Introduction

In semiconductor design, tiny variations in process parameters, voltage, and temperature (PVT) affect circuit behaviour. It is crucial to analyse these changes to creating reliable and efficient electronic devices. This project involves performing Exploratory Data Analysis (EDA) on PVT datasets and creating regression models which capture the relations between these parameters and predict leakage and delay values for unseen PVT values.

Here, we generate datasets for the NOT, AND2, AND3, OR2 and XOR2 cells, where

1. The delay datasets contain 30000 samples each, with delays for high to low transitions of each of the inputs.
2. The leakage datasets contain 30000 samples for each input combination to the cells.

These datasets are generated for the 16nm HP, 22nm HP, 22nm MGK, 32nm HP, 32nm MGK, 45nm HP and 45nm MGK technology nodes. The project captures the variations across these technology nodes as well, through proper EDA.

The data is then used to train an innovative meta learning model with various types of base learners, as discussed below.

Dataset Generation

Construction of SPICE Netlists: Netlists for all standard gates are constructed based on provided circuit configurations. These netlists serve as the blueprints for simulating circuit behavior under various PVT conditions.

PVT Variation Matrix Generation:

- The first step in dataset generation involves creating a PVT variation matrix of a desirable size, in this case, 30,000 samples.
- Sample PVT variables from predefined distributions:
    - Temperature: Uniform distribution from -55°C to 125°C
    - Supply Voltage (*pvdd*): Uniform distribution with ±10% variation from nominal
    - Load Capacitance (*cqload*): Uniform distribution from 0.01f to 5f
    - Channel Length (*lmin*) and channel width (*wmin*): Monte Carlo
    - Process parameters (*toxe*, *toxm*, *toxref*, *tox_par*, *xj*, *ndep*): Monte Carlo distribution with mean equal to nominal value from PTM file and standard deviation of mean/30
- Normal distribution with a standard deviation derived from the mean value.
- This method ensures the comprehensive representation of semiconductor fabrication conditions, capturing both the uniform variability and the nuanced process-specific fluctuations inherent in circuit design.

Spice Simulation and Data Collection:

- For each input combination of the standard cells, simulations are conducted using NGSPICE, a netlist simulator using a python script.
- The modified netlists, incorporating the generated PVT samples, are used to simulate circuit behavior.
- During simulation, static leakage power and propagation delay are measured and recorded as output variables.
- These values are extracted from the simulation results and stored in a dataset, along with corresponding input parameters and PVT variations.

Data Processing and Storage:

- The collected data is processed and organized into a structured format suitable for analysis.
- Python scripts are utilized to automate the data processing tasks, including the creation of a Pandas DataFrame and saving the dataset in CSV format.
- The dataset includes input variables such as temperature, voltage, and *cqload*, along with process-specific parameters and target output variables (leakage power and propagation delay).

Implementation

A Python script was developed to automate the dataset generation process. The script performs the following tasks:

1. Reads the nominal values and distribution limits for PVT variables.
2. Defines functions for sampling from uniform and Monte Carlo distributions.
3. Generates a truth table for all possible input combinations based on the number of inputs.
4. For each input combination:
    - Samples 30k PVT combinations.
    - Modifies the SPICE netlist with the sampled PVT values and input combination.
    - Runs NGSPICE simulations to obtain the leakage power and propagation delays.
    - Stores the PVT values, input combination, leakage power, and propagation delays in a data list.
5. Creates a pandas DataFrame from the data list and saves it as a CSV file.

Commands to run the Python script

The script takes two command-line arguments: the **cell name** and the **number of inputs.** The syntax and sample output is shown below:

```
nobhendu@Nobhendus-MacBook-Air leakage % python3 script.py NOT 1
  Inputs: [0]: 100%|████████████████████| 30000/30000 [06:02<00:00, 82.74it/s]
  Inputs: [1]: 100%|████████████████████| 30000/30000 [06:06<00:00, 81.81it/s]
nobhendu@Nobhendus-MacBook-Air leakage % python3 script.py NOR2 2
  Inputs: [0, 0]: 100%|████████████████████| 30000/30000 [06:10<00:00, 80.89it/s]
  Inputs: [0, 1]: 100%|████████████████████| 30000/30000 [06:08<00:00, 81.50it/s]
  Inputs: [1, 0]: 100%|████████████████████| 30000/30000 [06:22<00:00, 78.40it/s]
  Inputs: [1, 1]: 100%|████████████████████| 30000/30000 [06:16<00:00, 79.60it/s]
nobhendu@Nobhendus-MacBook-Air leakage % python3 script.py NAND2 2
  Inputs: [0, 0]: 100%|████████████████████| 30000/30000 [06:12<00:00, 80.44it/s]
  Inputs: [0, 1]: 100%|████████████████████| 30000/30000 [06:23<00:00, 78.31it/s]
  Inputs: [1, 0]: 100%|████████████████████| 30000/30000 [06:31<00:00, 76.60it/s]
  Inputs: [1, 1]: 100%|████████████████████| 30000/30000 [06:29<00:00, 77.06it/s]
```

Dataset Results

The dataset generation process resulted in three CSV files, one for each standard cell. The final dataset looks like a table (30k pvt * 2^{no.of inputs} total samples for leakage and 30k for delay) with additional leakage and delay columns concatenated into it.

The columns in each CSV file are as follows:

- Input columns (e.g., Vin_A, Vin_B for NAND2 and NOR2)
- *temp* (temperature in Celsius)
- *pvdd* (supply voltage)
- *cqload* (load capacitance)
- *lmin*, *wmin* (minimum channel length and width)
- *toxe_n*, *toxm_n*, *toxref_n* (NMOS process parameters)
- *toxe_p*, *toxm_p*, *toxref_p*, *toxp_par* (PMOS process parameters)
- *xj_n*, *xj_p* (junction depth for NMOS and PMOS)
- *ndep_n*, *ndep_p* (doping concentration for NMOS and PMOS)
- *P_leak* (static leakage power) or *delay*

The generated datasets can be used for training regression-based machine learning models in Project-2 to predict the static leakage power and propagation delays for the given standard cells under various PVT conditions.

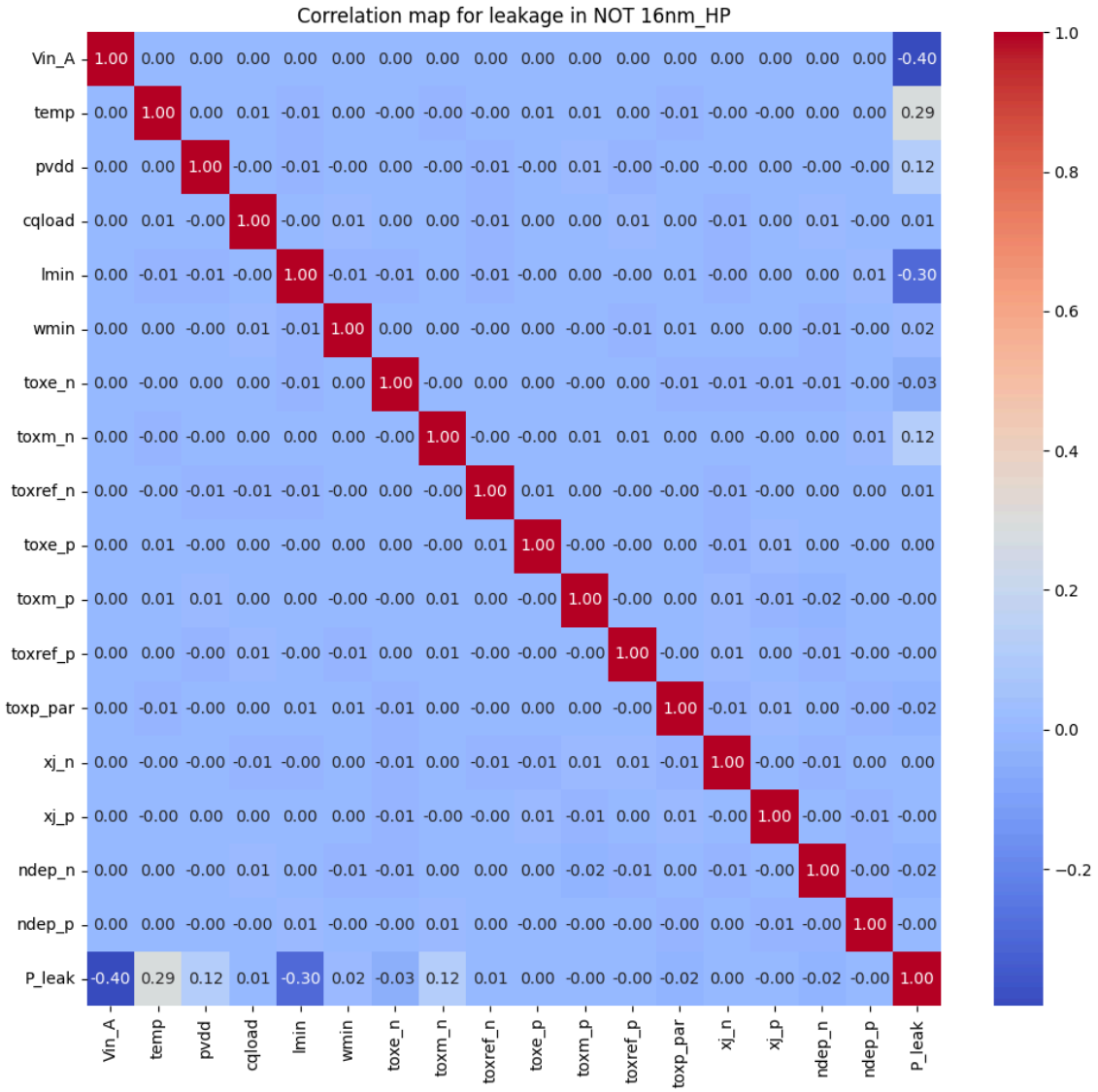
Data Visualization and Analysis

Finding Correlation between outputs and input parameters

We have plotted correlation maps for leakage power and delay with all input parameters.

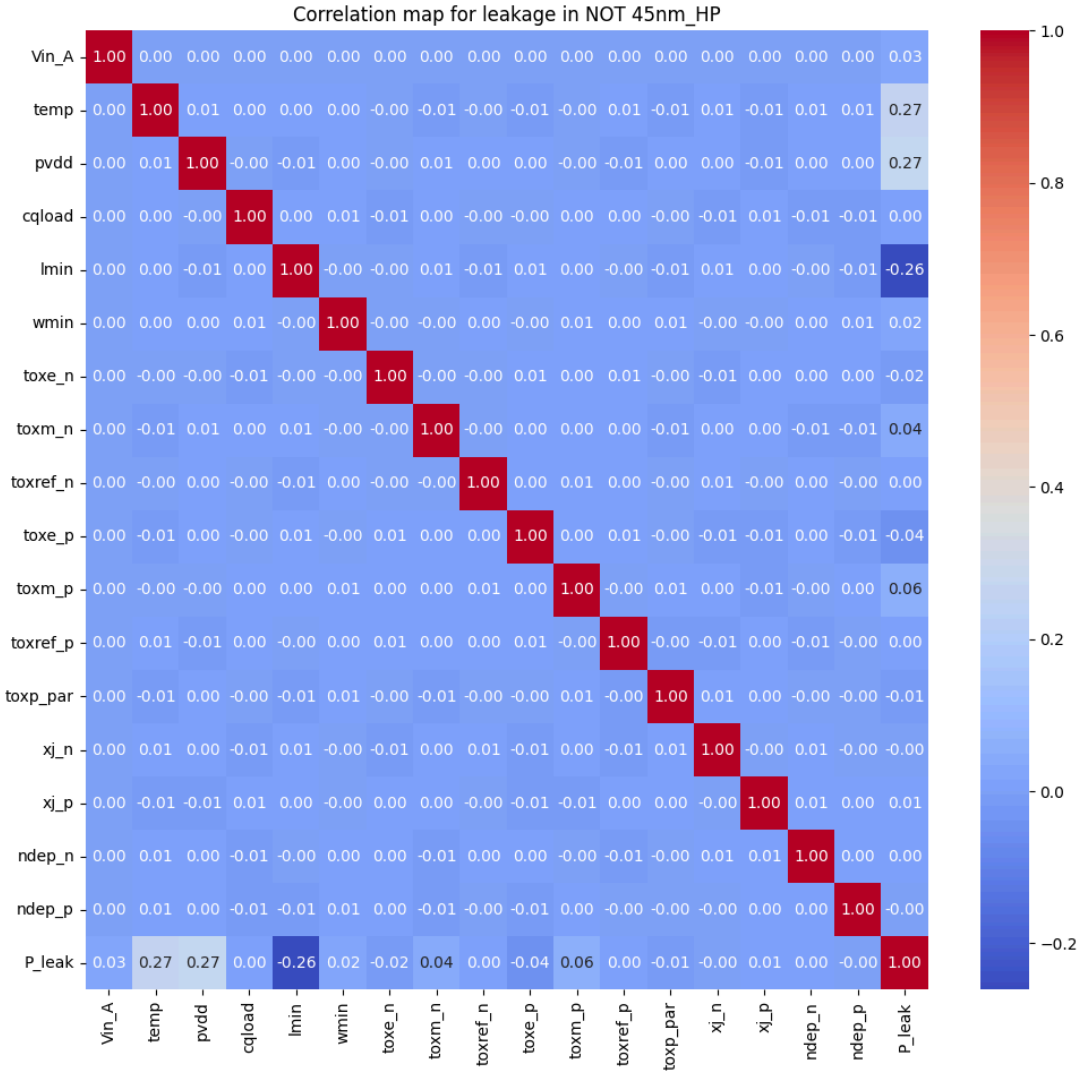
Leakage Correlation Analysis:

