Enhancing PDK Library Validation with Machine Learning: A Novel Approach to Layout Comparison

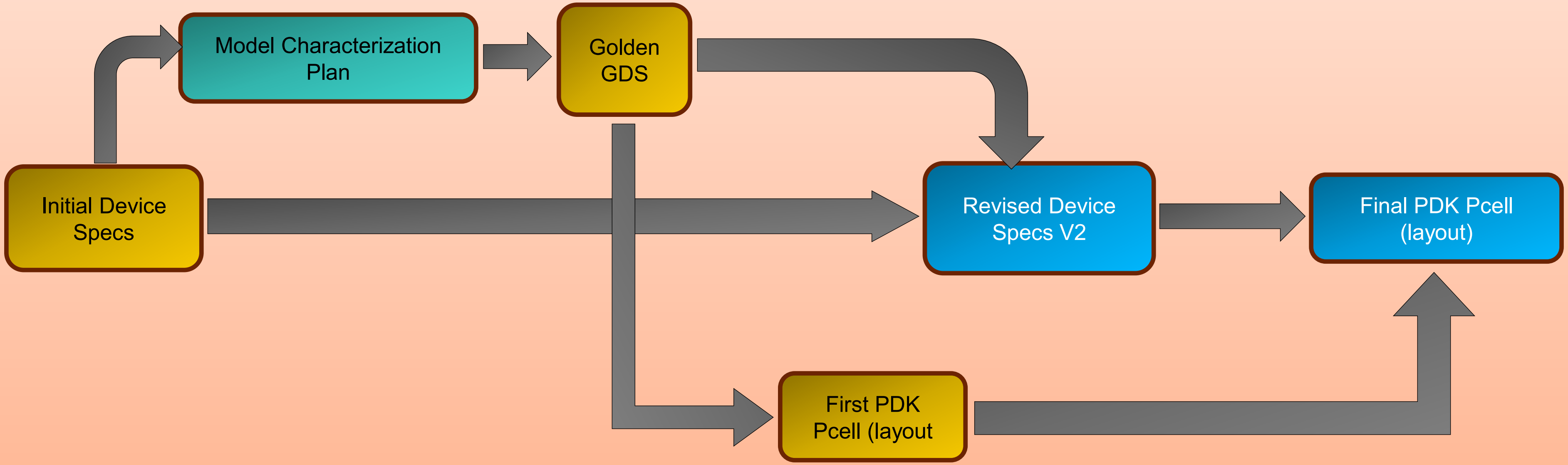
Patent pending

Farzana Akhter, Engineering Intern, Design Enablement PDK
Nolan Pavek, Principal Eng Design Enablement PDK
Romain Feuillette, Director, Design Enablement PDK

Context and Motivation

Patent pending

PDK models are based on silicon data from "Golden GDS" layouts, which serve as the benchmark for model accuracy. As a device evolves, its physical layout (PCell) may need updates to accommodate model fine tuning or improve performance. Ensuring these updates remain consistent with the original Golden GDS is crucial for maintaining model accuracy.



The **main challenge** for the PDK team is ensuring that updates to the production layout PCell do not introduce unintended changes that compromise the model fidelity of the Golden GDS. Although tools like GDSXOR can identify layout differences, they struggle to distinguish between expected and unexpected modifications, making it difficult to verify layout integrity without extensive manual checks.

Limitations and solution

Patent pending

Limitations

Time consuming Processes: Manual review of layout variations is time consuming

Risk of Model-Layout Mismatch: Without a thorough comparison mechanism, unintended changes may go unnoticed.

