DVD Project

Analyzing Delay and Leakage Characteristics Across Technology Nodes: A ML Approach for Predictive Modeling

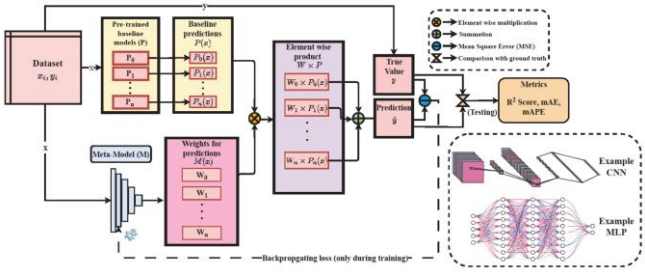
Mayank Shivhare, Ashish Chokhani, Vedant Nipane, Pronoy Patra

Department of Electronics and Communication Engineering,
Student of Engineering

International Institue Of Information Technology, Hyderabad

Hyderabad, india
2021102026, 2021102016, 2021102040, 2021112019
mayank.shivhare, ashish.chokhani, vedant,nipane@students.iiit.ac.in and pronoy.patra@research.iiit.ac.in

***Abstract* — This report delineates the process of generating datasets pivotal for training regression-based machine learning models, serving as a foundational step for Project-2. The dataset generation process entails two primary components: the creation of Process, Voltage, and Temperature (PVT) combinations through sampling from predetermined distributions, and the utilization of these combinations to simulate circuits, with Static Leakage power and Propagation delay serving as target variables.**
Then we do data analysis of the generated dataset followed by the proposed ML model to efficiently estimate the leakage and delay of circuits using the values for the inputs.



Link for Dataset: Click Here

I. Contributions

Ashish: Ashish played a key role in the project by contributing to the generation of netlists, which are essential for defining the circuit structures. Additionally, he was instrumental in developing Python scripts tailored to the specific task of generating datasets focusing on the propagation delay from low to high values. He also invested him time for data analysis so that we can infer great details from it.

Mayank: Mayank's contributions were vital to the project's success. He actively participated in the generation of netlists, which laid the foundation for simulating circuit behavior. Moreover, he dedicated his efforts to developing Python scripts tailored to the generation of datasets with a focus on propagation delay from high to low values. He spent time in trying out and developing the new ML model which would best fit the requisite task.

Pronoy: Pronoy made valuable contributions to the project by specializing in Python scripting for the generation of datasets related to leakage power. His expertise in scripting languages and understanding of circuit simulation processes were instrumental in developing scripts that accurately captured leakage power variations under different conditions. He focused mostly on developing ML models and testing it.

Vedant: Vedant played a pivotal role in the project by focusing on the generation of netlists for all three circuits under study. His expertise in circuit design and simulation enabled him to create accurate and efficient netlists that accurately represented the circuit structures. He also contributed in data analysis.

II. Introduction

This project involves generating datasets from simulations to train regression-based machine learning models. It is a pre-requisite step for Project-2. Dataset generation comprises of two main parts, generating Process, Voltage and Temperature (PVT) combinations by sampling these variables from pre-decided distributions and using these values to simulate circuits.

In electronic design automation (EDA), the ability to predict and optimize circuit performance under varying conditions is indispensable. However, acquiring empirical data to train predictive models necessitates exhaustive

experimentation, often proving time-consuming and resource intensive. Herein lies the significance of simulation-based dataset generation, offering a cost-effective and efficient alternative to empirical methods. By leveraging simulations to emulate real-world scenarios and systematically vary PVT parameters, this approach facilitates the creation of diverse and representative datasets crucial for training machine learning models.

The primary objective of this assignment is twofold: firstly, to generate datasets reflective of real-world circuit behavior under varying PVT conditions, and secondly, to prepare these datasets for subsequent utilization in training regression-based machine learning models. The scope encompasses the generation of PVT combinations, the simulation of circuit behavior using these combinations, and the measurement of Static Leakage power and Propagation delay, which serve as the target variables in the resulting dataset.

Throughout this process, emphasis is placed on measuring the Static Leakage power and Propagation delay of the circuit, which serve as pivotal target variables within the dataset. This introduction provides an overview of the objectives, methods, and significance of the dataset generation process, paving the way for a comprehensive exploration in subsequent sections of this report.

Secondly, we focus on developing ML tools to get a model to predict the values for leakage power and delay with high accuracy.

Once the dataset is generated, the next task would be to do Data Analysis. This will give us immense idea about the correlations among the data. And hence will help us to reduce any possible redundancy.
For that we adopted multiple methods like 3D surface plot, dual axis plots, correlation plots, PCA, GMM etc.

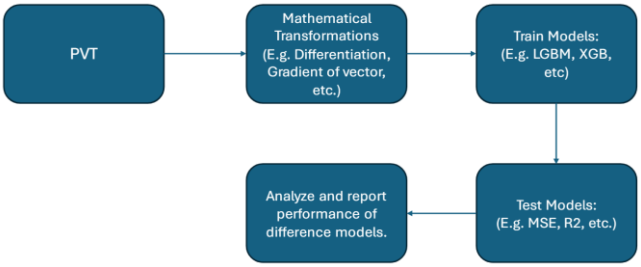


Fig 2.1 Flowchart for the Project

Followed by this, we had an extensive study on what model would fit the purpose. We propose an Ensemble based model to get a good model to predict the delay and leakage values. We also did extensive verifications as well to show that the proposed model works well.