Correlation map for leakage in NOT 16nm HP:



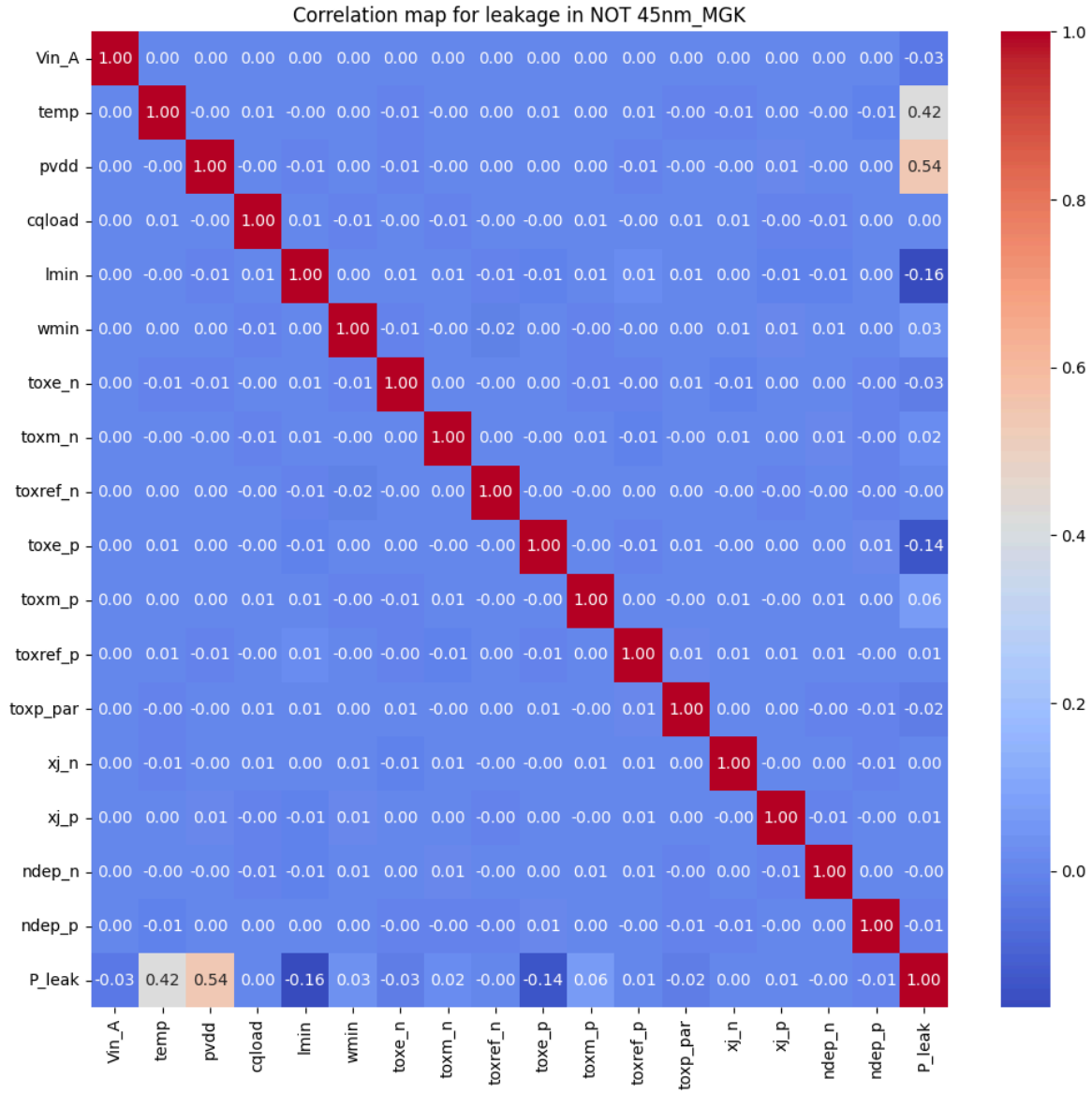
1. Input voltage (Vin_A) shows a negative correlation with leakage power, as expected, since higher input voltages lead to stronger inversion and reduced leakage currents.
2. Temperature (temp) and supply voltage (pvdd) show positive correlations, indicating their significant impact on leakage power in this node.
3. Channel length (lmin) exhibits moderate positive correlations, suggesting their effects on leakage.
4. Other parameters show minimal to no correlation, which suggests that they do not affect leakage.

Correlation map for leakage in NOT 45nm HP:



1. Temperature (temp) and supply voltage (pvdd) still show positive correlations, but the correlation with pvdd is now higher, indicating that it affects leakage more for bigger tech nodes.
2. Channel length (lmin) exhibits a slightly lower, but still moderate positive correlation.
3. Other parameters still show minimal to no correlation.

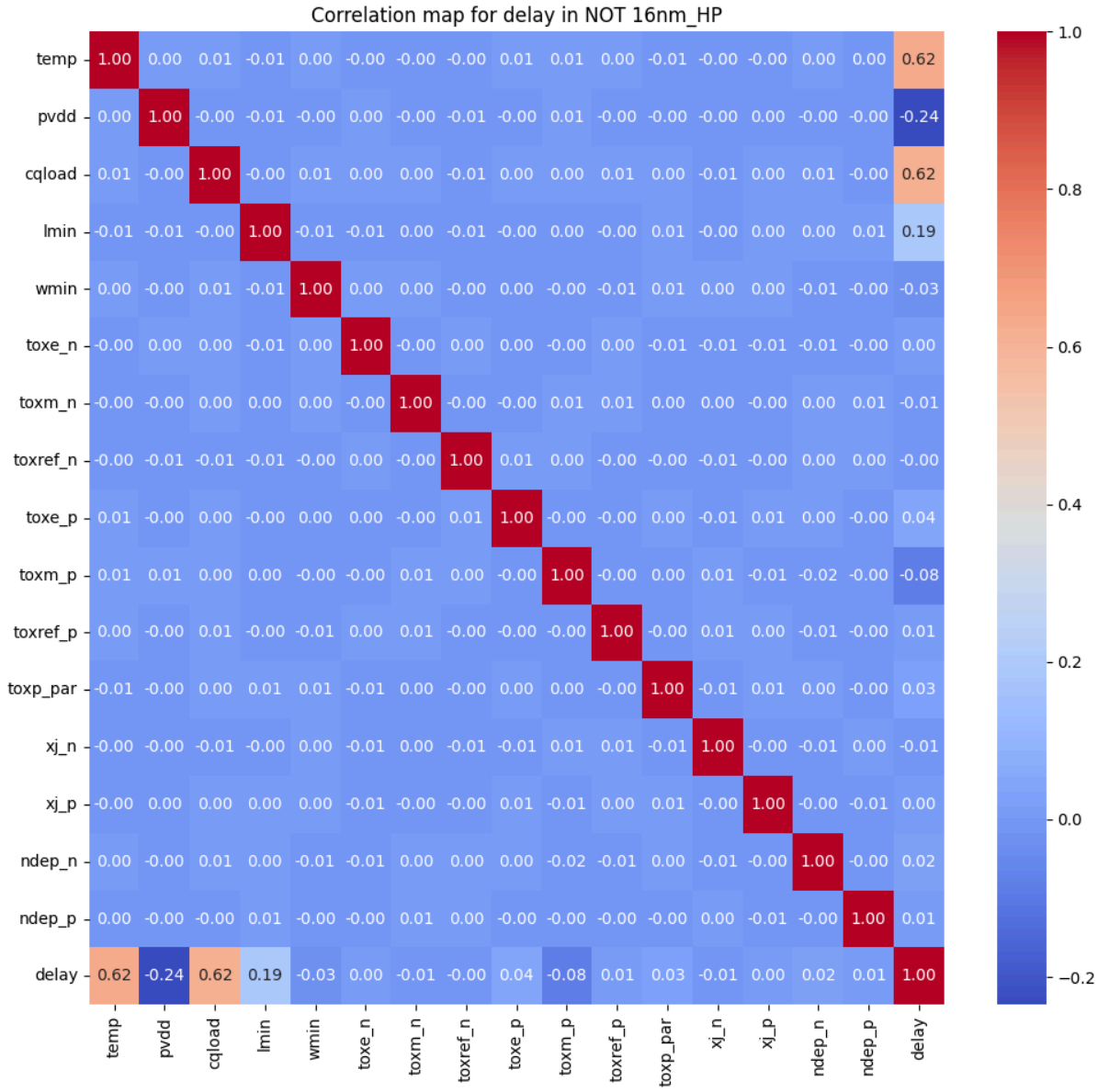
Correlation map for leakage in NOT 45nm MGK:



1. Temperature (temp) and supply voltage (pvdd) show greater positive correlations for MGK than for HP, indicating that they have greater impact on leakage power.
2. Channel length (lmin) exhibits a lower positive correlation.
3. toxe_p exhibits a moderate negative correlation.
4. Other parameters show minimal to no correlation, which suggests that they do not affect leakage.

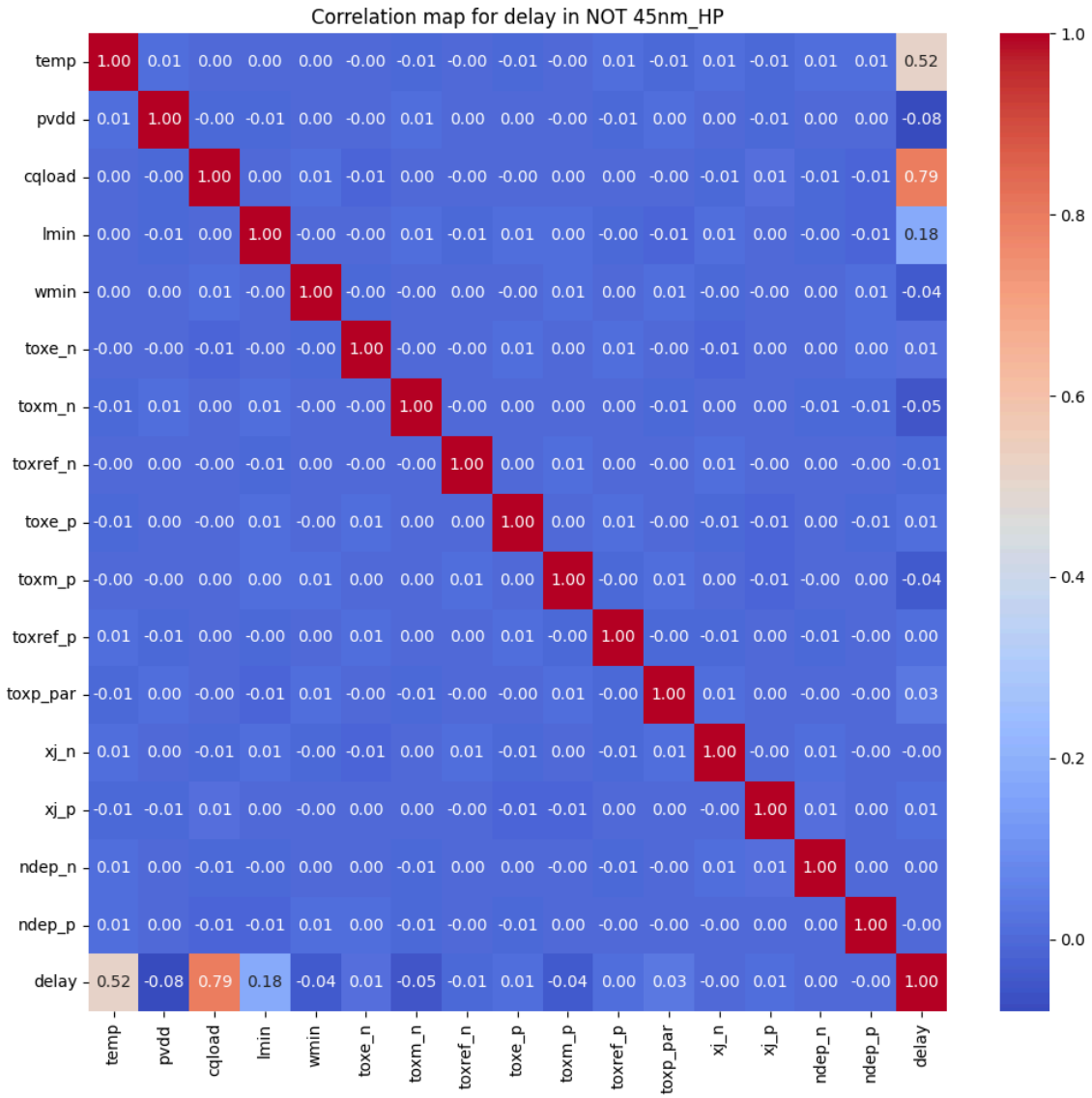
These observations highlight the varying impact of different parameters on leakage power across technology nodes, suggesting the need for node-specific optimizations and trade-offs during circuit design and leakage power management.

