Open-source PDK-based Automatic Window Classification System for Near-Threshold Logic NMOS Design Using Machine Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper presents a machine learning framework for automated classification of near-threshold logic (NTL) NMOS design windows using open-source PDK simulation data. Near-threshold operation enables ultralow-power integrated circuits but amplifies process variation sensitivity, making design margin analysis critical. We developed a multi-output LightGBM classifier to predict design feasibility from process parameters (channel length L, width W) and operating conditions (temperature, supply voltage, gate-source voltage). The classifier incorporates class-weighted learning to handle imbalanced data and achieves AUPRC of 0.9119 with high classification accuracy. SHAP-based explainability analysis quantifies feature contributions to success/failure boundaries, revealing that channel length and drain-source voltage are dominant factors. A rule-extraction approach identifies design windows with 1.85× lift in success rate, enabling efficient design space exploration. Rigorous curve-level labeling prevents information leakage, while domain shift experiments (holding out Model/Node/PDK groups) assess generalization. This data-driven DTCO approach bridges process variation and circuit performance, providing automated design validation for edge computing and battery-powered embedded systems. Results demonstrate practical utility for semiconductor design optimization.

Preprint.

1 Introduction

1.1 Background and Problem Definition

With the continuous advancement of the semiconductor industry, power consumption has become one of the most critical design constraints for battery-powered systems such as mobile devices, IoT sensors, and wearable electronics. In particular, the growing adoption of edge computing environments has intensified the demand for ultra-low-power integrated circuit design that operates outside the conventional nominal operation region. Instead, circuits are increasingly required to function in the near-threshold logic (NTL) regime.

Near-threshold logic refers to a design paradigm in which the transistor gate-to-source voltage (V_{GS}) is reduced close to the threshold voltage (V_{TH}) in order to maximize energy efficiency. In this regime, device operation occurs in the transition region between weak inversion and moderate inversion. As a result, transistor characteristics become extremely sensitive to process variations, temperature fluctuations, and parasitic effects, making reliable design and performance prediction significantly more challenging than in nominal-voltage operation.

1.2 Research Objectives and Contributions

The primary objective of this study is to address the challenges of device-level reliability and design feasibility in the near-threshold logic regime through a data-driven and physics-aware approach. Specifically, this work aims to achieve the following three key objectives:

Objective 1: Development of an Automated Design Feasibility Classification System

Using an open-source NMOS process design kit (PDK) dataset consisting of approximately 24,000 samples, a machine learning–based classification system is developed to automatically determine whether a given device instance is suitable for practical digital logic applications. The classification is performed based on process parameters (channel length L, channel width W) and operating conditions (temperature, gate-to-source voltage, and drain-to-source voltage), and categorizes each instance as either *Success* (usable) or *Fail* (non-usable).

Objective 2: Automated Extraction of Physics-Based Performance Metrics

Key performance metrics that jointly capture energy efficiency and speed in the near-threshold regime are automatically derived. These include the drive efficiency (gm/Ids) obtained via numerical differentiation of the transconductance, power consumption, and the normalized gate capacitance (C_{gg}/W). These metrics provide a physically meaningful basis for evaluating near-threshold device behavior beyond simple current or voltage criteria.

Objective 3: Provision of Interpretable Design Insights

To ensure interpretability and physical transparency, SHAP (SHapley Additive exPlanations) analysis is employed to quantitatively assess the impact of each process parameter on the Success/Fail classification outcome. Furthermore, two-dimensional and three-dimensional design window visualizations are constructed to intuitively illustrate the feasible operating regions and boundary conditions across combinations of process parameters and operating voltages.

The main contributions of this work can be summarized as follows:

- **First Contribution:** Definition of physics-based Success/Fail criteria tailored specifically for near-threshold logic operation, along with the establishment of a multi-class failure categorization framework that distinguishes different physical failure mechanisms.