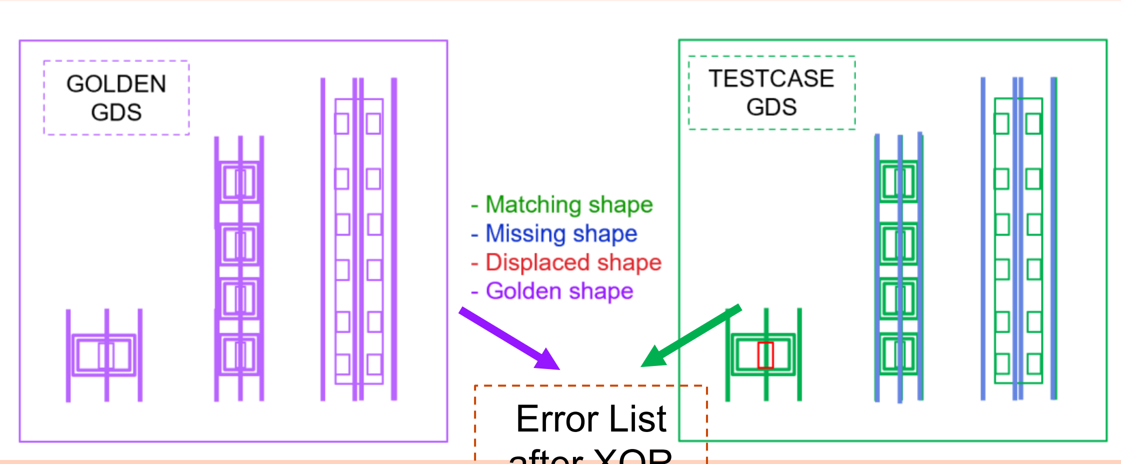
Solution

Develop a validation mechanism that distinguishes between intentional layout updates and unintended deviations from the Golden GDS.

Utilize Machine learning to enhance the automation of classifying and flagging layout changes

Automating PDK Validation with Machine Learning

Patent pending



Error List after XOR comparison

Gather GDS files for comparison

Noise reduced to focus on meaningful changes

Identify impacts to functionality or manufacturability

LAYER DIFFERENCES

nfet_CDNS_730206218738 bbox different from Golden GDS
nfet_CDNS_730206218348 bbox different from Golden GDS
Number of shapes on layer: PC drawing don’t match ref:15 golden: 17
Number of shapes on layer: RX drawing don’t match ref:15 golden: 13
Label pitch1 with origin (3.1888, 1.089) not found in reference GDS
Label pitch1 with origin (5.1888, 1.499) not found in reference GDS

- **Geometric & Dimensional Issues:** Shape, scaling, alignment errors.
- **Layer & Masking Errors:** Incorrect or missing layers.
- **Connectivity & Placement Errors:** Misconnections, floating elements, misplaced or duplicated components.
- **PCell Parameter Errors:** Incorrect or unintended parameter changes.
- **Marker/Annotation Issues:** Misaligned or missing markers and annotations.

Automating PDK Validation with Machine Learning

Patent pending

How ML improves PDK Validation

Error Detection and Optimization: ML enhances layout checks, predicts potential issues, and reducing trial-and-error efforts.

Enhanced Simulation and Data Management: ML boosts simulation accuracy and automates data classification for efficient workflows.

Apply ML to Classify Layer Differences

Possible output from the ML model:

No Error: Inconsequential differences that can be ignored.

Minimal Concern: Minor issues that may require further investigation.

Possible Error: Potential problems that could range from minor to significant.

Definite Error: Issues that need to be addressed and fixed.