Correlation map for delay in NOT 16nm HP:



1. Temperature (temp) and load capacitance (cqload) have strong positive correlation with delay, indicating that higher temperatures and/or higher cqload lead to increased propagation delays.
2. Supply voltage (pvdd) has a moderate negative correlation with delay, suggesting that higher voltages result in lower delays.
3. Among the process parameters, lmin and toxm_p show relatively stronger correlations compared to other parameters.
4. Most other process parameters display negligible correlations with delay.

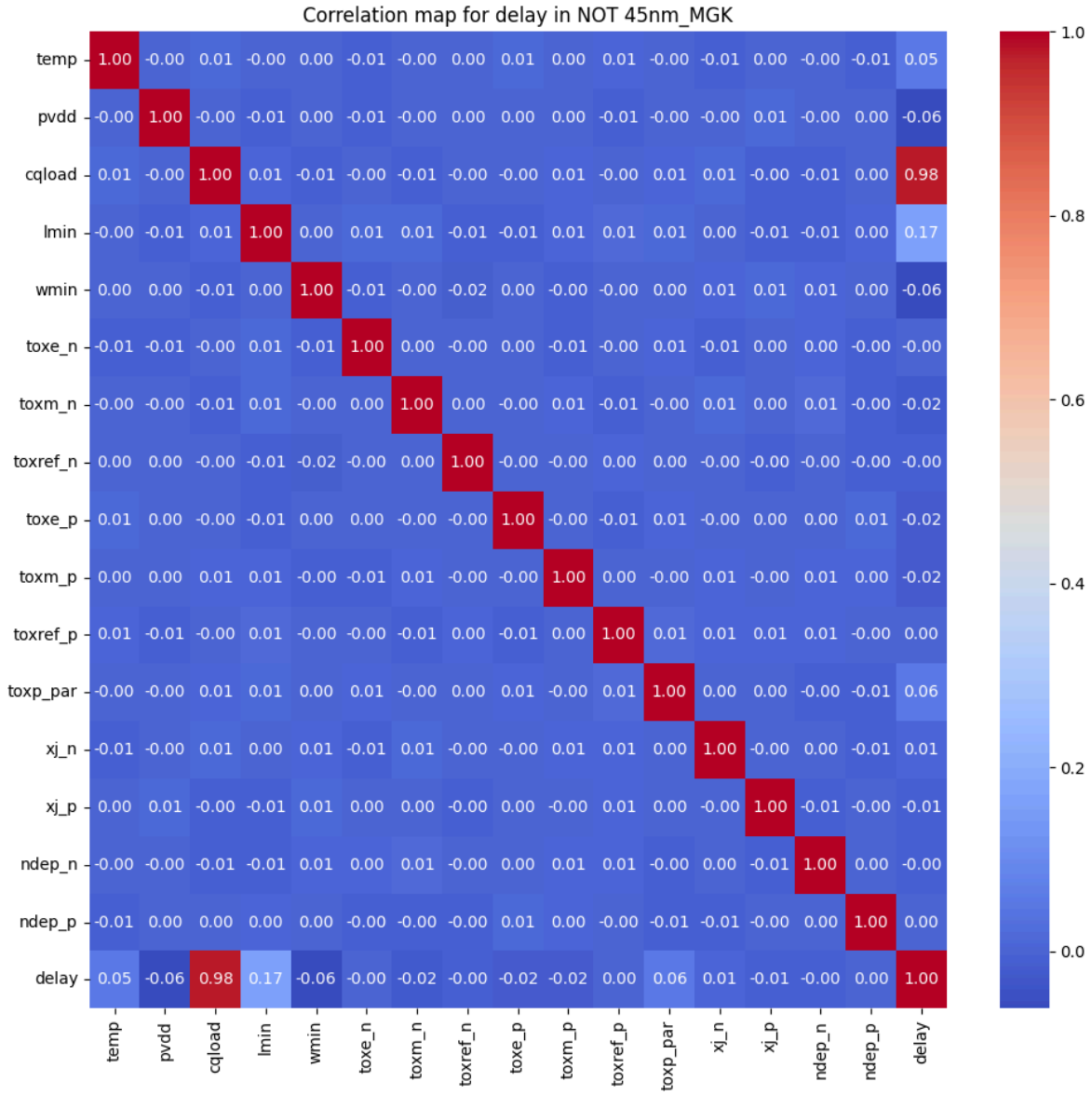
Correlation map for delay in NOT 45nm HP:



For the 45nm HP technology node, the correlation map reveals:

1. Temperature (temp) has a weaker, but still moderate positive correlation with delay.
2. Supply voltage (pvdd) has lesser correlation with delay.
3. Load capacitance (cqload) exhibits a stronger positive correlation with delay.
4. Most other process parameters display negligible correlations with delay.

Correlation map for delay in NOT 45nm MGK:



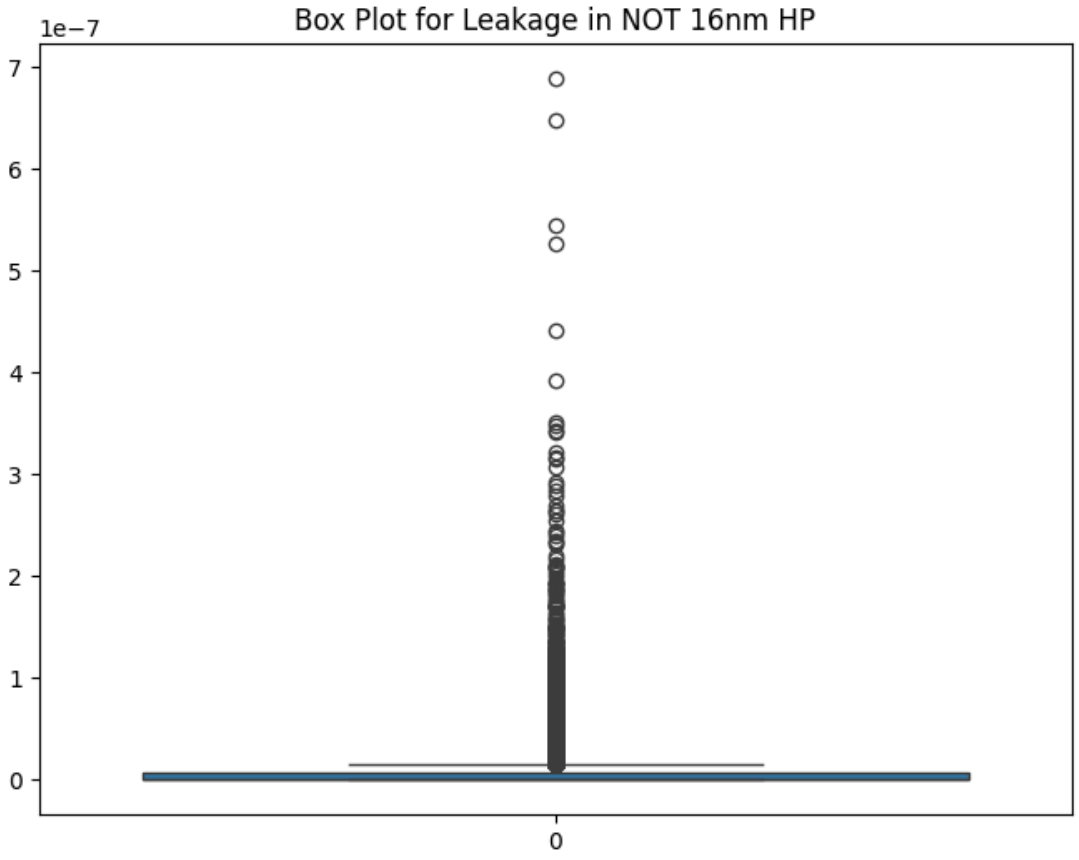
1. Temperature (temp) and supply (pvdd) have almost negligible correlations with delay, as opposed to moderate to strong correlations observed above.
2. Load capacitance (cqload) exhibits a much stronger positive (close to unity) correlation with delay.
3. Most other process parameters display negligible correlations with delay.

The main comparison between HP and MGK is that in HP, delay is affected positively by temperature; this correlation is not observed in the MGK tech node. The MGK tech node, on the other hand, has a very strong correlation with the load capacitance (cqload); this is observed to a slightly weaker extent in the HP tech node.

Box Plot Analysis:

The following box plot and outlier detection methods give insights into the distribution and potential anomalies in the leakage power data for the NOT gate in the 16nm HP technology node.

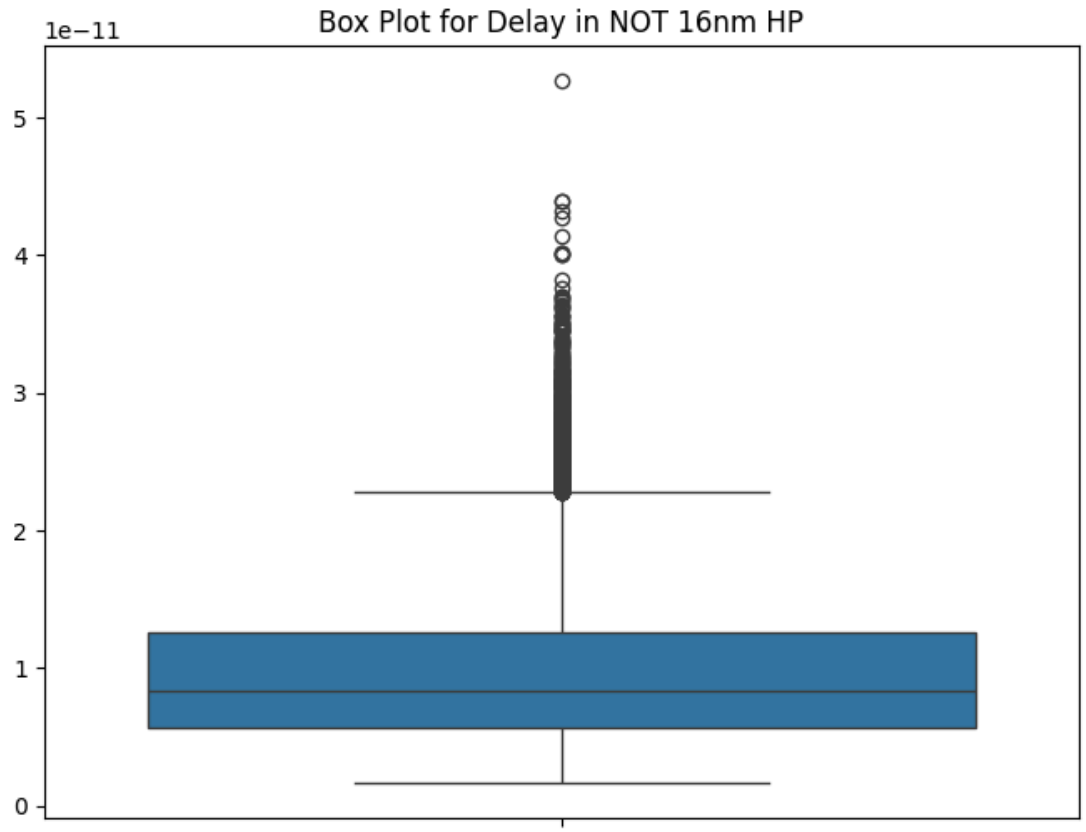
Analysis of the **leakage** box plot:



1. The box plot shows a relatively compact distribution of leakage power values, with the median (horizontal line inside the box) around 2e-7.