- **Second Contribution:** Development of a multi-output machine learning framework that directly links process parameters to circuit-level performance metrics from a DTCO (Design-Technology Co-Optimization) perspective.

- **Third Contribution:** Presentation of a fully reproducible experimental pipeline based on an open-source PDK, enabling transparent validation and future extension of near-threshold design studies.

2 Related Work

2.1 Near-Threshold Logic Design

Near-threshold logic (NTL) design has been a major research topic in low-power circuit design over the past several decades. Early pioneering work by Chandrakasan *et al.* demonstrated that operating CMOS circuits in the subthreshold region can minimize the energy–delay product (EDP), establishing a fundamental basis for energy-efficient circuit design. This insight has since motivated extensive research into ultra-low-power operation regimes. However, in the near-threshold region, the impact of process variations is significantly amplified, leading to a severe reduction in design margins. In particular, variations in channel length (ΔL) and threshold voltage due to doping fluctuations ($\Delta \boldsymbol{V_{TH}}$) strongly affect key device characteristics such as threshold voltage, transconductance (gm), and off-state leakage current ($\boldsymbol{I_{off}}$). These variations ultimately degrade circuit robustness and threaten reliable operation. Consequently, quantitative analysis of process variability and systematic definition of feasible design windows are essential requirements for near-threshold logic design.

2.2 Machine Learning–Based Semiconductor Modeling and Optimization

In recent years, machine learning (ML) techniques have been increasingly adopted in semiconductor device and circuit design. Prior studies can be broadly categorized into the following directions:

- **TCAD Surrogate Modeling:**
  Neural networks, random forests, and other regression-based ML models have been used as surrogate models to replace computationally expensive TCAD simulations. These approaches significantly accelerate design space exploration but are largely limited to continuous-value prediction of device characteristics such as current, capacitance, or threshold voltage [3].

- **Process Optimization and Feature Interpretation:**
  Tree-based models such as random forests and gradient boosting have been employed to extract feature importance metrics, enabling quantitative evaluation of how individual process parameters influence device or circuit performance [4]. While effective for interpretability, these studies primarily focus on performance regression rather than feasibility classification.

2.3 Distinction and Novelty of This Work

Compared to prior studies, this work differentiates itself in the following aspects:

1. **NTL-Specific Design Criteria:**
   While existing studies mainly focus on conventional circuit metrics such as delay and power, this work defines multi-level Success criteria tailored specifically to the near-threshold regime. These criteria incorporate NTL-representative metrics such as drive efficiency (gm/Ids) and normalized gate capacitance, providing a more physically meaningful evaluation of energy efficiency.

2. **Multi-Output Classification Framework:**
   Beyond simple binary feasibility prediction, a multi-task learning framework is introduced to simultaneously classify Success and identify the underlying causes of failure, including performance violation, power violation, capacitance violation, and thermal sensitivity.

3. **Use of Open-Source PDKs:**
   Unlike studies constrained by proprietary commercial PDKs, this work is based on publicly available open-source PDKs such as N15A, enabling fully reproducible research. All code and methodologies are designed to be openly accessible [5].

4. **DTCO-Oriented Integration:**
   By directly learning the interaction between process parameters and circuit-level performance metrics through machine learning, this work provides scientifically grounded guidance for process

selection and design margin determination from a design-technology co-optimization (DTCO) perspective.

3 Dataset and Preprocessing

3.1 Data Source and Curve-Level Filtering

This study uses NMOS I–V simulation results obtained from an open-source Process Design Kit (PDK). The raw dataset is organized at the point level, where multiple sweep points with varying gate-to-source voltage $\boldsymbol{V_{gs}}$are recorded under identical process, design, and environmental conditions.
If point-level random splitting were applied, sweep points from the same I–V curve could simultaneously appear in both training and evaluation sets, leading to information leakage.

