted for their target values (for regression problems) by following the decision tree from the root node to the leaf node, using the input instance's feature values.

The Random Forest algorithm is frequently employed in machine learning for classification and regression purposes. Random forest is an ensemble technique that leverages numerous decision trees inorder for prediction generation. This algorithm is particularly effective as it can effectively manage a significant number of features and is less susceptible to overfitting, which is a common drawback of using individual decision trees. Its strong points include its broad applicability across different fields, its high accuracy, robustness, and capability to handle both numerical and categorical data.

The mean squared error (MSE) is a widely used measure for quantifying the average squared deviation between actual values and predicted values in a regression problem. It serves as a numerical indicator which defines regression model best fits the data. The MSE is computed as follows:obtain the difference of predicted and actual values and by squaring the difference, then averaging them, and penalizing larger errors more severely than smaller ones. A lower MSE value signifies a superior fit of the regression model to the given data.

Mathematically, MSE is calculated using:

$$MSE = Average((y\_pred - y\_actual)^2) \quad (3)$$

Where,
y_pred = the predicted values
y_actual =the actual values
n = the total number of observations.

The R2 score, alternatively named as the coefficient of determination, evaluates the accuracy of a regression model in relation to the actual data. It measures the variance of the dependent variable is accounted for, by the model's independent variables. The R2 score is on a scale from 0 to 1, with a value of 1 signifying a good fit and a value of 0 indicating that the model fails to describe variation in the dependent variable.
The calculation for the R2 score is as follows:

$$R2 = 1 - (SSR/SST) \quad (4)$$

SSR, or the Sum of Squared Residuals, is calculated by squaringthe difference of the predicted values and the actual values of the dependent variable and sum them. On the other hand, SST, or the Total Sum of Squares, is obtained by summing the squared differences between the actual values and the mean of the dependent variable. A higher R2 score indicates a stronger fit of the model to the data, indicating that a larger portion of the variability in the dependent variable is explained by the independent variables.

## 3 . Methodology

In our model, physical design (PD) flow considers the input parameters obtained from floor plan (.def), constraints (.sdc), and net list (.ddc) files to initiate the process, and reports can be generated at each step to gather stage-specific data. The database creation involved utilizing the Genus RTL synthesis tool to synthesize the net lists and Cadence Innovus Implementation System v18.1 for gathering the placement and clock tree synthesis (CTS) procedures. The study employed technology nodes of 45nm and 90nm. Table I displays the ISCAS'89 benchmark circuits. The Python3 programming language was used to implement the proposed framework for clock tree synthesis prediction, and it was executed on Google Colab, a cloud-based development environment. In order to generate the necessary database for the framework, Cadence Innovus[21], a popular tool for physical design, is utilized. For the implementation of the regression model in Python, various predefined modules are leveraged, including sklearn, matplotlib, seaborn, pandas, and numpy. These modules provide essential functionalities for data processing, model training, and performance evaluation. Sklearn offers a comprehensive set of machine learning algorithms and tools.

Table 1.Benchmark cırcuıts

| Bench mark | Circuit used | size |
|---|---|---|
| ISCAS'89 circuits | S298 | Small |
| | S344 | |
| | S382 | |
| | S400 | |
| | S832 | |
| | S1196 | |
| | S1238 | Medium |
| | S5378 | |
| | s13207 | |
| | s15850 | |
| | s35932 | |
| | s38417 | Large |
| | s38584 | |

Table 2 displays the modeling parameters and the corresponding range of values used for prediction.

Table 2.Input parameters and their values

| Name of the Parameter | Value Range |
|---|---|
| Aspect ratio | 0.7,0.75,0.8,1.0 |
| Core Utilization | 0.6,0.7,0.75,0.8,1.0 |
| Fanout | 50,100 |
| Clock Latency | 1.2 |
| Clock Skew | 1.0 |
| Max transition | 0.2,0.5 |
| Max cap | 0.3 |

In our proposed model, the first step is to prepare the input data. Each data instance consists of a set of features (independent variables) and a target variable (dependent variable). When the input parameters from Table 2 and the pre-CTS data along with post CTS data are concatenated together to provide the input data. The next phase involves a split, as shown in Fig 3.