2. The interquartile range (IQR), represented by the box's height, is relatively narrow, indicating that most of the data is clustered around the median.
3. However, there are several outliers present, represented by the circles and dots above and below the whiskers (vertical lines extending from the box).

Analysis of the **delay** box plot:



1. **Distribution**: The box plot shows a relatively compact distribution of delay values, with the median (horizontal line inside the box) around 1.5e-11 seconds.
2. **Interquartile Range (IQR)**: The IQR, represented by the box's height, is relatively narrow, indicating that most of the delay values are clustered within a small range around the median.
3. **Whiskers**: The whiskers (vertical lines extending from the box) extend to roughly 1e-11 and 5e-11 seconds, encompassing the majority of the data points.

4. **Outliers**: There are several outliers present, represented by the circles above the upper whisker. These outliers correspond to delay values significantly higher than the typical range.
5. **Skewness**: The distribution appears to be slightly positively skewed, with the upper whisker extending further from the box than the lower whisker. This suggests that there are more instances with higher delays than expected based on a symmetric distribution.
6. **Potential Causes**: The presence of outliers and skewness in the delay data could be attributed to various factors, such as extreme combinations of process, voltage, and temperature (PVT) variations, or potentially erroneous data points.
7. **Impact on Modeling**: Outliers and skewness can potentially impact the performance and robustness of machine learning models trained on this data. It may be necessary to investigate the root causes of the outliers and consider appropriate data preprocessing techniques (e.g., log-transformation, outlier removal, or robust modeling techniques) to mitigate their impact.

Comparison with Leakage: Compared to the leakage box plot, the delay box plot appears to have a more compact distribution with fewer extreme outliers, suggesting that the delay data may be less affected by extreme PVT variations or other sources of variability.

Outlier Detection:

Leakage Outliers:

```
Outliers using Interquartile Range (IQR) method:
(6970,)
```

```
Outliers for using Z-score:
(941,)
```

The significant difference in the number of outliers identified by the two methods suggests that the leakage data distribution is likely skewed or has heavy tails, violating the assumption of normality.

1. The **IQR method** is more robust to skewed distributions and identifies a large number of outliers (6970), indicating that a substantial portion of the data deviates from the central tendency.
2. The **Z-score method**, which assumes a normal distribution, flags fewer outliers (941), potentially missing some of the extreme values that are still within the bounds of a skewed or heavy-tailed distribution.
3. The presence of such a large number of outliers, especially identified by the IQR method, raises concerns about the quality and consistency of the leakage data.
4. Further investigation is needed to determine if these outliers represent legitimate but rare scenarios or if they are indicative of systematic errors or noise in the data collection or simulation process.

Delay Outliers:

```
Outliers (IQR method):
(3833,)
```

```
Outliers (Z-score method):
(1147,)
```

The delay outlier detection results also show a significant difference between the two methods, although not as extreme as in the case of leakage.

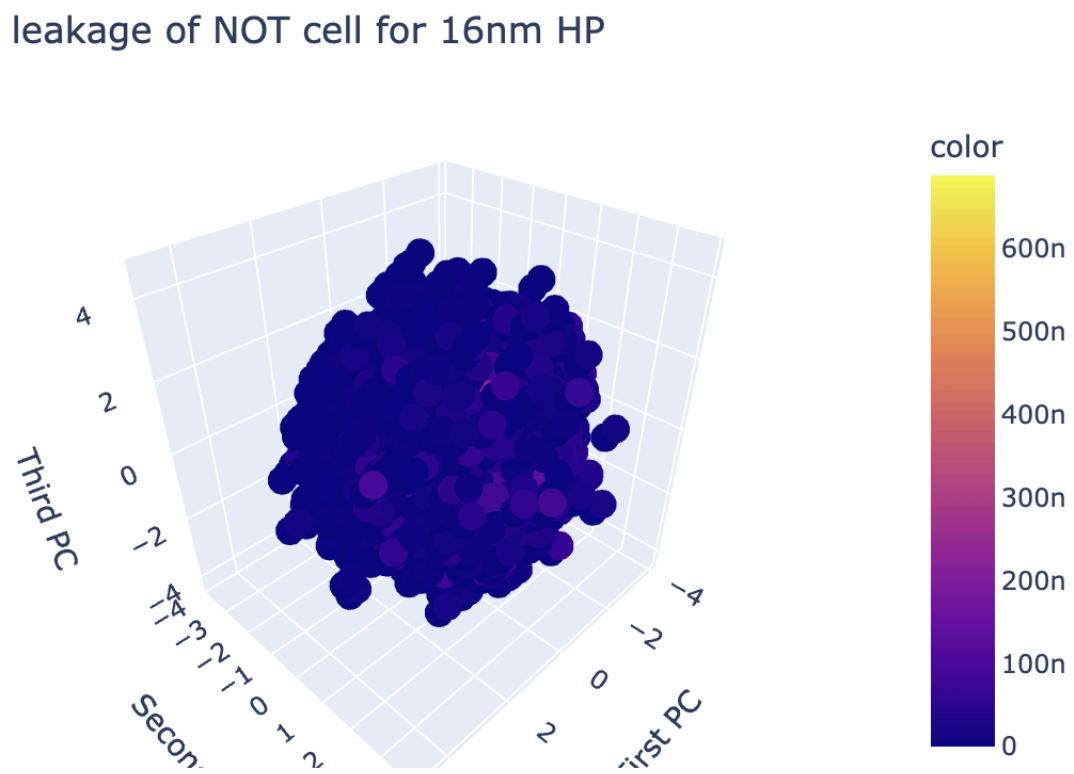
1. The **IQR method** identifies a larger number of outliers (3833), suggesting that the delay data distribution may also be skewed or have heavy tails.
2. The **Z-score method** flags a more moderate number of outliers (1147), potentially missing some extreme values that deviate from the assumed normal distribution.
3. The presence of outliers in the delay data is not unexpected, as delays can be heavily influenced by extreme combinations of process, voltage, and temperature variations.
4. However, the large number of outliers identified by the IQR method warrants further investigation to ensure the data quality and identify potential sources of noise or errors.

Overall, the outlier detection results highlight the importance of careful data preprocessing and handling of outliers in machine learning pipelines. Depending on the underlying distribution and the nature of the outliers, different strategies may be required, such as:

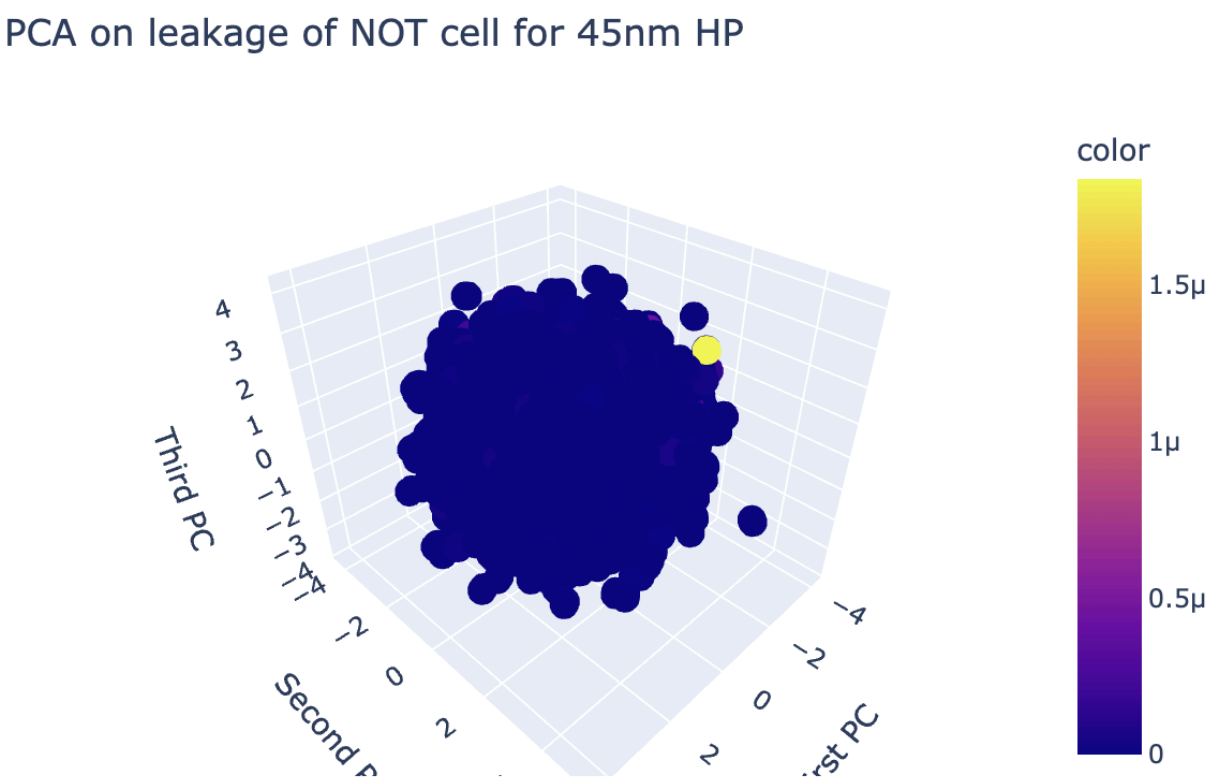
1. Data transformation (e.g., log-transformation) to reduce skewness and bring the distribution closer to normality.
2. Robust statistical techniques or machine learning models that are less sensitive to outliers.
3. Outlier removal or imputation, if the outliers are deemed to be erroneous or not representative of the true data distribution.
4. Investigating and addressing potential issues in the data collection or simulation process that may be contributing to the presence of outliers.

By carefully addressing the outlier issue, the quality and reliability of the data can be improved, potentially leading to better performance and robustness of the machine learning models trained on this data.

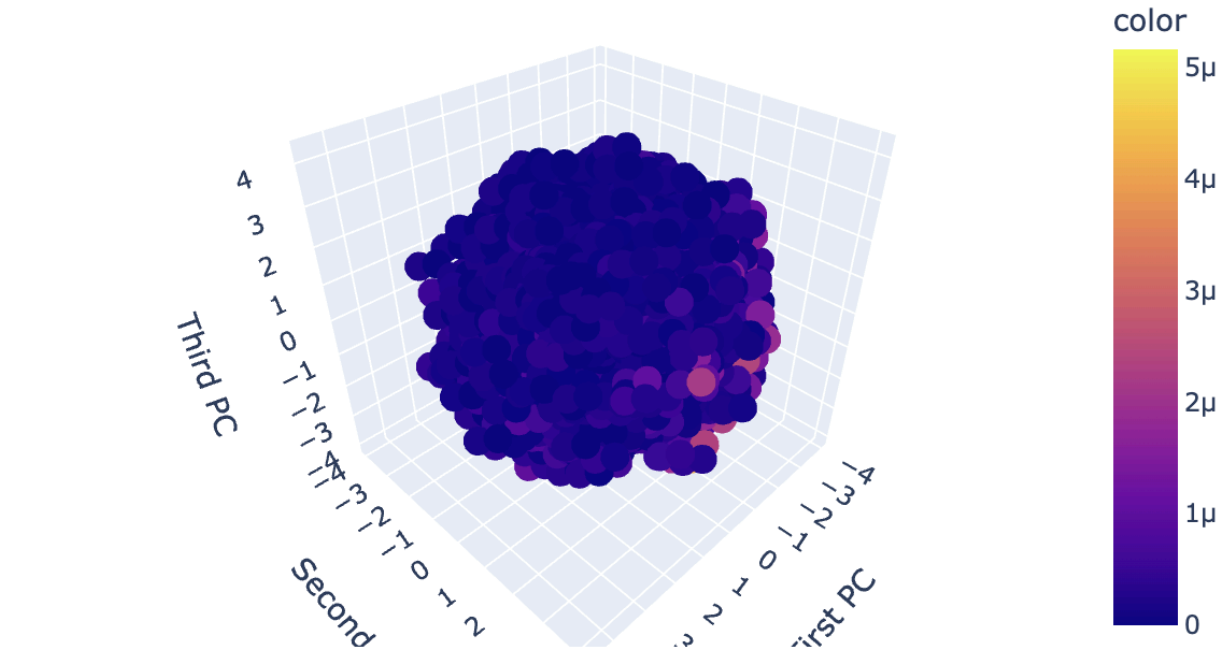
PCA visualization:



This 3D scatter plot shows a compact and dense cluster of data points, indicating that the principal components explain a significant portion of the variance in the leakage data for the 16nm HP technology node. The tight clustering suggests that the leakage values are relatively consistent across the different process, voltage, and temperature (PVT) variations in this advanced technology node.