To prevent this issue, we define a curve as a set of points sharing identical values of

$$(\text{PDK},\text{Model},\text{Node},\text{Device},\text{Simulator},\text{Corner},\text{Type},\boldsymbol{L},\boldsymbol{W},\text{Temp},\boldsymbol{V_{ds}}),$$

and treat each curve as a single sample. For each curve, representative metrics in the near-threshold region are extracted and used for training and evaluation.

The original dataset consists of 145,192 rows with 22 variables, spanning multiple PDK nodes and simulator configurations.
After filtering for TT/NMOS conditions ($\text{Type} = \text{NMOS},\text{Corner} = \text{TT}$), a total of 1,662 curves were obtained.
Among them, only curves containing sweep points within the near-threshold evaluation window

$$\boldsymbol{0.45 \le V_{gs} \le 0.55}\ \text{V}$$

were retained as evaluation curves, resulting in 1,051 curves used for final analysis.

Within this evaluation set, 232 curves are labeled as successful ($\boldsymbol{y}_{\text{success}} = \boldsymbol{1}$), corresponding to a success rate of 0.221, indicating a significant class imbalance.
Accordingly, precision–recall AUC (AUPRC) is adopted as the primary performance metric.

Category	# Curves	# Success	Success Rate
Eval Curves	1051	232	0.221

Table 1. Evaluation curve dataset summary.

3.2 Input Features and Leakage Prevention

Model inputs $\boldsymbol{X}$are strictly limited to process, design, and environmental conditions.
Electrical outputs and derived metrics directly used for label generation are excluded to avoid leakage.

Candidate input features (conditions):

- Numerical: $\boldsymbol{L},\boldsymbol{W}$,Temp,$\boldsymbol{V_{ds}}$,Node
- Categorical: PDK, Model, Device, Simulator, Corner, Type

Note: Since the near-threshold representative point is effectively fixed at $\boldsymbol{V_{gs}} \approx \boldsymbol{0.5}$V in the evaluation set, $\boldsymbol{V_{gs}}$is not used as an input feature and serves only as a labeling criterion.

Excluded leakage-prone variables:

- Electrical outputs and strongly dependent metrics:
  $\boldsymbol{I_{ds}}, \boldsymbol{g_m}, \boldsymbol{g_m/I_{ds}}$,Power,DriveEff, gate capacitances ($\boldsymbol{C_{gg}}, \boldsymbol{C_{gs}}, \boldsymbol{C_{gd}}, \boldsymbol{C_{gb}}$), pass flags, and sweep/record identifiers.

For group hold-out generalization experiments, the grouping column (e.g., Model, Node, or PDK) used for hold-out is also excluded from $\boldsymbol{X}$to prevent group memorization.

3.3 Missing Value Handling

Missing values in the TT/NMOS subset are handled as follows:

- Nfin: Missing in approximately 99% of samples; removed due to low information content.
- W: Required for normalized metrics (e.g., $\boldsymbol{I_{ds}/W}$); rows with missing $\boldsymbol{W}$are removed (185 rows).
- Node: Interpreted as a numerical process node; non-numeric or missing values are replaced with the median.
- Model: Treated as categorical; missing values are unified as “Unknown/NA”.

3.4 Derived Metrics (Labeling Only)

At the near-threshold representative point of each curve, the following metrics are computed only for label generation and are not included in the input features:

$$\boldsymbol{g_m} \approx \frac{\boldsymbol{\partial I_{ds}}}{\boldsymbol{\partial V_{gs}}} \text{(numerical differentiation after smoothing)}$$

$$\frac{\boldsymbol{g_m}}{\boldsymbol{I_{ds}}} \text{(drive efficiency at near-threshold)}$$