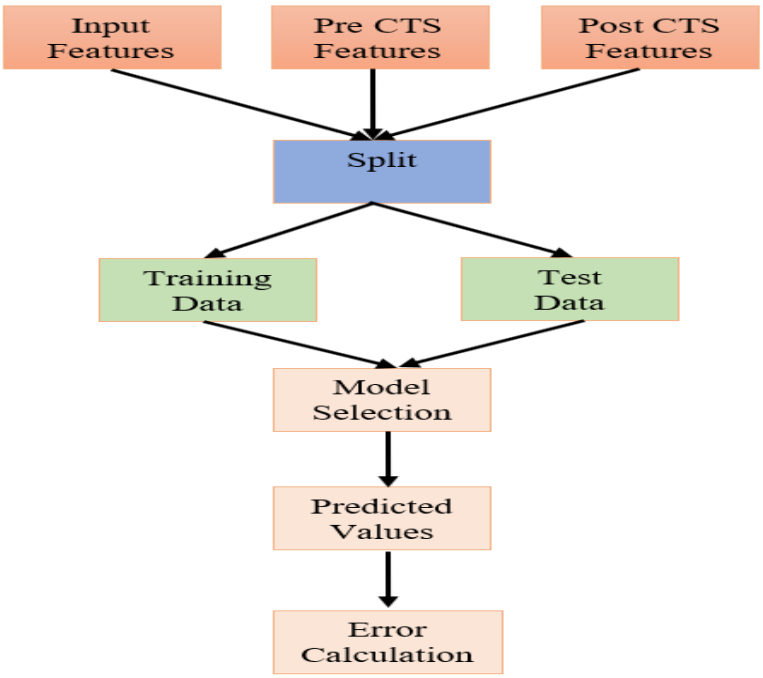


Fig.3. Proposed flow chart

In our proposed model, the initial step involves data preparation. Each data instance consists of a set of independent variables (features) and a dependent variable (target). The input data comprises the input parameters from Table II, as well as the pre-CTS and post-CTS data, concatenated together.The next phase includes a split, as shown in Figure 3. Our model splits the training and testing data in a 70:30 ratio. Model selection considers Linear Regression initially, followed by CART-based Decision Trees and Random Forest.After the completion of training process, the same model is used for predictions. We run predictions on the X_test data and save the results as Y_pred. To evaluate the performance of the regression models, we calculate the Mean Square Error (MSE) and R2 score using Y_pred and Y_test as inputs. The MSE serves as a commonly used metric to assess the model's performance, while the R2 score indicates the residual error or the disparity between the actual data points and the predicted values generated by the regression model.

## 4.Results and Dıscussıon

As discussed earlier to ensure proper evaluation and validation of the proposed framework, the dataset is divided into two distinct subsets: the training data, which accounts for 70% of the total dataset, and the testing data, comprising the remaining 30%.

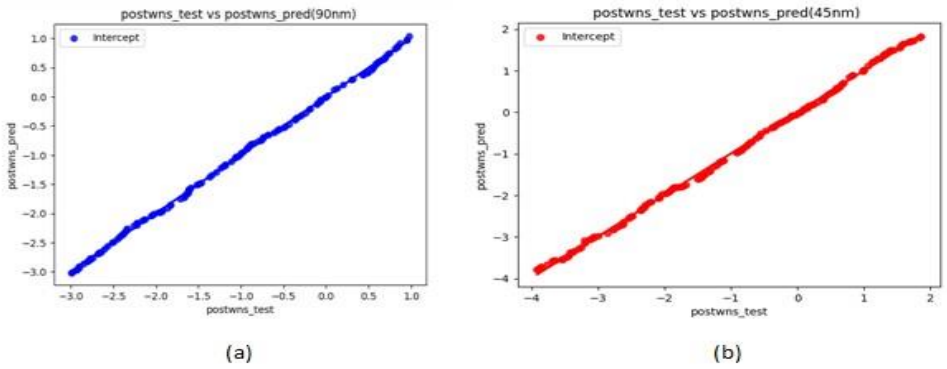


**Fig.4. WNS comparision between predicted data and test data for both technologies**

Figure 4 show the comparison between the actual data that was tested and the data that was projected for both of the technology nodes. A marginal difference may be seen between the predicted data and the test data, which clearly demonstrates that just a marginal difference exists between the two sets of data. The line y=x shows that the predicted results are the same as the test data; nevertheless, the variation is greater for the 45nm technology compared to the 90nm technology.In Table 3, for the random forest technique, the achieved R2 score is an impressive 0.999835, which signifies a highly accurate prediction capability.

Table 3.Errors obtained through various techniques

| **Technology Node** | **90nm** | | **45nm** | |
|---|---|---|---|---|
| Technique | MSE | R2 Score | MSE | R2 Score |
| Multi Regression | 8.82x10-4 | 0.89 | 6.82x10-4 | 0.87 |
| CART Decision Tree | 5.48x10-5 | 0.99996 | 3.48x10-5 | 0.99986 |
| Random Forest | 2.3x10-5 | 0.9999835 | 2.2x10-5 | 0.99815 |

The comparison results of post-CTS WNS with predicted WNS of proposed model. Using Random forest technique, the values obtained for the test circuits are similar to predicted values .It has been observed that the random forest approach demonstrates exceptional accuracy in predicting the WNS values. The predictions obtained using the random forest technique closely match the values generated by the tool, indicating the effectiveness of this method. Furthermore, the comparative analysis of the results is conducted for two different technology nodes.