III. INSTRUCTIONS

A. Objective:

To adhere to the prescribed requirements of the assignment, a meticulous approach to dataset generation was undertaken. The following sections detail the methodology employed in generating 10,000 samples using the 45nm-HP (High Performance) PTM file, ensuring the appropriate distribution of Process, Voltage, and Temperature (PVT) values to satisfy Monte-Carlo distribution criteria. The simulation platform utilized for this task was NGSPICE, a versatile tool for electronic circuit simulation.

B. Determining PVT Distributions:

A crucial aspect of dataset generation is the determination of PVT distributions that accurately reflect real-world conditions while encompassing the specified ranges. The temperature range was set from -55 to 125 Celsius, following a uniform distribution to capture a broad spectrum of operating conditions. Nominal voltage was established at 1V, with uniform variations of ±10% to emulate practical voltage fluctuations.

In line with the Monte Carlo distribution principles, process parameters were varied around nominal standards derived from the 45nm-HP PTM file. Specifically, parameters such as toxe, toxm, toxref, tox par, ndep, and xj for both PMOS and NMOS transistors were adjusted with a $\pm 3\sigma$ variation, where σ represents the standard deviation calculated as the mean divided by 30. This methodology ensured a diverse yet realistic representation of process variations.

For delay considerations, cqload variations ranging from 0.01f to 5f were incorporated using a uniform distribution. This wide range of values accounts for different load conditions, capturing variations in circuit performance across diverse scenarios.

C. Simulation setup using NGSPICE

NGSPICE was employed as the simulation platform due to its robust capabilities in circuit simulation and its compatibility with SPICE netlists. The SPICE netlists for the standard gates (NOR2, NAND2, and NOT) were constructed according to the provided specifications. These netlists served as the basis for simulating the circuits under varying PVT conditions.

D. Generating PVT Samples

To generate the required 10,000 samples, PVT values were systematically swept to cover the defined ranges while adhering to the Monte-Carlo distribution. Each sample encompassed a unique combination of Process, Voltage, and Temperature values, ensuring a diverse dataset representative of real-world scenarios.

E. Data Collection and storage

Throughout the simulation process, data on Static Leakage power and Propagation delay of the circuits were collected. These metrics served as the target variables for the dataset.

Leakage power was measured through DC analysis, while delay was assessed via transient analysis, employing PWL signals as inputs.

F. Resulting dataset Structure

The resulting dataset adhered to the specified parameters and distributions, comprising 10,000 samples with associated leakage and delay data. Each sample encapsulated the unique PVT combination along with corresponding circuit performance metrics, facilitating subsequent analysis and machine learning model training.

IV. METHODOLOGY

The implementation methodology outlined below delineates the step-by-step process followed to execute the dataset generation procedure, encompassing the construction of SPICE netlists, generation of PVT distributions, and subsequent simulation to generate leakage and delay datasets.

A. Construction of SPICE Netlists

The initial phase of the implementation involved the construction of SPICE netlists for three standard gates: NOR2, NAND2, and NOT. The netlists were created according to the provided specifications, as depicted in Figures 1, 2, and 3. These netlists served as the basis for simulating the behavior of the standard gates under varying PVT conditions.

B. Generation of PVT Distributions

To generate a comprehensive dataset, 10,000 PVT distributions were systematically generated within the predefined bounds. Each distribution encapsulated unique combinations of Process, Voltage, and Temperature values. It is important to note the cautionary instruction regarding the generation of samples based on input combinations. For instance, if an input gate has four combinations (00, 01, 10, 11), four samples were generated for each PVT value to ensure thorough coverage of input scenarios.

C. SPICE Simulation for Leakage and Delay Dataset Generation

Using the constructed SPICE netlists, the next step involved sweeping across the generated PVT samples to simulate the behaviour of the standard gates under varying conditions. Two key metrics, Static Leakage power, and Propagation delay were measured during the simulation process.

Temperature Dependence
Increase in temperature causes both transconductance, K_{PN} parameter and the threshold voltage, V_{THN} to decrease. Although both decrease with temperature, the former causes a decrease in current while the latter causes an increase in current.

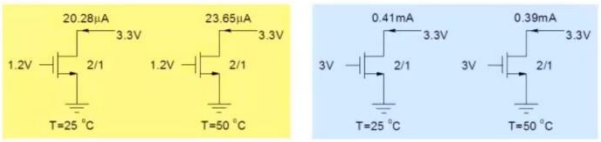


Fig 4.1 Temperature dependence of MOSFETs

- **Leakage Analysis:** For leakage power estimation, cqload was dropped, and DC analysis was performed. This enabled the calculation of Static Leakage power under different PVT conditions.
- **Delay Analysis:** To assess propagation delay, DC inputs were dropped, and transient analysis was conducted using PWL signals as inputs, considering a single PVT at a time. This facilitated the measurement of delay metrics such as delay lh nodea, delay hl nodea, delay hl nodeb, and delay hl nodeb for gates with two inputs.