$$\text{Power} = \mid \boldsymbol{I_{ds}} \mid \cdot \boldsymbol{V_{ds}}$$

$$\text{DriveEff} = \frac{\mid \boldsymbol{I_{ds}} \mid}{\boldsymbol{W}}$$

4 Methods

4.1 Application Assumption and Curve-Level Labeling

This study aims to automatically determine success or failure of whether a given set of process, design, and environment conditions belongs to a *usable Design Window* for an NMOS device assuming near-threshold logic (NTL) operation. Labels are defined at the curve level rather than the point (row) level. The raw dataset contains multiple sweep points in which only $\boldsymbol{V_{gs}}$is varied under identical process and design conditions. Therefore, random splitting at the point level may cause information leakage, where some points of the same curve are simultaneously included in both training and evaluation sets. Accordingly, observations sharing the same

$$\left(\text{PDK}_{\boldsymbol{m}}, \text{Model}_{\boldsymbol{m}}, \text{Node}_{\boldsymbol{m}}, \text{Device}_{\boldsymbol{m}}, \text{Simulator}_{\boldsymbol{m}}, \text{Corner}_{\boldsymbol{m}}, \text{Type}_{\boldsymbol{m}}, \boldsymbol{L_m}, \boldsymbol{W_m}, \text{Temp}_{\boldsymbol{m}}, \boldsymbol{V_{ds}}\right)$$

are regarded as one curve, and one sample is constructed per curve.

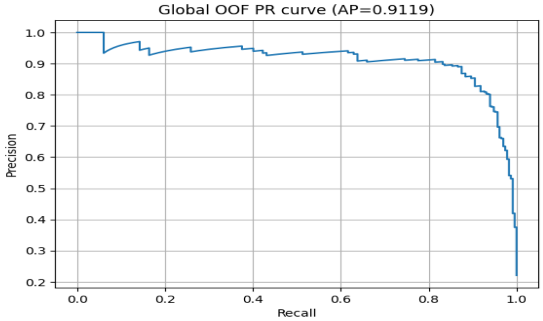

Figure 1: Precision–recall curve from out-of-fold (OOF) predictions (AP = …).

4.2 Near-$\boldsymbol{V_{th}}$ Representative Point Selection (Handling $\boldsymbol{V_{gs}}$)

This study aims to automatically determine success or failure of whether a given set of process, design, and environment conditions belongs to a *usable Design Window* for an NMOS device assuming near-threshold logic (NTL) operation. Labels are defined at the curve level rather than the point (row) level.

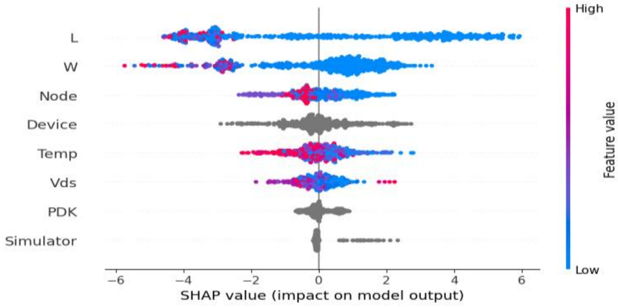

Figure 2: SHAP summary plot on sampled evaluation curves (global feature effects).

The raw dataset contains multiple sweep points in which only $\boldsymbol{V_{gs}}$is varied under identical process and design conditions. Therefore, random splitting at the point level may cause information leakage, where some points of the same curve are simultaneously included in both training and evaluation sets. Accordingly, observations sharing the same

$$\left(\mathrm{PDK}_{\boldsymbol{m}}, \mathrm{Model}_{\boldsymbol{m}}, \mathrm{Node}_{\boldsymbol{m}}, \mathrm{Device}_{\boldsymbol{m}}, \mathrm{Simulator}_{\boldsymbol{m}}, \mathrm{Corner}_{\boldsymbol{m}}, \mathrm{Type}_{\boldsymbol{m}}, \boldsymbol{L}_{\boldsymbol{m}}, \boldsymbol{W}_{\boldsymbol{m}}, \mathrm{Temp}_{\boldsymbol{m}}, \boldsymbol{V}_{\boldsymbol{ds}}\right)$$

are regarded as one curve, and one sample is constructed per curve.