In this visualization for the 45nm HP node, the data points form a more dispersed cluster compared to the 16nm HP node. This implies that the leakage values have a higher variability across different PVT conditions in the 45nm HP node. However, the cluster is still relatively compact, indicating that the principal components capture a substantial portion of the variance in the leakage data.



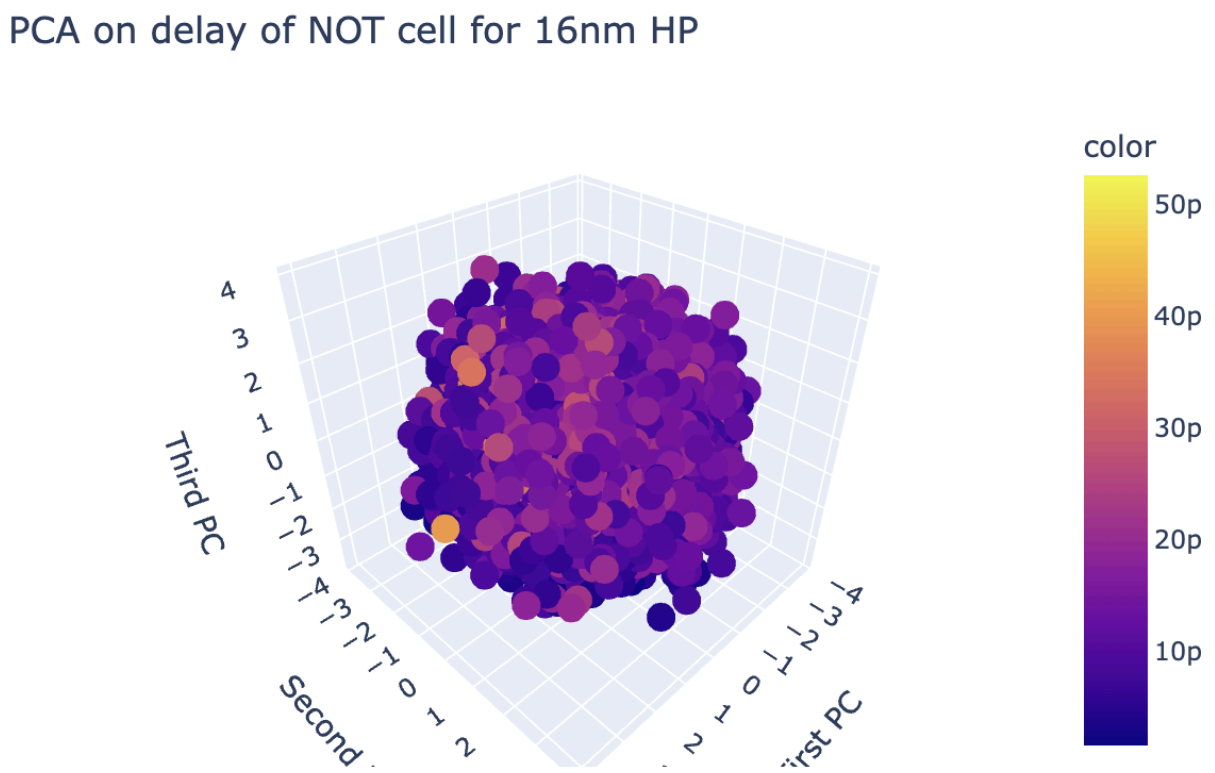
The PCA visualization for the 45nm MGK node shows a significantly more spread-out and elongated cluster of data points compared to the HP nodes. This suggests that the leakage values have a higher degree of variability across different PVT conditions in the 45nm MGK node. The principal components may not be able to capture as much of the variance in the leakage data for this node, indicating the presence of more complex relationships or non-linear effects.

Comparing the HP and MGK nodes: The HP (High Performance) nodes (16nm and 45nm) exhibit more compact and dense clusters in the PCA visualizations, indicating that the leakage values are more consistent and predictable across different PVT conditions. This is expected, as HP nodes are optimized for performance and have tighter control over process variations.

On the other hand, the MGK node (45nm) shows a more dispersed and elongated cluster, suggesting higher variability in the leakage values across different PVT conditions. MGK nodes are designed for

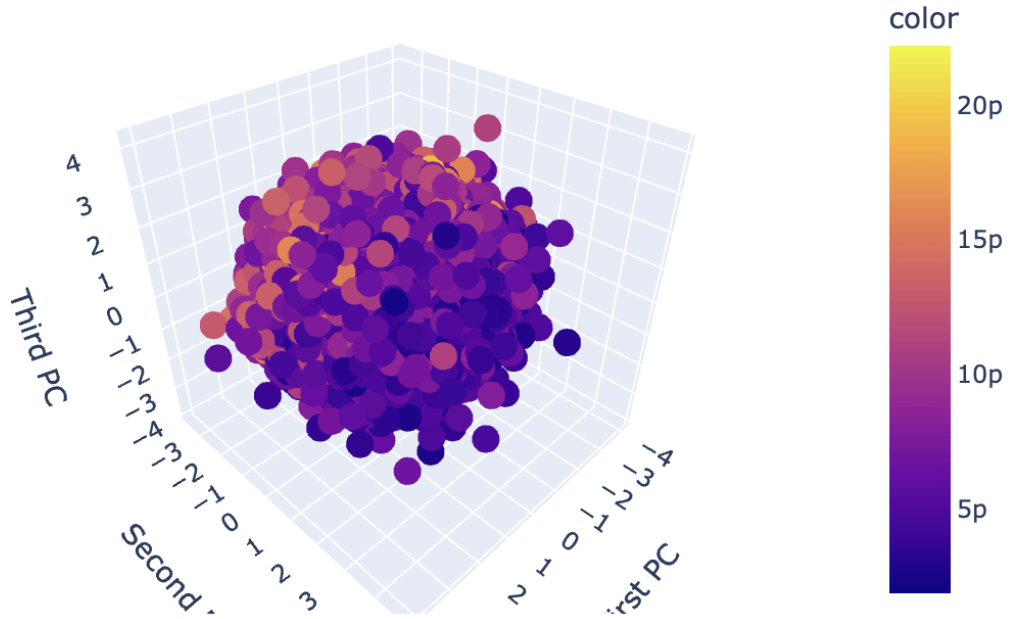
power efficiency and may have a wider range of process variations, leading to higher variability in leakage values.

In general, as the technology node advances from 45nm to 16nm, the leakage values become more consistent and predictable, as evident from the tighter clustering in the PCA visualizations for the 16nm HP node.

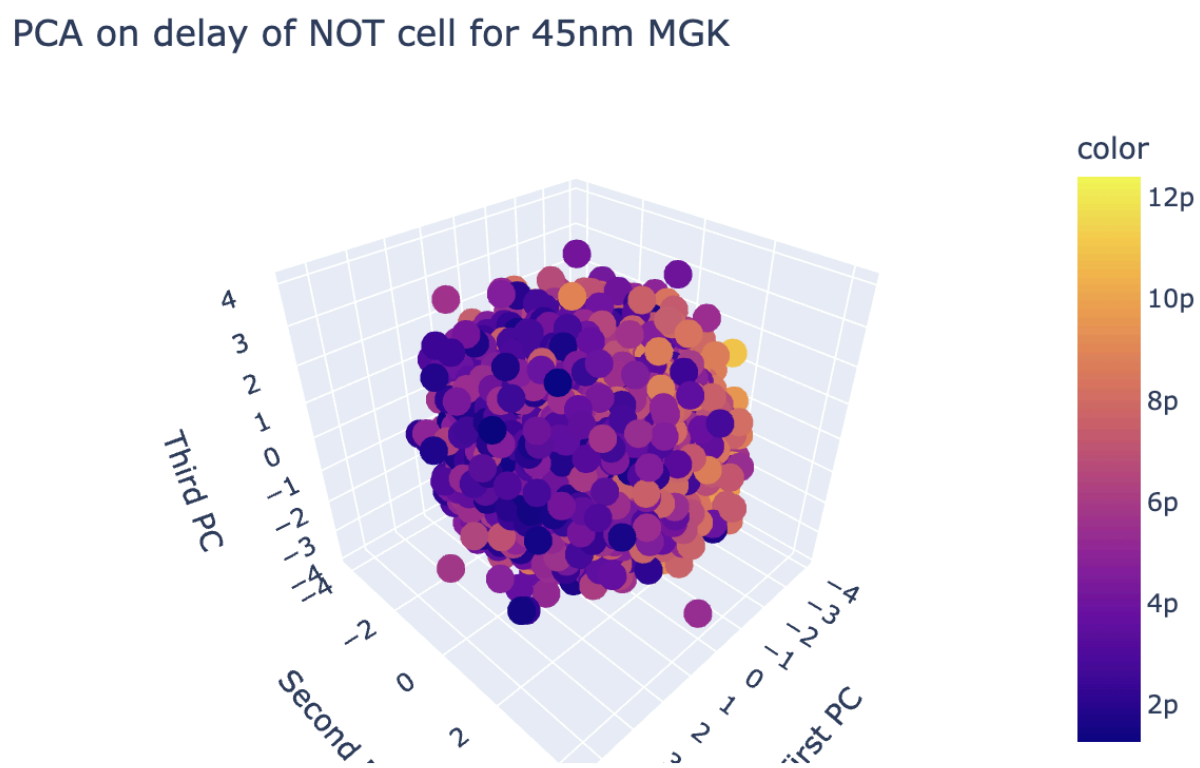


1. The data points form a dense, spherical cluster in the 3D PCA space, indicating a strong correlation among the principal components (likely process variations, temperature, and supply voltage) and their influence on the delay.
2. The delay values range from around 10ps to 50ps, represented by the color gradient from dark red to yellow.
3. The tight clustering suggests that the delay is relatively sensitive to variations in the principal components at this advanced technology node.

PCA on delay of NOT cell for 45nm HP



1. Here, the data points form a slightly more spread-out cluster compared to the 16nm node.
2. The delay range is from approximately 5ps to 20ps, represented by the color scale.
3. The broader distribution indicates that the delay is less sensitive to variations in the principal components at this relatively older 45nm node compared to the more advanced 16nm node.



1. The data points form a dense cluster, similar to the 16nm HP node, but with a smaller delay range of around 2ps to 12ps.
2. Compared to the 45nm HP node, the 45nm MGK node exhibits a tighter clustering, suggesting higher sensitivity to variations in the principal components.

Comparing HP and MGK nodes at 45nm:

- The 45nm HP node shows a broader distribution of delay values, indicating less sensitivity to variations.
- In contrast, the 45nm MGK node exhibits a denser clustering, implying higher sensitivity to variations, likely due to the multi-gate architecture.