Best- and Worst-Case Delay Values
- Temperature varying between 0-125^0 C,
- Supply voltage varying between VDD $\{1 \pm 10\%\}$
- Process variation of 3σ

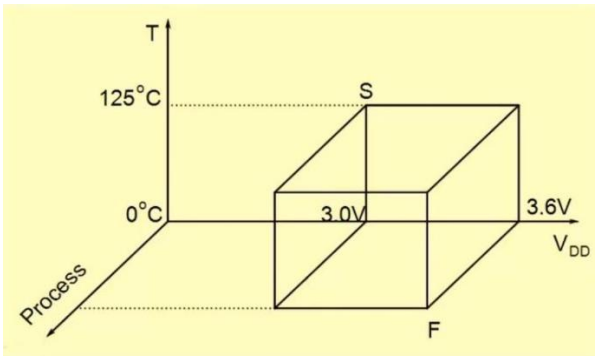


Fig 4.2 Delay variations

Delay Calculation

Using the delay data in the datasheets ($t_{intrinsic}$, K_{load}, and C_{load}) and the delay derating factors, the estimated total propagation delay is

$$t_{TPD} = (K_{Process}) \bullet [1+(K_{Volt} \bullet \Delta Vdd)] \bullet [1+(K_{Temp} \bullet \Delta T)] \bullet t_{typical}$$

$$t_{typical} = t_{intrinsic} + (K_{load} \bullet C_{load})$$

where:

t_{TPD}	=	total propagation delay (ns);
$t_{typical}$	=	delay at typical corner—1.8 V, 25°C, typical process (ns);
$t_{intrinsic}$	=	delay through the cell when there is no output load (ns);
K_{load}	=	load delay multiplier (ns/pF);
C_{load}	=	total output load capacitance (pF);
$K_{Process}$	=	process derating factor, where process is slow, typical, or fast;
K_{Volt}	=	voltage derating factor (/V);
ΔVdd	=	Vdd —1.8 V;
K_{Temp}	=	temperature derating factor (/°C);
ΔT	=	junction temperature — 25°C.

Fig 4.3 Delay calculation formula

D. Dataset Compilation

The resulting dataset comprised 10,000 samples, each representing a unique PVT combination. For leakage

analysis, the dataset contained associated leakage values, while delay analysis yielded delay metrics corresponding to each input combination. These leakage and delay metrics were concatenated into the dataset, resulting in a comprehensive dataset structure like Table 1.