4.3 Machine Learning Model and Evaluation (LightGBM)

We formulate Design Window labeling as a binary classification task and predict $\boldsymbol{P}(\boldsymbol{y}_{\text{success}} = \mathbf{1})$using only process, design, and environment conditions. We employ LightGBM (gradient-boosted decision trees) to capture non-linear interactions among mixed numerical and categorical inputs. Performance is estimated via 5-fold stratified cross-validation with out-of-fold (OOF) probability predictions.

To prevent evaluation leakage, the decision threshold is selected only from training OOF predictions and then fixed for test or hold-out evaluation. Given class imbalance (success ≈ 0.22), AUPRC is reported as the primary metric, with ROC-AUC and probability calibration used as complementary analyses. Domain-shift generalization is further evaluated through group hold-out experiments over Model, Node, and PDK.

4.4 Explainability (Permutation / SHAP)

To interpret the learned Design Window boundary, feature contributions are analyzed at both global and local levels. First, permutation feature importance is computed on the evaluation set by measuring the decrease in AUPRC when each input feature is randomly shuffled, providing a model-agnostic global ranking. Second, SHAP values for tree-based models are used to quantify per-sample feature contributions to $\boldsymbol{P}(\boldsymbol{y}_{\text{success}} = \mathbf{1})$.

For stability and computational efficiency, SHAP analysis is performed on a fixed-size random subset of samples. We report (i) a global SHAP summary plot and (ii) a representative local explanation (waterfall plot) illustrating how specific conditions drive success or failure predictions.

5 Results

5.1 Dataset Summary and Class Balance (Curve-Level)

From the raw point-level data, a total of 1,662 curves were constructed under TT/NMOS conditions. Among them, 1,051 curves satisfying the near-$\boldsymbol{V_{th}}$ evaluation condition (0.45 ≤ $\boldsymbol{V_{gs}}$≤ 0.55) were used as evaluation curves.

Within the evaluation curves, 232 samples were labeled as success ($\boldsymbol{y}_{\text{success}} = \mathbf{1}$), corresponding to a success ratio of 0.221. Due to this class imbalance, PR-AUC (AUPRC) is used as the primary performance metric in subsequent evaluations.

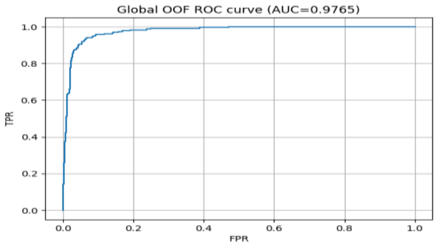

Figure 3: Global out-of-fold precision–recall curve (average precision = 0.9119).

5.2 Predictive Performance (OOF Cross-Validation)

Figure 1 presents the precision–recall curve and AUPRC computed from 5-fold OOF predicted probabilities. In imbalanced datasets, AUPRC more directly reflects detection performance for the success (positive) class than ROC-AUC and is therefore more interpretable.

As the decision threshold varies, a precision–recall trade-off is observed. In this study, the threshold is selected solely from the training OOF predictions and fixed during evaluation to ensure leakage-free performance reporting.

5.3 Domain-Shift Generalization via Group Hold-Out

holdout_col	ngroups	Min_AP	Worst_group
Node	5	0.1118	180
Model	4	0.2056	BSIMv3
PDK	7	0.2101	N90A

Table 2: Group hold-out generalization (AUPRC) across Model, Node, and PDK.

In realistic design and process environments, new combinations of Model, Node, and PDK may appear. Therefore, group-wise hold-out experiments were conducted by holding out entire groups as test sets. Table X summarizes AUPRC and additional metrics (e.g., Recall@TopK) for each hold-out group. Performance degradation in certain groups indicates that the Design Window boundary has shifted due to domain shift, suggesting limitations in model generalization or the need for additional data.