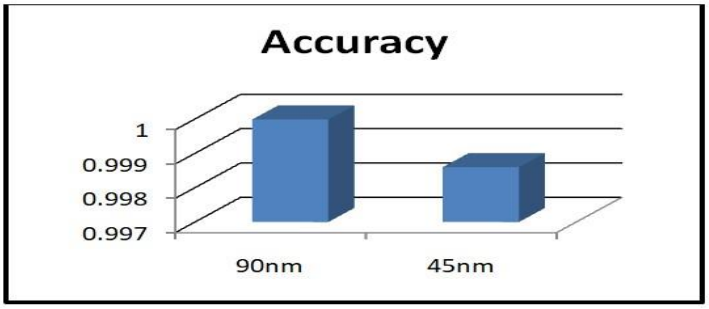

Fig.5.Accurcy obtained

Our result's accuracy was far better than post-CTS, and it held up even as the data size increased. Accuracy result is shown in figure 5.For 45 nm technology accuracy is slightly less as the predicted values slightly differ from actual values.

## 5 . Conclusion

This paper has presented a comparative study of various machine-learning methodologies for predicting the Worst Negative Slack (WNS) after clock tree synthesis (CTS) based on pre-CTS parameters. Among the machine learning techniques evaluated, it has been observed that the random forest approach demonstrates exceptional accuracy in predicting the WNS values. The predictions obtained using the random forest technique closely match the values generated by the tool, indicating the effectiveness of this method. Furthermore, the comparative analysis of the results is conducted for two different technology nodes, namely 90nm and 45nm.This study enhances the current understanding of machine learning approaches for foreseeing post-CTS WNS in integrated circuit designs. It highlights the potential of machine learning in the realm of physical design, particularly in the prediction of post-CTS WNS using pre-CTS parameters.

## 7. References

[1] E. Salman and E. G. Friedman, High Performance Integrated Circuit Design. New York, NY, USA: McGraw-Hill, 2012.

[2] Friedman, Eby G. "Clock distribution networks in synchronous digital integrated circuits." Proceedings of the IEEE 89, no. 5 (2001): 665-692.

[3] Pilling, David J., and Henry B. Sun. "Computer-aided prediction of delays in LSI logic systems." In Proceedings of the 10th Design Automation Workshop, pp. 182-186. 1973.

[4] Kahng, Andrew B., Bill Lin, and Siddhartha Nath. "Enhanced metamodeling techniques for high-dimensional IC design estimation problems." In 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1861-1866. IEEE, 2013.

[5] Jeong, Kwangok, Andrew B. Kahng, Bill Lin, and Kambiz Samadi. "Accurate machine-learning-based on-chip router modeling." IEEE Embedded Systems Letters 2, no. 3 (2010): 62-66.

[6] Kahng, Andrew B., Bill Lin, and Siddhartha Nath. "High-dimensional metamodeling for prediction of clock tree synthesis outcomes." In 2013 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP), pp. 1-7. IEEE, 2013.

[7] Kahng, Andrew B., and Stefanus Mantik. "A system for automatic recording and prediction of design quality metrics." In Proceedings of the IEEE 2001. 2nd

International Symposium on Quality Electronic Design, pp. 81-86. IEEE, 2001.

[8] Kwon, Yonghwi, Jinwook Jung, Inhak Han, and Youngsoo Shin. "Transient clock power estimation of pre-CTS netlist." In 2018 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-4. IEEE, 2018.

[9] Z. Qi, Y. Cai and Q. Zhou, "Accurate prediction of detailed routing congestion using supervised data learning," 2014 IEEE 32nd International Conference on Computer Design (ICCD), 2014, pp. 97- 103, doi: 10.1109/ICCD.2014.6974668.

[10] Cheng, Wei-Kai, Yu-Yin Guo, and Chih-Shuan Wu. "Evaluation of routability-driven macro placement with machine-learning technique." In 2018 7th International Symposium on Next Generation Electronics (ISNE), pp. 1-3. IEEE, 2018.

[11] Hyun, Daijoon, Yuepeng Fan, and Youngsoo Shin. "Accurate wirelength prediction for placement-aware synthesis through machine learning." In 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 324-327. IEEE, 2019.

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

[13] Jain, Arpit, Pabitra Das, and Amit Acharyya. "Artificial Neural Network Based Post-CTS QoR Report Prediction." In 2022 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 682-686. IEEE, 2022.

[14] Barboza, Erick Carvajal, Nishchal Shukla, Yiran Chen, and Jiang Hu. "Machine learning-based pre-routing timing prediction with reduced pessimism." In Proceedings of the 56th Annual Design Automation Conference 2019, pp. 1-6. 2019.

[15] Shalev-Shwartz, Shai, and Shai Ben-David. Understanding machine learning: From theory to algorithms. Cambridge university press, 2014.