$$10k\ pvt * 2^{no.of\ inputs}$$

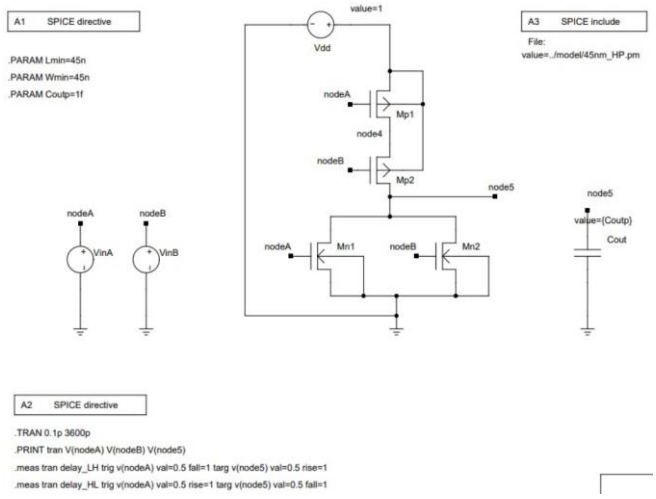


Fig 1. NOR Gate

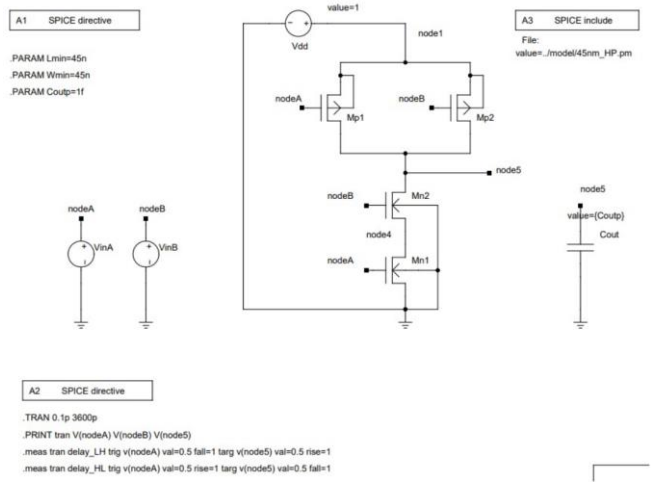


Fig 2. NAND Gate

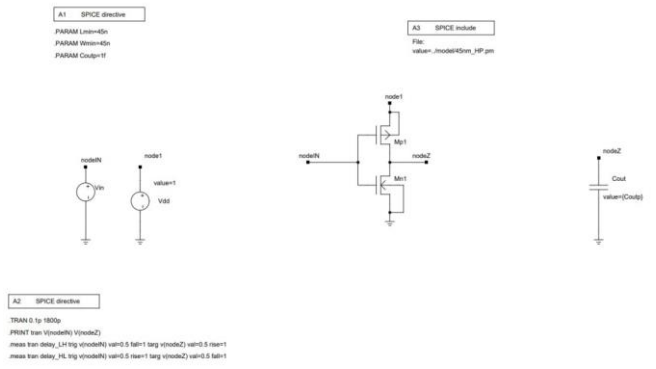


Fig 3. NOT Gate

Vin_A	Vin_B	temp	pvdd	cqload	lmin	wmin	toxe_n	toxm_n	toxref_n	toxe_p	toxm_p	toxref_p	toxp_par	xj_n	xj_p	ndep_n	ndep_p
0	0	120	0.83	2.61E-15	2.15E-08	2.32E-08	8.25E-10	1.07E-09	1.09E-09	1.02E-09	1.07E-09	1.09E-09	1.07E-09	6.99E-09	7.09E-09	5.50E+18	4.38E+18
0	1	120	0.83	2.61E-15	2.15E-08	2.32E-08	8.25E-10	1.07E-09	1.09E-09	1.02E-09	1.07E-09	1.09E-09	1.07E-09	6.99E-09	7.09E-09	5.50E+18	4.38E+18
1	0	120	0.83	2.61E-15	2.15E-08	2.32E-08	8.25E-10	1.07E-09	1.09E-09	1.02E-09	1.07E-09	1.09E-09	1.07E-09	6.99E-09	7.09E-09	5.50E+18	4.38E+18
1	1	120	0.83	2.61E-15	2.15E-08	2.32E-08	8.25E-10	1.07E-09	1.09E-09	1.02E-09	1.07E-09	1.09E-09	1.07E-09	6.99E-09	7.09E-09	5.50E+18	4.38E+18
0	0	-34	0.75	6.17E-16	2.09E-08	2.25E-08	8.10E-10	1.02E-09	1.06E-09	1.02E-09	1.13E-09	1.04E-09	1.11E-09	7.29E-09	7.26E-09	5.48E+18	4.32E+18
0	1	-34	0.75	6.17E-16	2.09E-08	2.25E-08	8.10E-10	1.02E-09	1.06E-09	1.02E-09	1.13E-09	1.04E-09	1.11E-09	7.29E-09	7.26E-09	5.48E+18	4.32E+18
1	0	-34	0.75	6.17E-16	2.09E-08	2.25E-08	8.10E-10	1.02E-09	1.06E-09	1.02E-09	1.13E-09	1.04E-09	1.11E-09	7.29E-09	7.26E-09	5.48E+18	4.32E+18
1	1	-34	0.75	6.17E-16	2.09E-08	2.25E-08	8.10E-10	1.02E-09	1.06E-09	1.02E-09	1.13E-09	1.04E-09	1.11E-09	7.29E-09	7.26E-09	5.48E+18	4.32E+18

Table 1 Example values for PVT distributions

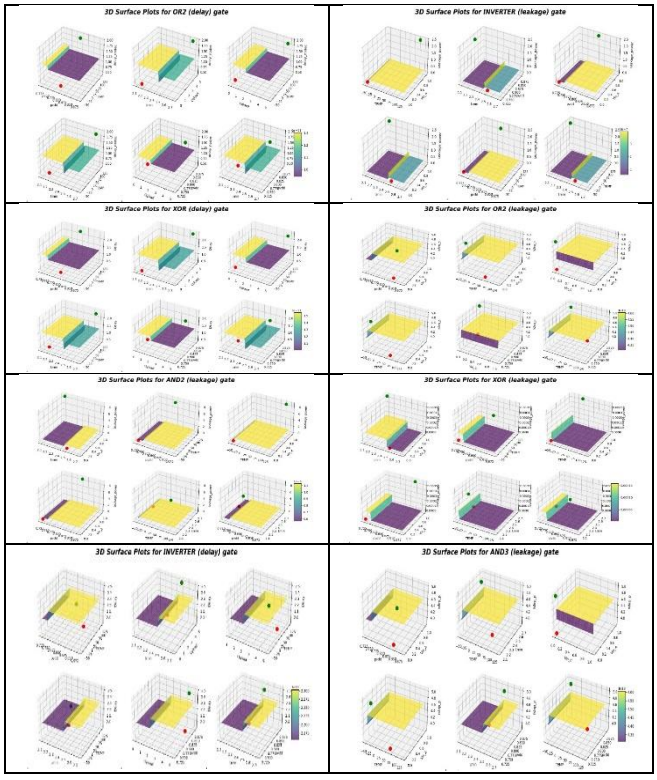
V. DATA ANALYSIS

Once the data is generated from the previous section, the next task is to do data analysis to obtain valuable insights.
Many data analysis techniques were used to better visualize the dataset and gain important insights of the data which can be exploited while using it for training the ML models.