4

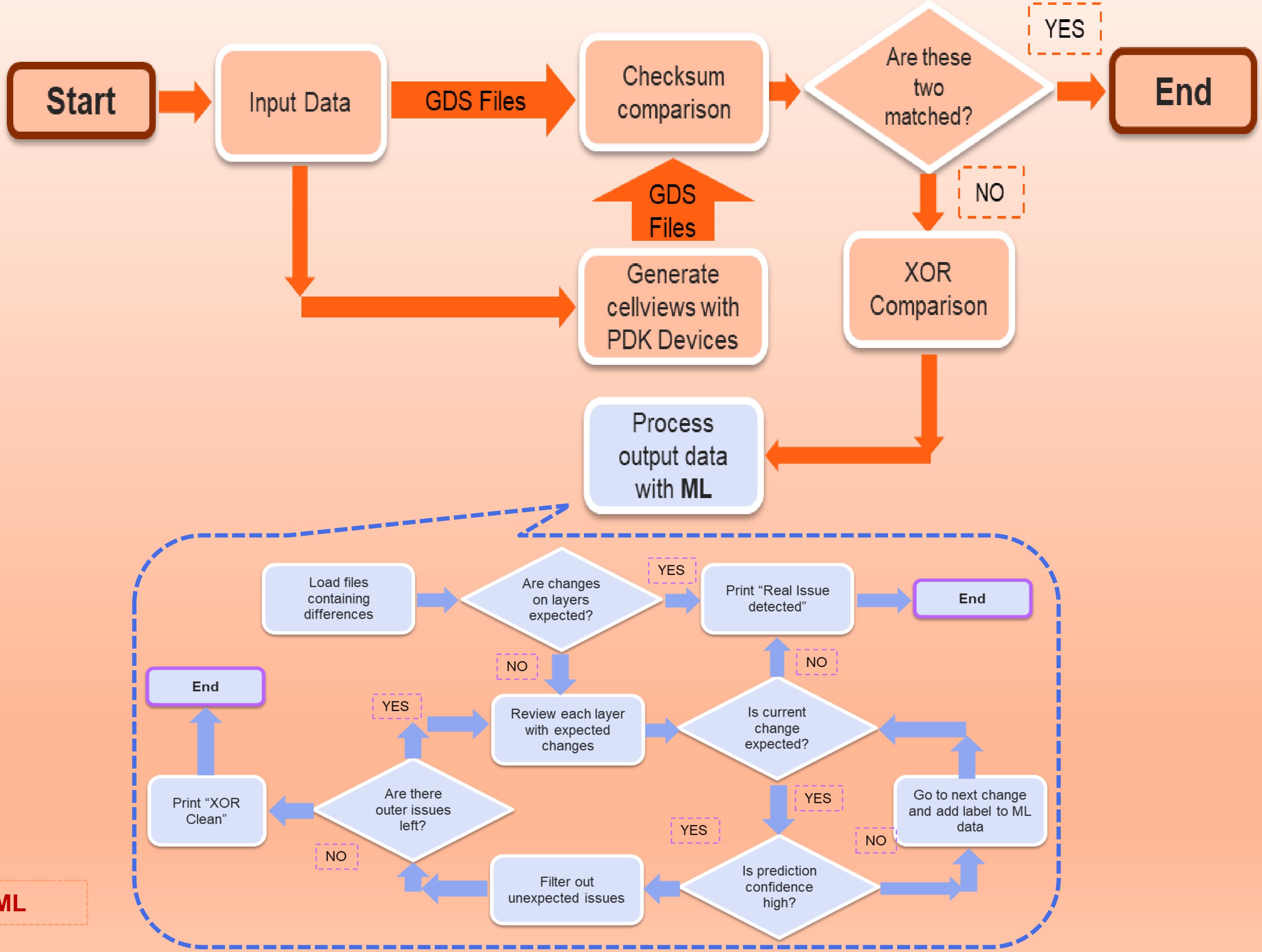
ERROR TYPE
No Error
No Error
Possible Error
Possible Error
No Error
No Error

The ML model categorizes differences into four distinct classes, allowing engineers to identify and resolve layout issues with precision quickly.

Proposed ML Enhancements

Machine learning automates the identification and classification of discrepancies between production pcells and Golden GDS files in large, complex designs.

It accurately classifies differences based on learned patterns, significantly improving efficiency and reducing the potential for human error.