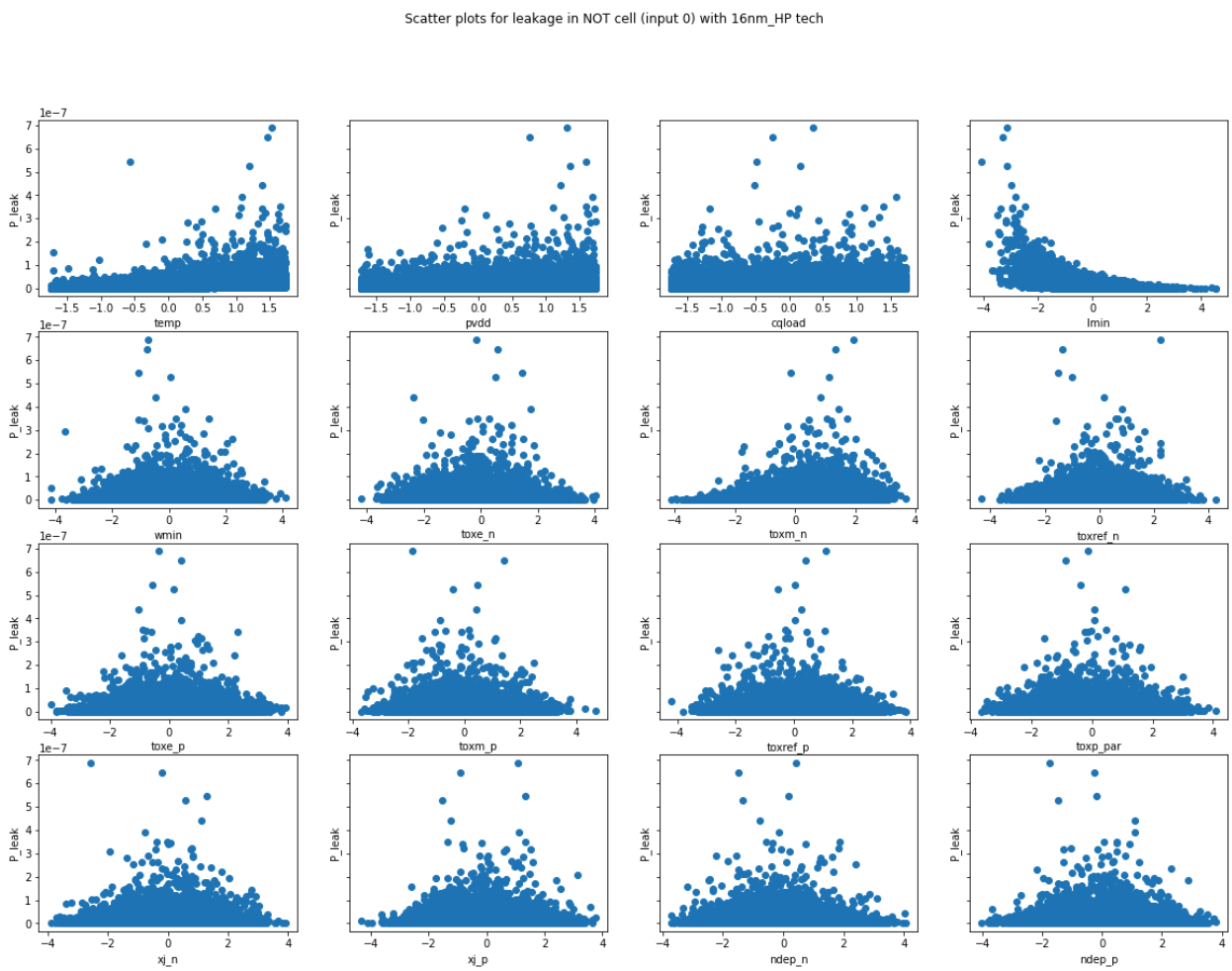
Comparing across technology nodes:

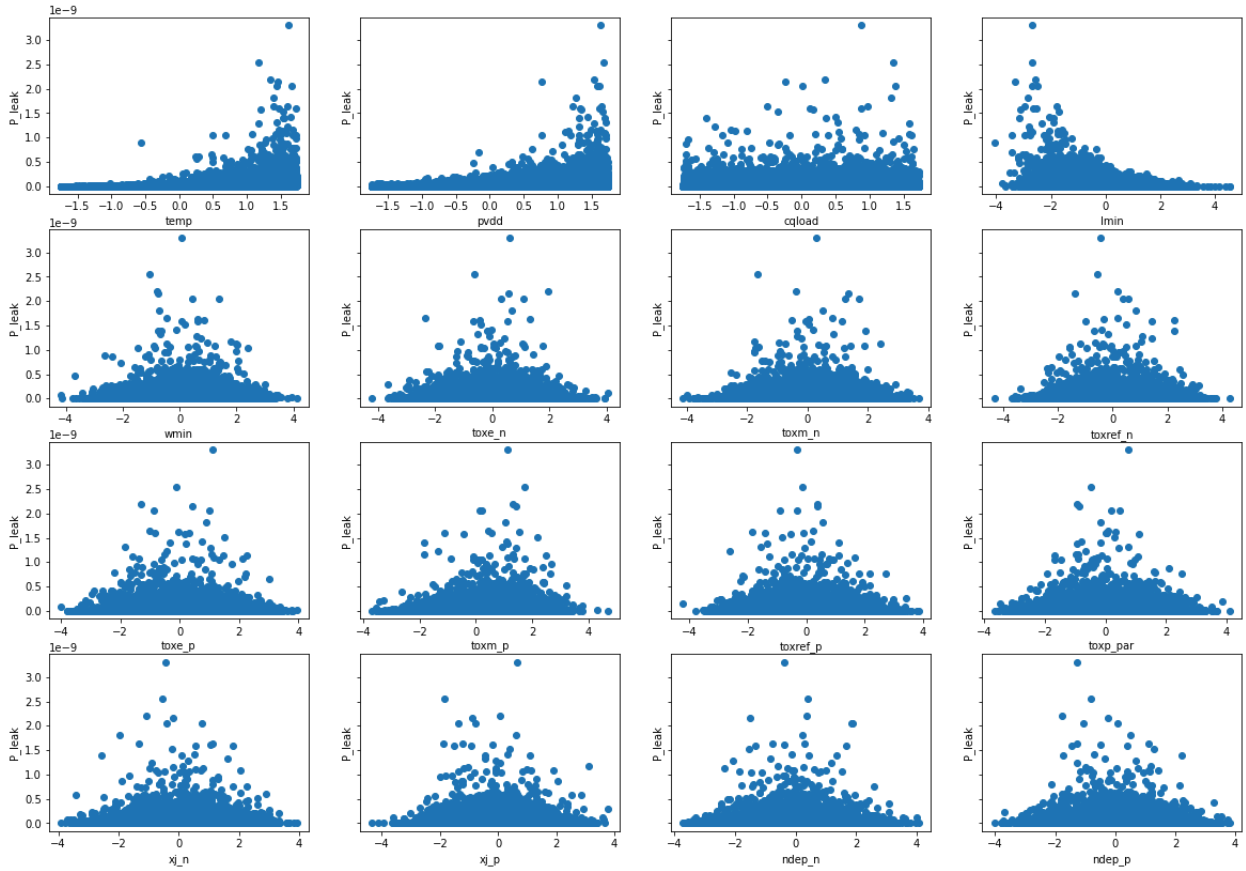
- The most advanced 16nm HP node shows the highest delay values and a dense clustering, indicating higher sensitivity to variations.
- The 45nm HP node exhibits a more spread-out distribution and lower delay values compared to the 16nm node, suggesting lower sensitivity to variations.
- The 45nm MGK node falls in between the 16nm HP and 45nm HP nodes in terms of delay range and clustering density, indicating moderate sensitivity to variations.

NOTE: PCA is not informative.

Scatter Plots for leakage:

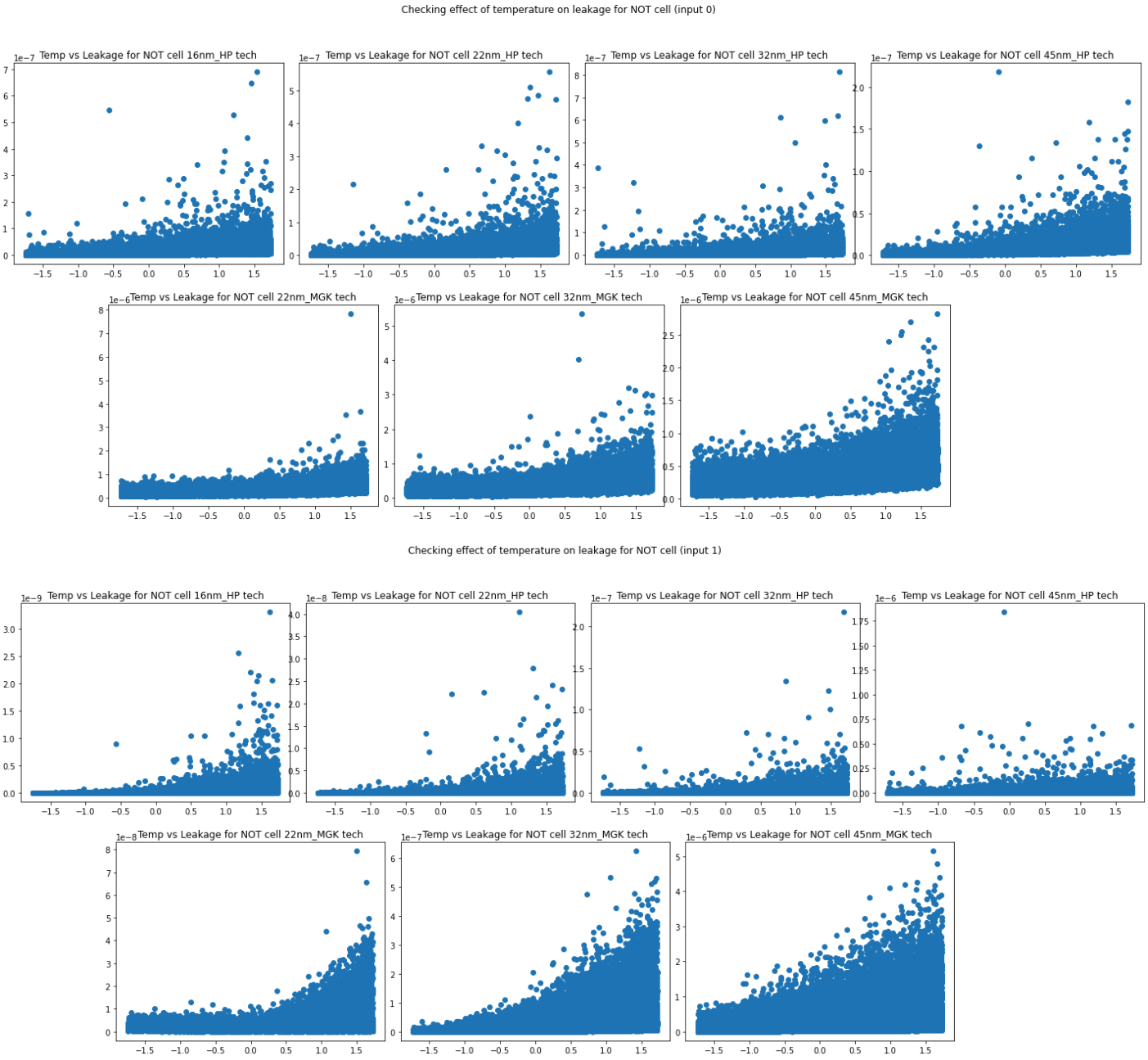
Scatter plots for leakage in NOT cell with 16nm HP tech:





1. We can see that our observations from the correlation maps agree with the general trends exhibited in the scatter plots above.

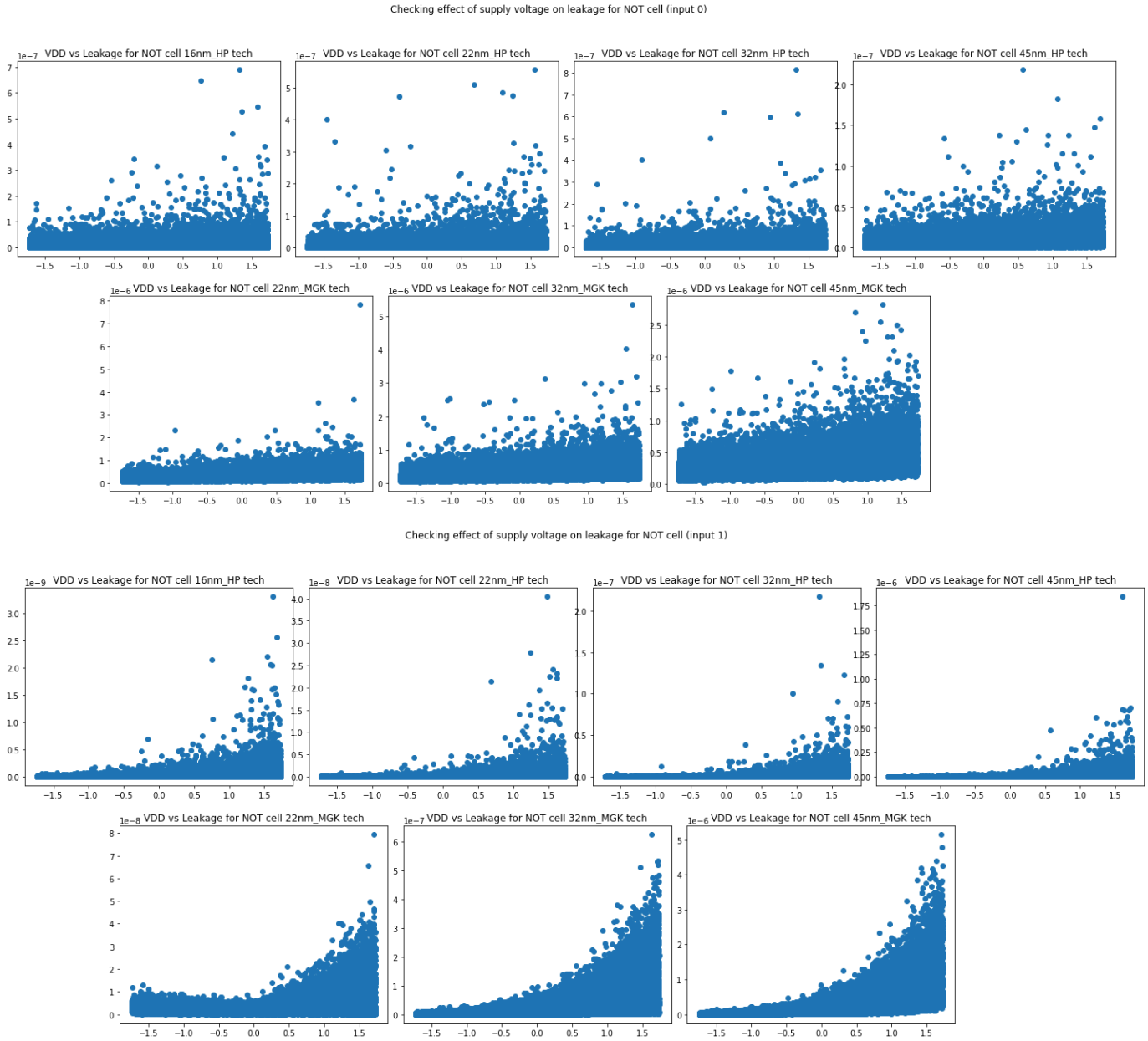
Temperature vs. Leakage for NOT cell across all tech nodes:



1. The leakage power increases exponentially with temperature across all technology nodes, which is an expected behavior due to the temperature dependence of carrier concentrations and diffusion mechanisms.