The following data analysis techniques and their corresponding conclusions are mentioned:

A. 3D Surface Plots

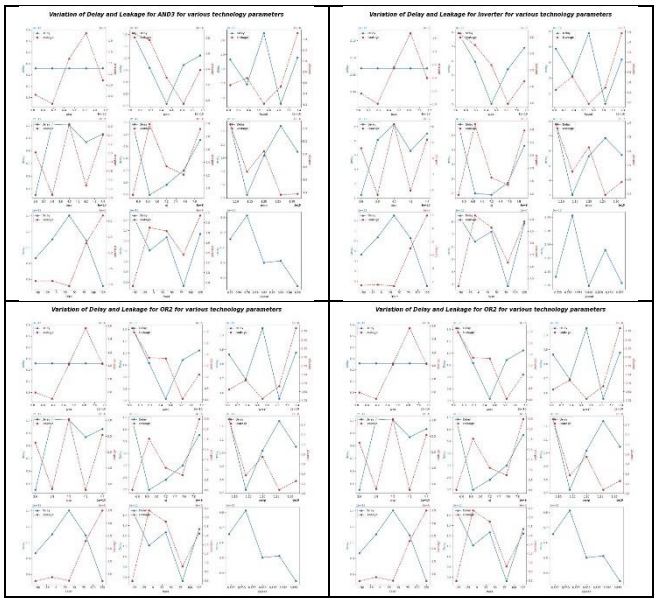
We have used three-dimensional plots, which are graphical representations of data in three-dimensional space. They are useful for visualizing relationships between three variables simultaneously, offering insights that may not be readily apparent in traditional two-dimensional plots.
We have chosen two parameters and showed how leakage and delay varies with it. We also show the minimum and maximum values to get a quick comparison of the parameters affect leakage and delay.



B. Dual Axis Plots

Dual axis plots, also known as secondary axis plots, are graphical representations that involve two separate y-axes sharing a common x-axis. They are used when two sets of

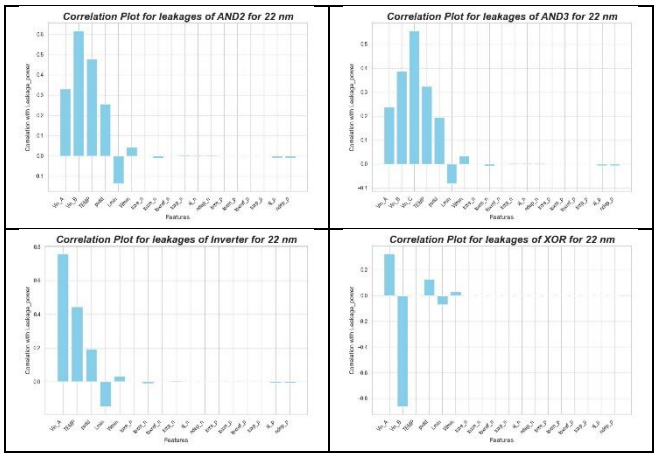
data with different scales need to be compared against the same independent variable.



C. Correlation Plot

A correlation plot, also known as a correlation matrix or heatmap, is a graphical representation used to visualize the correlation between variables in a dataset.

A correlation plot displays the correlation coefficients between pairs of variables in a dataset. The correlation coefficient quantifies the degree of linear relationship between two variables, ranging from -1 to 1. A value close to 1 indicates a strong positive correlation, a value close to -1 indicates a strong negative correlation, and a value close to 0 indicates no linear correlation.



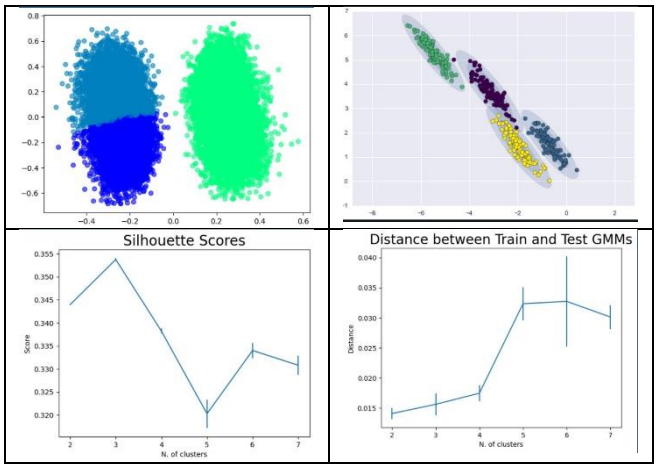
D. GMM

GMM is a probabilistic model that assumes that the data is generated from a mixture of several Gaussian distributions. It's a soft clustering algorithm, meaning that each data point is assigned a probability of belonging to each cluster.

The advantage that GMM clustering holds over k-means is that GMM can handle any type of clusters while k-means works fine for circular clusters only.

Silhouette score is a metric used to evaluate the quality of clustering results. It measures how similar an object is to its own cluster compared to other clusters. The silhouette score ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

We draw clusters with color coding to visualize how the clusters are being made. And find the optimal clusters based on the above metric.



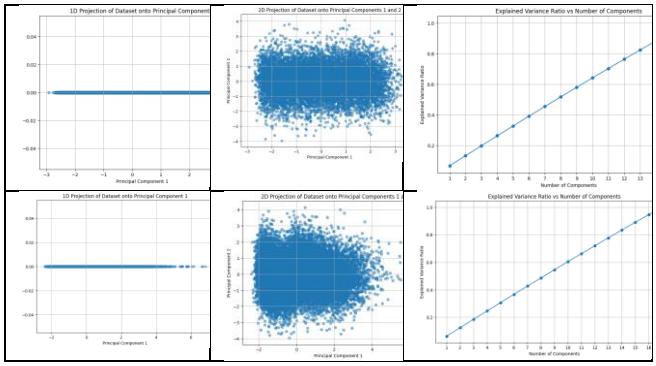
E. PCA

In our data analysis journey, we utilized Principal Component Analysis (PCA) as a powerful tool to uncover hidden patterns and reduce the complexity of our dataset.