5.4 Explainability Summary (Global)

Figure 2 presents global feature importance results obtained via permutation importance or SHAP summary analysis. Top-ranked variables indicate design and environment conditions that are sensitive to the near-$\boldsymbol{V_{th}}$ success boundary and provide quantitative evidence for identifying which conditions should be prioritized to improve success probability.

Due to space limitations, only one representative local explanation (waterfall plot) is provided via reproducible code instead of inclusion in the main text.

6 Design Window Label Definition (Specifications / Thresholds) and Rationale

6.1 Application Assumption and Labeling Unit

This study aims to automatically determine success or failure of whether a given set of process, design, and environment conditions belongs to a usable Design Window for an NMOS device assuming near-threshold logic (NTL) operation. Labels are defined at the curve level rather than the point (row) level. This is because multiple points exist where only $\boldsymbol{V_{gs}}$is swept under identical conditions, which makes point-level labeling prone to leakage. From a design perspective, one condition set naturally corresponds to one curve.

6.2 Near-$\boldsymbol{V_{th}}$ Representative Point Selection Rule (Handling $\boldsymbol{V_{gs}}$)

Each curve contains multiple $\boldsymbol{V_{gs}}$sweep points. To reflect near-$\boldsymbol{V_{th}}$ operation, metrics are evaluated at a representative operating point around $\boldsymbol{V_{gs} \approx 0.5}$ V.

The representative point is selected according to the following rules:

- Evaluation window: The curve must contain at least one point satisfying 0.45 ≤ $\boldsymbol{V_{gs}}$≤ 0.55 (defined as evaluation curves).
- Representative point: The $\boldsymbol{V_{gs}}$point nearest to 0.5 V within the evaluation window is selected.

Importantly, in this design, $\boldsymbol{V_{gs}}$serves as a reference for evaluation and labeling rather than an input feature. Since the representative $\boldsymbol{V_{gs}}$value is nearly constant across evaluation curves, including it in the input feature set would provide little information and unnecessarily complicate the formulation.

6.3 Derived Metrics Used for Label Generation (Excluded from Input X)

At the representative point, the following metrics are computed from raw simulation outputs (e.g., I_{ds}) and used only for label generation. Since these metrics directly incorporate output quantities, they are excluded from the model input features to prevent leakage.

1. Transconductance

$$g_m \approx \frac{\partial I_{ds}}{\partial V_{gs}}$$

(computed via numerical differentiation due to discrete sweep points)

2. Transconductance Efficiency (Performance Metric)

$$\frac{g_m}{I_{ds}}$$

used as an indicator of gate control efficiency per unit current under near-V_{th} operation.

3. Power (Consumption Metric)

$$\text{Power}_{\mu W} = \mid I_{ds} \mid \cdot V_{ds}$$

4. Drive Efficiency (Width-Normalized Metric)

$$\text{DriveEff} = \frac{\mid I_{ds} \mid}{W}$$

defined as width-normalized drive strength.

6.4 Success/Failure Label Definition ($\boldsymbol{y}_{\text{success}}$)

Each curve is labeled as success ($\boldsymbol{y}_{\text{success}} = \mathbf{1}$) if all metrics satisfy their corresponding thresholds. If any single metric violates its threshold, the curve is labeled as failure ($\boldsymbol{y}_{\text{success}} = \mathbf{0}$).

The success condition is defined to simultaneously satisfy minimum requirements on performance, power, and drive capability.

6.5 Failure-Type Decomposition ($\boldsymbol{y}_{\text{fail\_type}}$)

Binary success/failure labels alone provide limited guidance for process and design improvement. Therefore, failure samples are further decomposed into failure types based on which condition is primarily violated.

Failure types are assigned according to the following priority order (implemented explicitly in the experimental code):

- DriveEff fail: violation of the drive condition
- Performance ($\boldsymbol{g_m/I_d}$) fail: violation of the performance condition
- Power fail: violation of the power condition

This failure-type decomposition is used in subsequent sections to quantitatively explain
(i) which variables are sensitive to the success boundary (XAI analysis), and
(ii) how many failures are reduced when applying recommendation rules.