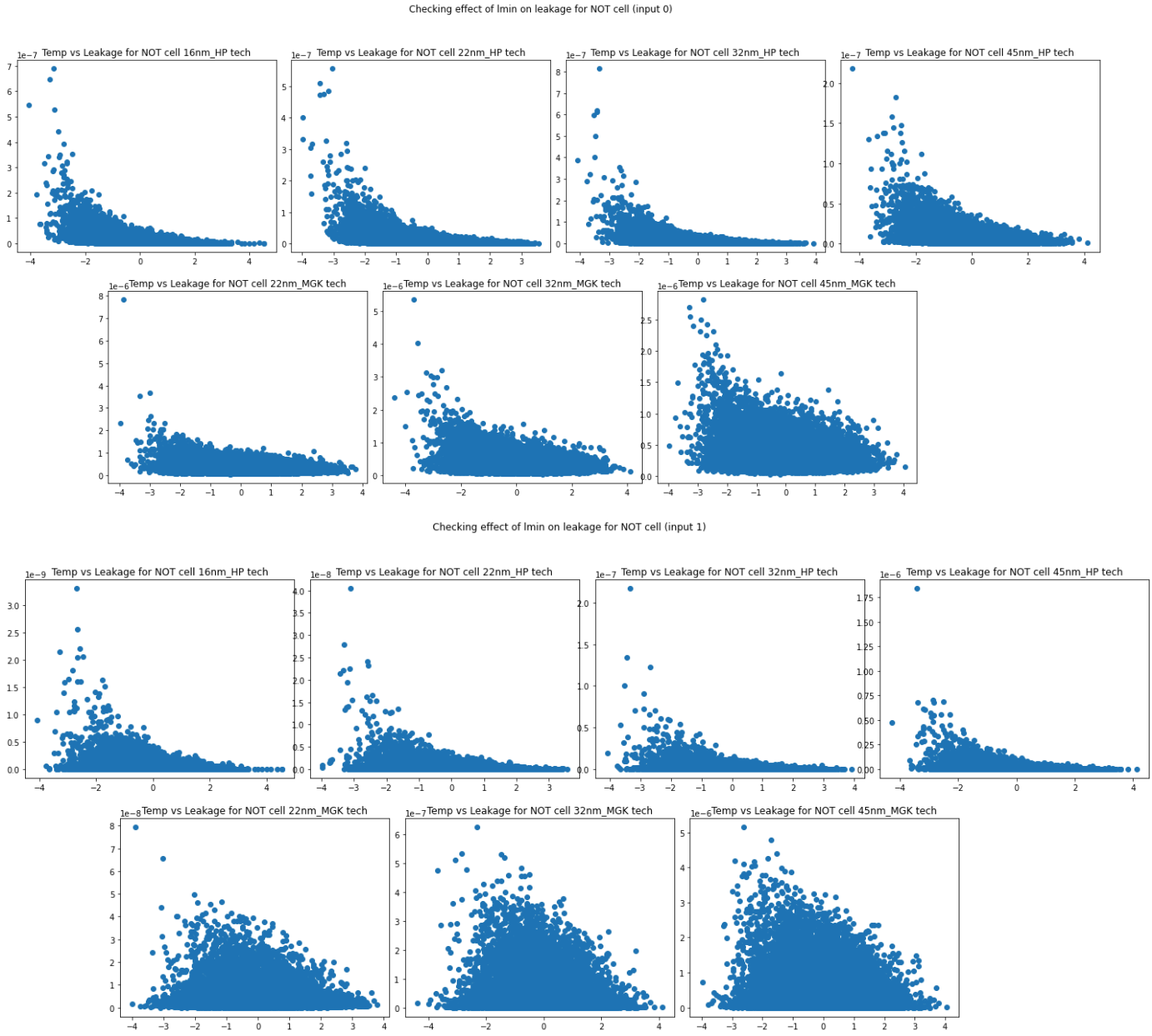
2. The spread or distribution of leakage values is wider at higher temperatures, indicating a stronger influence of process variations at elevated temperatures.
3. The overall leakage power levels decrease with more advanced technology nodes (e.g., 16nm HP has lower leakage than 45nm HP), which is a consequence of scaling and improved process control.
4. For the same technology node, MGK tends to exhibit higher leakage and also greater correlation with temperature, as compared to HP.

Effect of Supply Voltage (VDD) on Leakage across all tech nodes:



1. As the supply voltage (VDD) increases, the leakage current tends to increase across all technology nodes (16nm, 22nm, 32nm, 45nm) and models (HP and MGK), though this dependence is seen to a greater extent in the case of MGK.
2. This dependence is also shown to a much greater extent for input 1 than for input 0.

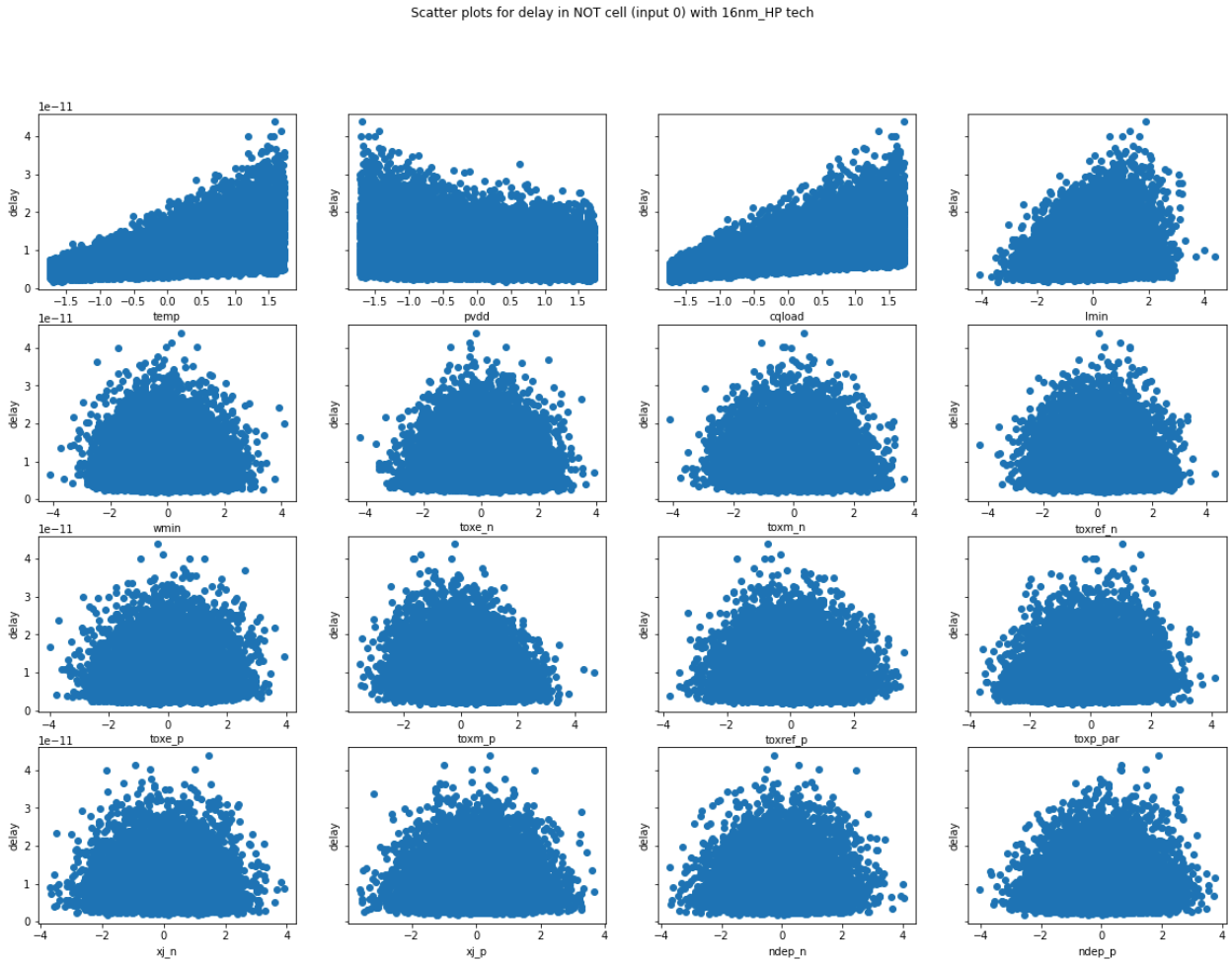
Effect of Minimum Channel Length (lmin) on Leakage across all tech nodes:

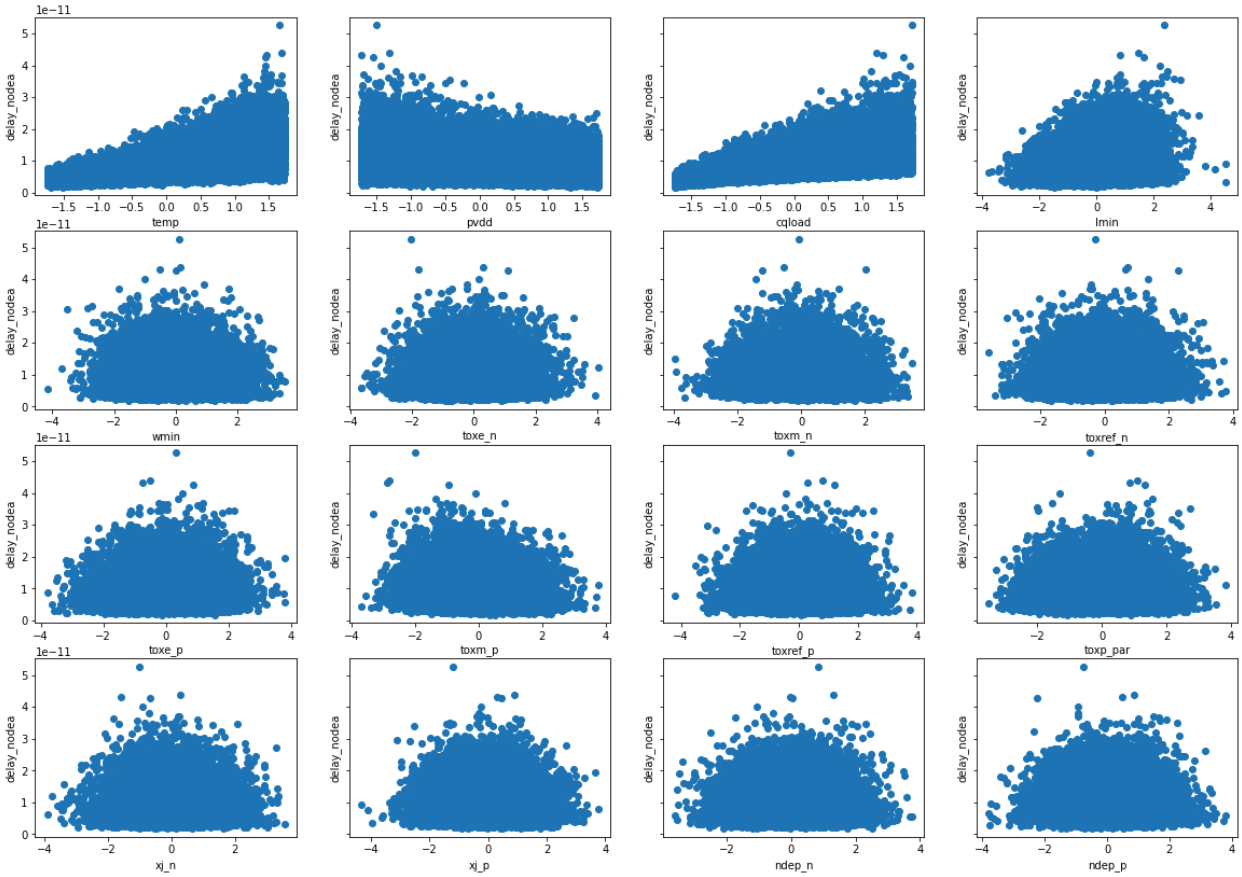


1. The leakage current shows a decreasing trend with increasing channel length (lmin) across all technology nodes.
2. The MGK model exhibits higher leakage values compared to the HP model for the same technology node, consistent with the observations from the VDD plots.
3. The decreasing dependence (for MGK) is visible for input 0 while disappearing for input 1, suggesting that minimum channel length does not affect leakage for transistors in the ON state.