It captures the most important information in our data while discarding the less important details. By doing this, PCA helps us visualize our data in a simpler way, making it easier to understand and analyze.

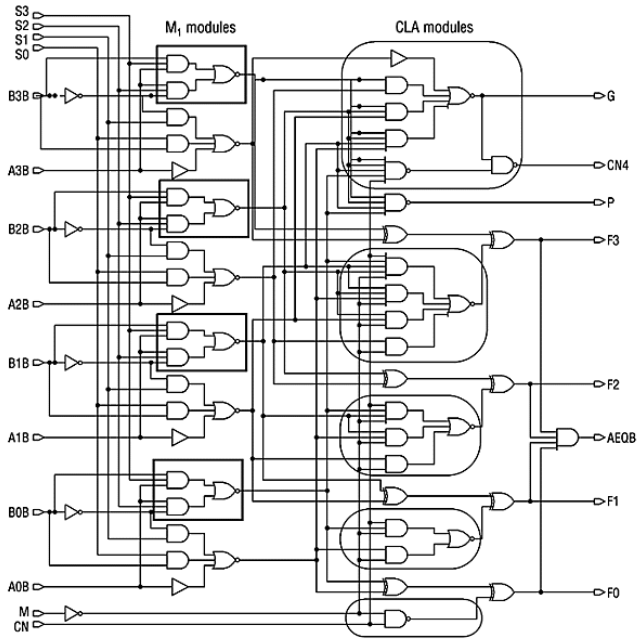
Explained variance quantifies the importance of each principal component in preserving the original information contained within our dataset.

It turned out all the parameters are important. The explained variance increases by 0.1 for every parameter trained equally.



VI. ML Framework

We have used the gates involved in ISCAS-85 C499 circuit for the building of ML models.



ISCAS-85 C499 Circuit Diagram

From the data analysis, we concluded the Ensemble method-based ML Framework.

A. Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is a probabilistic model representing a mixture of a finite number of Gaussian distributions with unknown weights, means and covariances. The Expectation-Maximization (EM) algorithm is used to estimate the parameters in the GMM.

GMM can be effectively used to cluster Gaussian distributions based on the EM algorithm.

- EXPECTATION STEP

The likelihood is defined by:

$$L(\pi,\mu,\Sigma)=\sum_{i=1}^{N} log\left(z_k^i\right)\sum_{R=1}^{k} p(x^i|z=i)p(z=k)$$

By resolving the Eqn., we have the posterior probability for component z_i^k at iteration t + 1 represented as:

$$\gamma_{t+1}(z_k^i)=P_{\pi(t),\mu(t),\Sigma(t)}(z=k|x^i)$$

- MAXIMIZATION STEP

We update our estimate of the mixture weight, mean and covariance of each Gaussian cluster at the iteration, t + 1, with:

$$\pi_k(t+1)=\frac{1}{N}\sum_{i=1}^{N}\gamma_{t+1}(z_k^i)$$

$$\mu_k(t+1)=\frac{\sum_{i=1}^{N} x^i\gamma_{t+1}(z_k^i)}{\sum_{i=1}^{N}\gamma_{t+1}(z_k^i)}$$

$$\Sigma_k(t+1)=\frac{\sum_{i=1}^{N} x^i\gamma_{t+1}(z_k^i)(x^i-\mu_k(t+1))(x^i-\mu_k(t+1))^T}{\sum_{i=1}^{N}\gamma_{t+1}(z_k^i)}$$

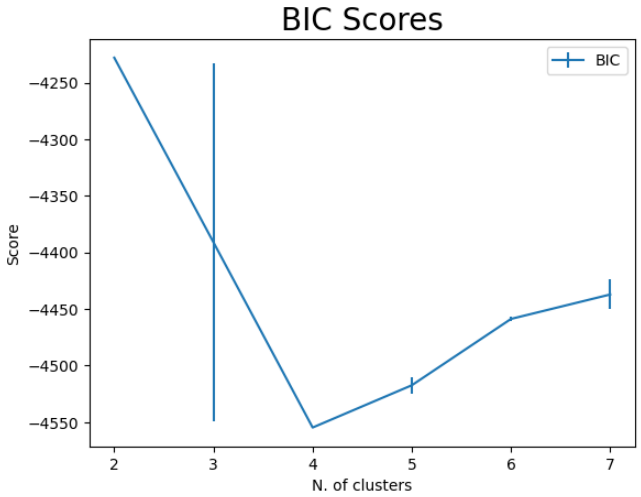
B. Bayesian Information Criterion

The Bayesian information criterion (BIC) is used as an estimate of the Bayes factor for two or more competing models. It is a suitable criterion to choose the optimal number of clusters for GMM. It is defined by:

$$BIC = ln(n)d - \ln(\hat{L})$$

Where, n is the number of data points, d is the number of parameters in the model and $\hat{L}$ is the maximized value for the fitted model likelihood.

When L increases, the BIC score will decrease which implies a better fitted model. The first term of the BIC expression represents the penalty incurred due to over-fitting when too many parameters are in the model.
The GMM model structure with the lowest BIC score is selected for the dataset clustering. In practice, there is no prior knowledge of which cluster a particular test data set would belong to. In order to avoid data leakage problem, the dataset is subject to GMM clustering only after the training and test datasets are split in our proposed flow.