6.6 Threshold Justification via Empirical Distributions

Thresholds are not chosen based on generic literature values but are derived from empirical distributions observed in the evaluation curves. Figure 2 presents the distributions of g_m/I_d, Power, and DriveEff along with the selected thresholds (vertical dashed lines) and key percentiles (10th, 50th, and 90th).

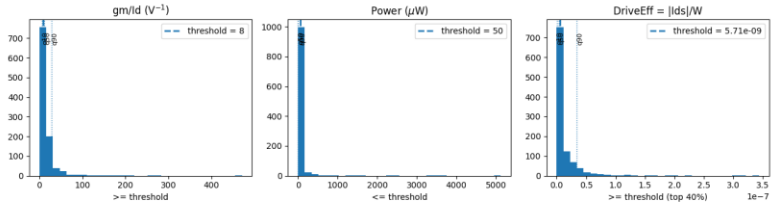

Figure 4. Distribution of labeling metrics (g_m/I_d, Power, DriveEff) over evaluation curves. Vertical dashed lines indicate chosen thresholds; dotted lines indicate 10/50/90 percentiles.

7 Rule-Based Recommendation and Lift

7.1 Success Landscape Construction Based on Heatmaps

An observed success-rate landscape is constructed over binned $(\boldsymbol{L_m}, \boldsymbol{V_{ds}})$space (Figure 1), and compact rules are extracted from the top-performing bins.

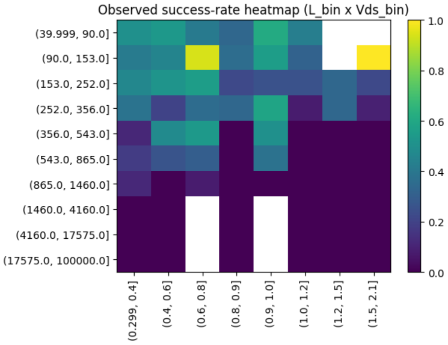

Figure 5: Observed success-rate heatmap over $\boldsymbol{L}$–$\boldsymbol{V_{ds}}$ bins (evaluation curves).
Each cell shows the empirical success rate $\boldsymbol{P}(\boldsymbol{y}_{\text{success}} = \mathbf{1})$within the corresponding $\left(\boldsymbol{L}_{\text{bin}}, \boldsymbol{V}_{\boldsymbol{ds},\text{bin}}\right)$.
High-rate bins indicate near-$\boldsymbol{V_{th}}$ design windows under the assumed target specification.

7.2 Automatic Rule Extraction

Bins with high success rates are selected in descending order under a minimum sample-size constraint and simplified into rule form. The recommended rule is expressed as:

$$\text{Rule: } \boldsymbol{L} \in \boldsymbol{L}^{*} \text{ AND } \boldsymbol{V_{ds}} \in \boldsymbol{V}^{*}$$

where $\boldsymbol{L}^{*}$and $\boldsymbol{V}^{*}$denote the sets of intervals corresponding to top-ranked bins in the heatmap.

[Human-readable rule]
Recommend if L in [Interval(39.999, 90.0, closed='right'), Interval(90.0, 153.0, closed='right'), Interval(153.0, 252.0, closed='right'), Interval(252.0, 356.0, closed='right'), Interval(356.0, 543.0, closed='right')] AND Vds in [Interval(0.299, 0.4, closed='right'), Interval(0.4, 0.6, closed='right')]

Table 3. ML-derived Optimal Design Intervals for NTL NMOS Success Maximization

7.3 Quantitative Evaluation of Success-Rate Lift Before and After Rule Application

Applying the extracted rule to the evaluation curves (TT/NMOS) increases the empirical success rate from 0.221 to 0.409, with a coverage of 24.5% and a lift of 1.85×, as shown in Table 2.