Scatter Plots for delay:

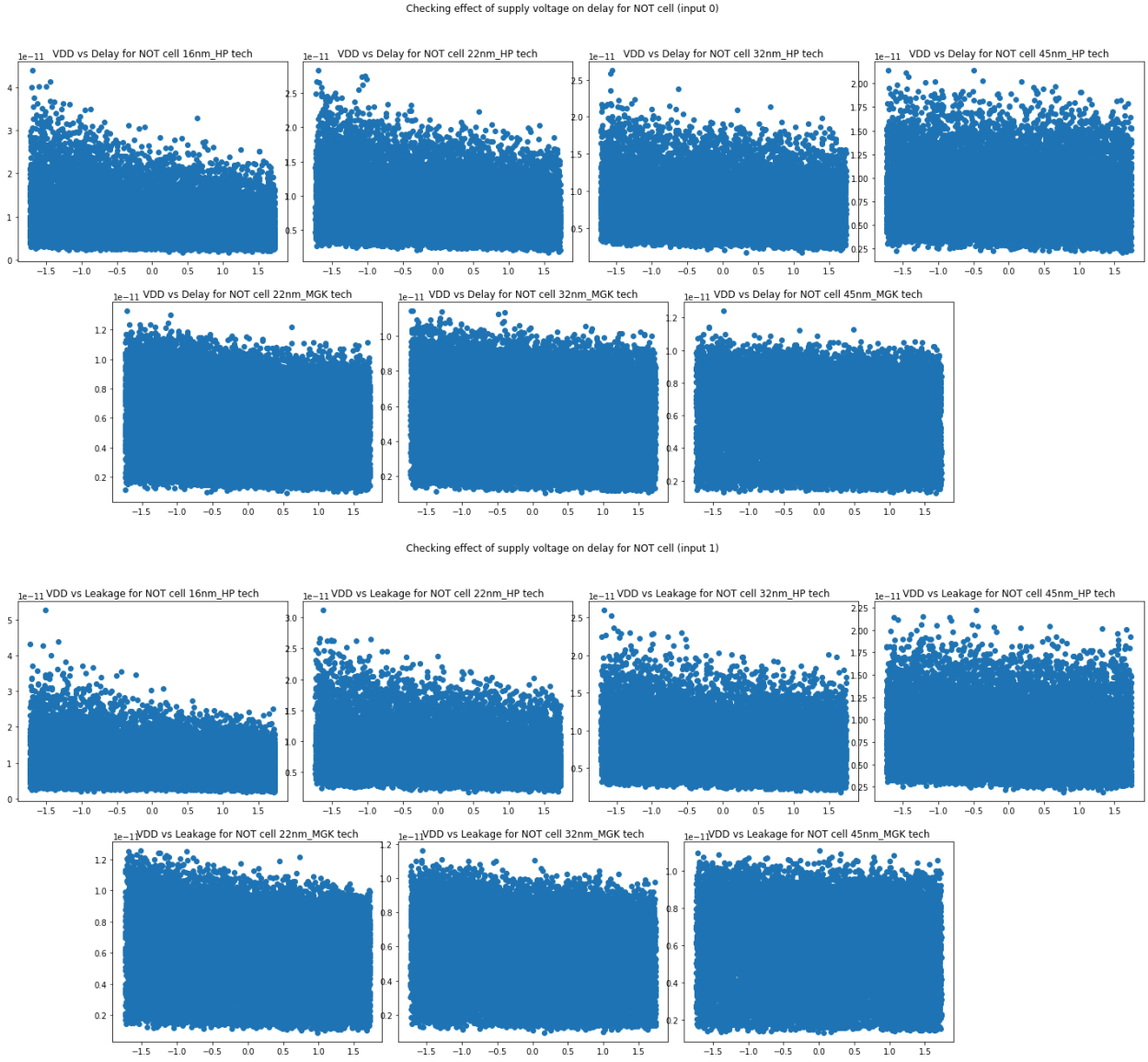
Scatter plots for delay in NOT cell for input 0 and input 1:





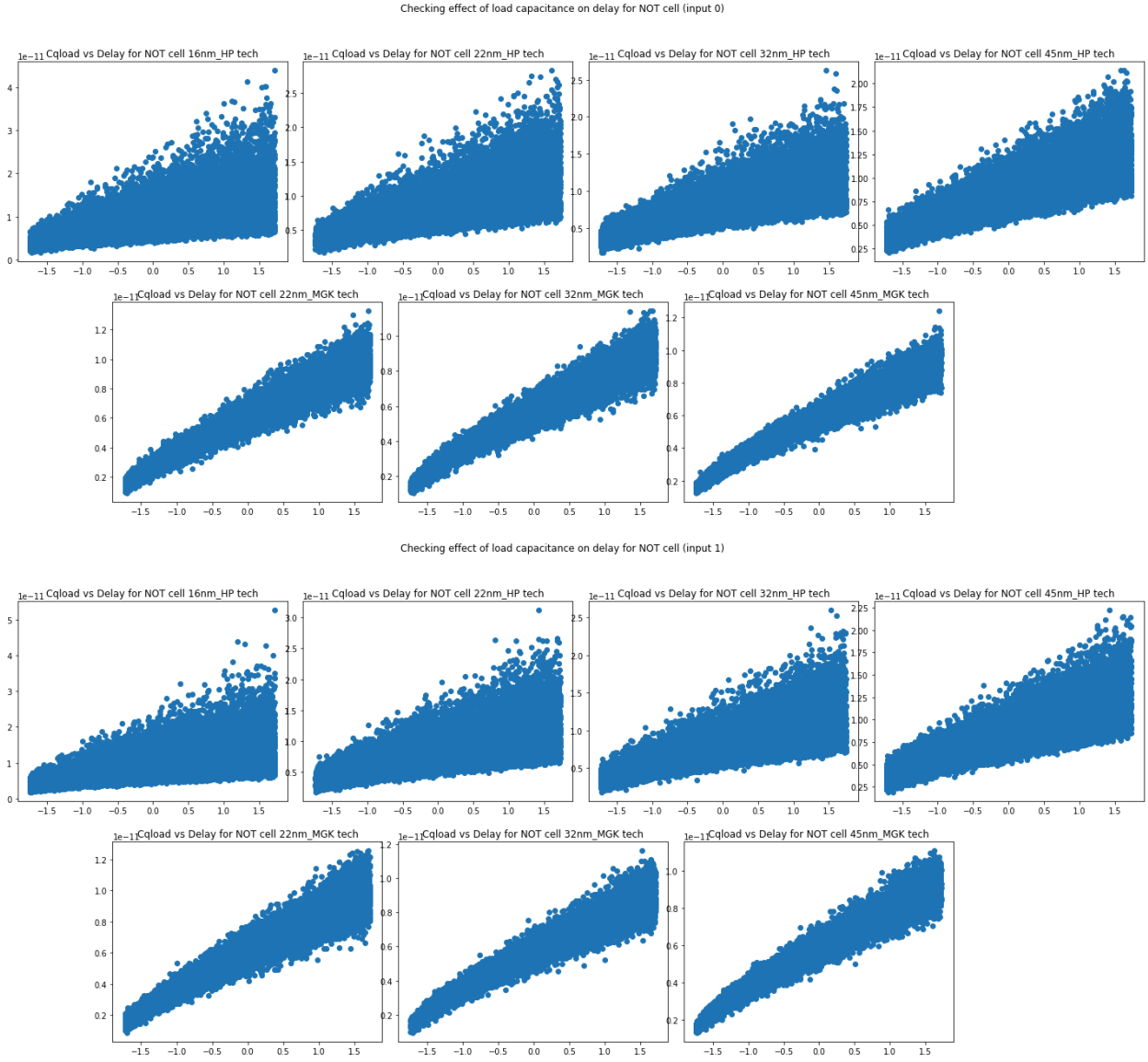
1. We again see that our observations from the correlation maps agree with the general trends exhibited in the scatter plots above.

Analysis of the effect of supply voltage on delay:



1. The supply voltage has a weakly moderate negative effect on the delay for HP tech nodes, which disappears as the node changes from 16nm to 45nm.
2. This dependence is not seen in the case of MGK.

Analysis of the effect of load capacitance on delay:



1. In all technology nodes, there is a clear positive correlation between the load capacitance (cqload) and the delay. As the load capacitance increases, the delay also increases, which is expected because higher capacitive loads require more time to charge or discharge, leading to slower switching speeds.
2. This positive correlation is visible to a much greater extent for MGK tech nodes than HP tech nodes, though the latter also shows this general trend.

Machine Learning modeling

Parameters to be estimated :- ***Power leakage***
Methodology adopted :- ***Meta Learning***

1. Introduction

- Project Objective: The project aims to estimate the leakage power of C499 (CLA) by training ML models on its constituent gates.
- Background Information: In the case of complex circuits, it is often difficult to estimate power leakage and its sources due to the complex architecture , which makes ML model training on constituent gates a lot simpler to estimate which can be used to estimate the parameters of the compound gates saves us a lot of time in the process.
- Problem Statement: Estimation of leakage power by using Machine Learning models on circuit PVT values and various other parameters.

2. Data Description

- Feature Explanation: We use multiple PVT parameters like temp,
- Data Preprocessing: We first sort out the 22nm tech node gates and their parameters and apply some basic pre-processing and normalization.

3. Methodology

- Model Selection: We have chosen CatBoost (gradient boost), Random Forest Regressor and Polynomial Regression with Ridge Regularization.
- Training Process: We initially train all 3 models for every gate which would act as our baseline predictor for our meta learners. We select the best base learners which along with the weights generated by our meta learner are summed by elementary multiplication followed by addition to give our final prediction, which would be compared to our ground truth (y_train) to calculate MSE which would be back propagated to our meta learner (CNN) to improve the weights for a better estimation.

4. Implementation

- The training of the meta learner for each gate must be used to estimate the leakage power for C499.

- We generate 10 random PVT samples for the C499 by running its netlist and then the leakages are estimated through NGSpice, which acts as ground truth to the final model.
- This circuit must be converted to a behavioral function which basically calls the gate function (carrying all its PVT values and other necessary conditions).
- Everytime the gate is called , we estimate the power leakage for the particular process and configuration and keep aggregating the total power leakage value for each gate encountered.

Challenges Encountered: The data generated isn't straightforward to directly feed into a meta learner for training. It requires pre-processing and some rough estimate of the hyper parameters to tune our meta learner and base learners in order to get good R2 scores.

5. Results and Evaluation

- Model Evaluation Metrics: We have obtained good R2 scores for the base learners and the meta learners (ranging from 0.6 to 0.9).

6. Conclusion

- Summary: The Machine Learning approach to estimating power leakage for complex circuits using training on constituent gates' parameter space is a novel approach.
- Future Work: We can perform mathematical transformations (such as differentiation, Laplace transform, etc) on our data to capture maximum of the feature space with less number of features (or) obtain more robust features. These transformations can in fact reduce our training and testing time and probably improve accuracy of the model itself .