As we can observe that for number of clusters = 4, we have the least BIC, hence the most preferable number of clusters.

C. FEATURE SELECTION AND MODEL SELECTION

We propose a nested 10-fold feature selection cross validation method to avoid the data leakage problem. The whole dataset is split into 90% training and validation dataset and 10% test dataset. The 90% dataset is used for the nested 10-fold feature selection and model selection cross validation

step. During each fold, the training dataset is used for feature selection and model evaluation.

Six popular and diversified regression models are selected for comparison.

1. **Lasso Regression:**

Harnessing the power of L1 regularization, Lasso regression stands out as a stalwart in tackling high-dimensional problems. By selectively shrinking coefficients towards zero, it not only mitigates model complexity but also enhances prediction performance, paving the way for streamlined feature selection and improved interpretability.

2. Support Vector Machine (SVM):

At the forefront of classification tasks, SVM emerges as a formidable contender, leveraging hyperplanes to meticulously carve out boundaries between classes. With the ingenious application of kernel tricks, SVM transcends linear constraints, adeptly navigating through both linear and non-linear problem landscapes, thereby empowering nuanced decision-making.

3. K Nearest Neighbor (KNN):

Simplicity meets efficacy in the realm of K Nearest Neighbor, where classification thrives on the premise of proximity-based similarity measures. Embracing the intuitive notion of nearest neighbors, KNN swiftly categorizes new data points, rendering it an indispensable tool in pattern recognition and beyond, owing to its ease of interpretation and seamless implementation.

4. Random Forest Regressor (RFR):

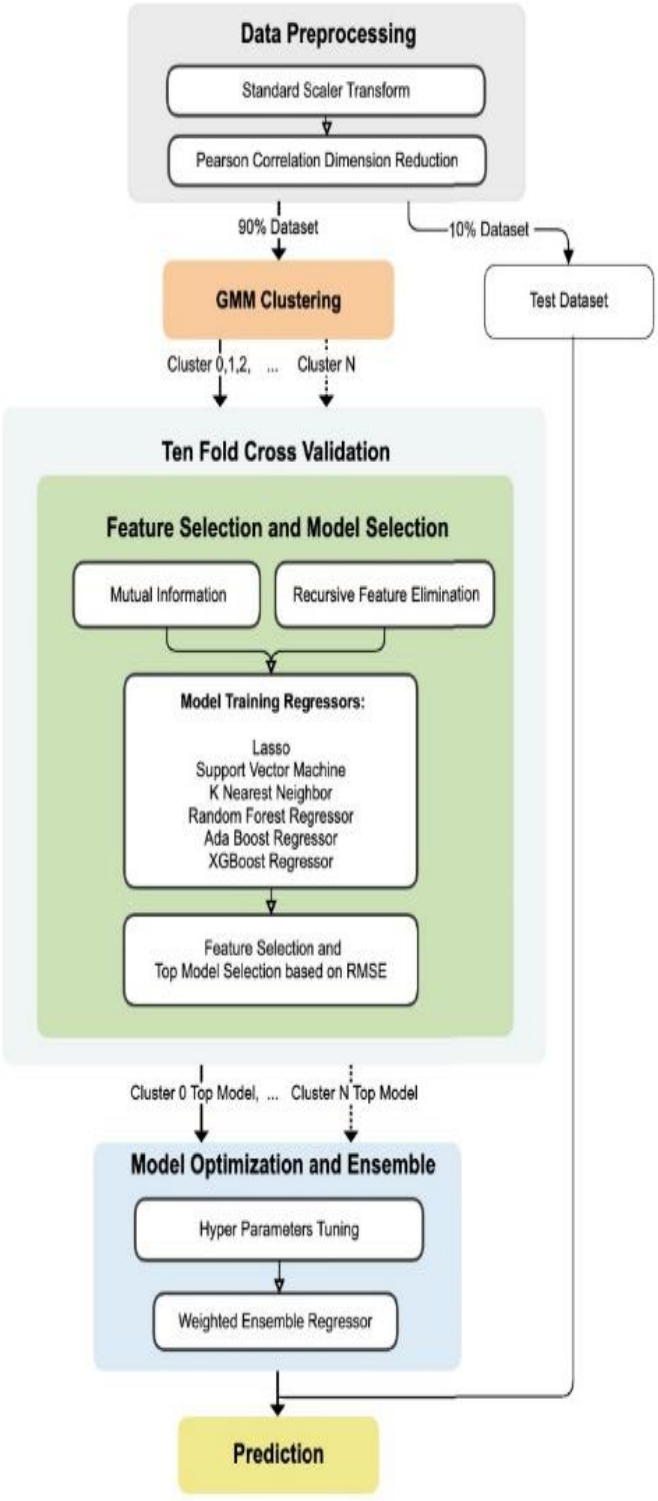
Embodying resilience and efficiency, Random Forest Regressor emerges as a beacon of stability in the regression domain. Embracing the bagging technique and amalgamating diverse decision trees, it orchestrates an ensemble approach, fortifying predictive prowess while safeguarding against overfitting, thereby instilling confidence in predictive outcomes.