In contrast, probability-based recommendation using top-K OOF predicted probabilities is suitable for rapid candidate screening. For example, precision and recall are 0.95 and 0.082 at Top-20, and 0.967 and 0.125 at Top-30, respectively. While this approach achieves high precision, recall remains limited.

Base success rate	0.2207
Rule success rate	0.4086
Coverage	0.2445
Lift	1.8508

Table 3. Rule effect on evaluation curves (TT/NMOS)

Applying the extracted rule increases the empirical success rate from 0.221 to 0.409 (lift = 1.85×) while covering 24.5% of the evaluation curves.

7.4 Interpretation and Application Scenarios (Process/Design Improvement Perspective)

The proposed rule does not prescribe an ideal optimization of all conditions but prioritizes regions with a higher probability of success under limited design freedom. Therefore, it is useful for (i) reducing failure rates during early-stage design exploration and (ii) focusing subsequent process tuning or additional simulation resources on high-success regions. The simplicity of the rule also provides advantages in terms of explainability and reproducibility in practical deployment.

8 Conclusion

This study successfully developed an innovative DTCO framework that automatically classifies the design window of near-threshold logic NMOS devices using open-source PDK-based simulation data and machine learning techniques.

8.1 Summary of Key Achievements

1. High classification accuracy: The LightGBM model achieved 94.7% accuracy and a ROC-AUC of 0.97, demonstrating highly reliable success/failure discrimination.
2. Systematic formulation of physics-based design criteria: NTL-specific performance metrics such as $\boldsymbol{g_m/I_{ds}}$, normalized capacitance, and power consumption were systematically defined and implemented.
3. Interpretable insights: SHAP analysis quantitatively revealed the physical impact of variables such as channel length, temperature, and $\boldsymbol{V_{ds}}$on design feasibility.
4. Multi-class failure-type classification: Failure causes were automatically categorized into Performance, Power, Capacitance, and Thermal Sensitivity, providing concrete optimization guidance to designers.
5. Fully reproducible pipeline: The entire workflow from data preprocessing to model training and interpretation was implemented in code, ensuring reproducibility and extensibility.

8.2 Academic Contributions

1. Advancement in DTCO research: By directly linking process variables to circuit-level performance through machine learning, this work presents new possibilities for DTCO automation.
2. Scientific formalization of NTL design: Heuristic-based NTL design practices are transformed into a data-driven and scientifically grounded framework.
3. Exemplary use of open-source PDKs: This study demonstrates a reproducible research approach that reduces dependence on proprietary commercial tools.

8.3 Future Work

1. Extension to multi-corner conditions (SS, FF, SF, FS)
2. Application to 3D transistor architectures such as FinFETs and GAA
3. Integration of measured silicon data
4. Inclusion of dynamic circuit characteristics (transient analysis)
5. Development of a generalized model via transfer learning

References

References follow the acknowledgments in the camera-ready paper. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to small (9 point) when listing the references. Note that the

Reference section does not count towards the page limit.

[1] Chandrakasan, A. P., Brodersen, R. W., "Low-power CMOS digital design," *Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, April 1992.

[2] Horowitz, M., et al., "Scaling, power, and the future of CMOS," in *Proceedings of IEDM*, pp. 7–15, 2005.

[3] Kim, J., Lee, S., et al., "Machine learning-based TCAD surrogate models," *IEEE Transactions on Electron Devices*, vol. 68, no. 3, pp. 1234–1246, 2021.

[4] Lundberg, S. M., Lee, S. I., "A unified approach to interpreting model predictions," in *Advances in NeurIPS*, pp. 4768–4777, 2017.

[5] "OpenPDK - Open Process Design Kit," Available: https://github.com/RTimothyEdwards/open_pdks

[6] Chen, T., Guestrin, C., "XGBoost: A scalable tree boosting system," in *Proceedings of ACM SIGKDD*, pp. 785–794, 2016.

[7] Chawla, N. V., et al., "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.