5. Adaptive Boosting (ADA):

Embarking on the journey of boosting techniques, Adaptive Boosting ascends to the zenith, acclaimed for its prowess in harnessing the strengths of decision trees as base estimators. Renowned as the epitome of out-of-the-box classification, ADA shines as a beacon of adaptability, galvanizing weak learners into a formidable ensemble, thus elevating predictive accuracy to unprecedented heights.

6. XGBoost (XGB):

Radiating excellence in the landscape of gradient boosting, XGBoost emerges as a paradigm of precision, striking an exquisite balance between variance and bias. Fueled by the gradient boosting framework, it navigates the terrain of decision trees with finesse, offering swift execution speed and optimal variance-bias trade-offs, thereby heralding a new era of predictive modeling excellence.



D. Model Evaluation

The metric for assessing model performance is taken to be the Root Mean Square Error (RMSE). At each fold, the model with the lowest RMSE is selected as the candidate model.

A 10-fold averaged RMSE is used to compare and decide which feature selection method should be used. The mean and standard deviation of the RMSE value is taken into consideration for top model selection. Each cluster is running the cross validation independently. Therefore, the top candidate models can be different for each cluster.

E. Model Optimization and Model Ensemble

After the top model is selected for each cluster, we use the 10-fold Grid Search method for models' hyper parameters' optimization. The Grid Search score is set as the negative mean squared error and 90% of the dataset is used in this step. The next step is to build a weighted ensemble regressor using the top model from each cluster. All the top models' prediction results are combined with different weights to provide a final result. The weight is defined as the percentage of each cluster's data size over the 90% dataset (including both training and validation) size. Following this, the GMM clustering based ensemble regressor is generated. Finally, an unbiased test result is computed using the 10% test dataset, which is untouched from the beginning of the whole process.

RESULTS

The R2 score, also known as the coefficient of determination, is a statistical measure used to evaluate the goodness of fit of a regression model. It indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.

The R2 score is a value between 0 and 1, where:

- 0 indicates that the regression model does not explain any of the variability in the dependent variable.
- 1 indicates that the regression model perfectly explains all the variability in the dependent variable.

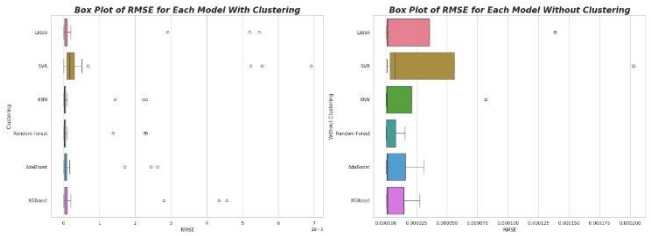
The R2 score obtained by our model is:
R2 score:0.9315904798240491

From the analysis above we can conclude that the optimal No. of GMM clusters for each gate on basis of BIC score:
'INVERTER':4
XOR' : 5
'AND_2' : 5
'AND_3': 10
'OR_2' : 5

For the final part, to verify our model verified well, we created a box plot from the measured mean and variance with and without clustering. The table and plot show the results:

	Model	Mean with Clustering	Std with Clustering	Mean without Clustering	Std without Clustering
0	AdaBoost	2.763856746384353e-06	6.975854876060281e-06	8.70025029625515e-06	1.3528421495915124e-05
1	KNN	2.342359914193569e-06	6.221539485457054e-06	2.0840424624465 3e-05	3.6052948969676665e-05
2	Lasso	5.241987508748968 5e-06	1.4332486885735182e-05	3.535839474333753e-05	6.112859476434277e-05
3	Random Forest	2.329168568967973 6e-06	6.220144202867451e-06	4.309792553558716e-06	6.945643575362161e-06
4	SVR	7.715459772370 79e-06	1.793672234675814 5e-05	5.383501043145 86e-05	8.822923871772546e-05
5	XGBoost	4.597797948448914e-06	1.217600492888991 6e-05	7.962704673746 59e-06	1.210871747382054e-05

Table for mean and variance



Box plot for the 6 regression models used

CONCLUSION

ACKNOWLEDGMENT

The Authors of this paper would like to thank International Institute of Information Technology, Hyderabad for the sponsored access to several scientific websites like IEEE. Thanks are given to Prof. Zia Abbas for providing support and this platform to study and present ideas on this topic. The authors also give thanks to the teaching assistants of the Digital VLSI Design course of the Institute for their help and useful comments on the paper.

REFERENCES

[1] Abbas, Z., & Olivieri, M. (Year). Impact of technology scaling on leakage power in nano-scale bulk CMOS digital standard cells. Microelectronics Journal (2014)

[2] Z. Abbas, V. Genua and M. Olivieri, "A novel logic level calculation model for leakage currents in digital nano-CMOS circuits," 2011 7th Conference on Ph.D. Research in Microelectronics and Electronics, Madonna di Campiglio, Italy, 2011.

[3] Z. Abbas, A. Mastrandrea and M. Olivieri, "A Voltage-Based Leakage Current Calculation Scheme and its Application to Nanoscale MOSFET and FinFET Standard-Cell Designs," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 12, pp. 2549-2560, Dec. 2014

[4] Deepthi Amuru, Andleeb Zahra, Harsha V Vudumula, Pavan K Cherupally, Sushanth R Gurram, Amir Ahmad, Zia Abbas, "AI/ML algorithms and applications in VLSI design and technology"