Processing output data from XOR comparison with ML

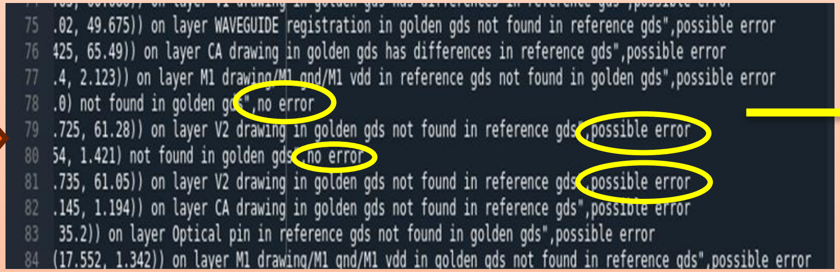
Proposed ML Enhancements

Patent pending

Detailed Workflow for ML-Based PDK Error Detection and Reporting

User input GDS files
→ ML Model loads data
→ ML compares layouts
→ Errors detection by ML
→ Automatic report generation
→ User reviews results

- Data is loaded and preprocessed
- Text is converted into sequences
- Error classification by CNN Model
- Predictions are saved as a report



Error label predicted by Machine Learning Model

Leveraging Python Based Machine Learning Model for PDK Validation

Evidence

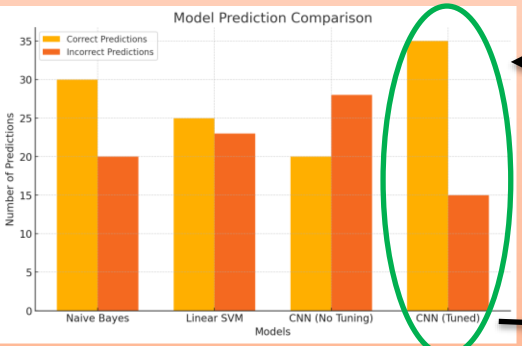
Model	Sampling	Class	Precision	Recall	F1-Score	Accuracy
Linear SVC	Down sampling	0	1.00	0.50	0.67	80%
		1	0.75	1.00	0.86	
	Up sampling	0	0.96	0.81	0.88	90%
		1	0.86	0.97	0.84	
Naïve Bayes	Down sampling	0	0.89	0.80	0.84	89%
		1	0.89	0.94	0.92	
	Up sampling	0	0.90	1.00	0.95	94%
		1	1.00	0.86	0.92	

Metric	Result
Good accuracy	Achieved
Classification	Not resolved
Over fitting	Not resolved

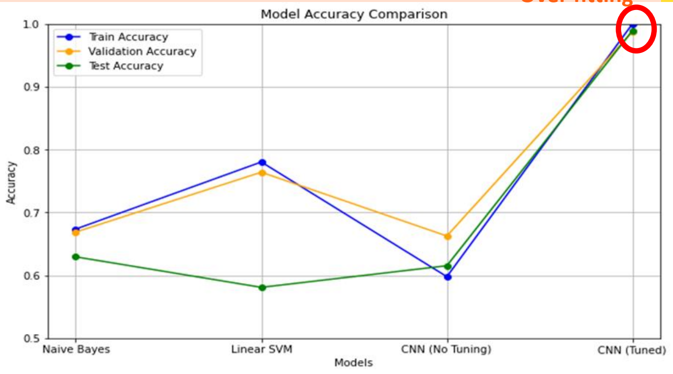
Shifted to multi-class classification

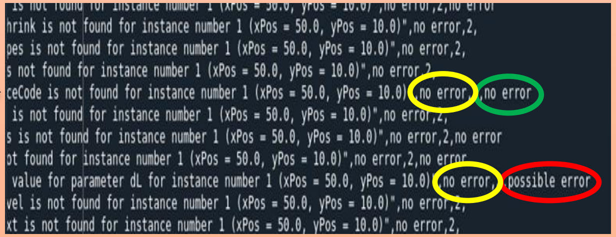
Metric	Result
Good accuracy	Achieved
Classification	Resolved
Over fitting	Not resolved

Level	Training Count	Testing Count
Definite error	257	254
Minimal concern	238	230
No error	162	186
Possible error	950	814



Over fitting





Actual error type

Correct prediction by ML

Incorrect prediction by ML

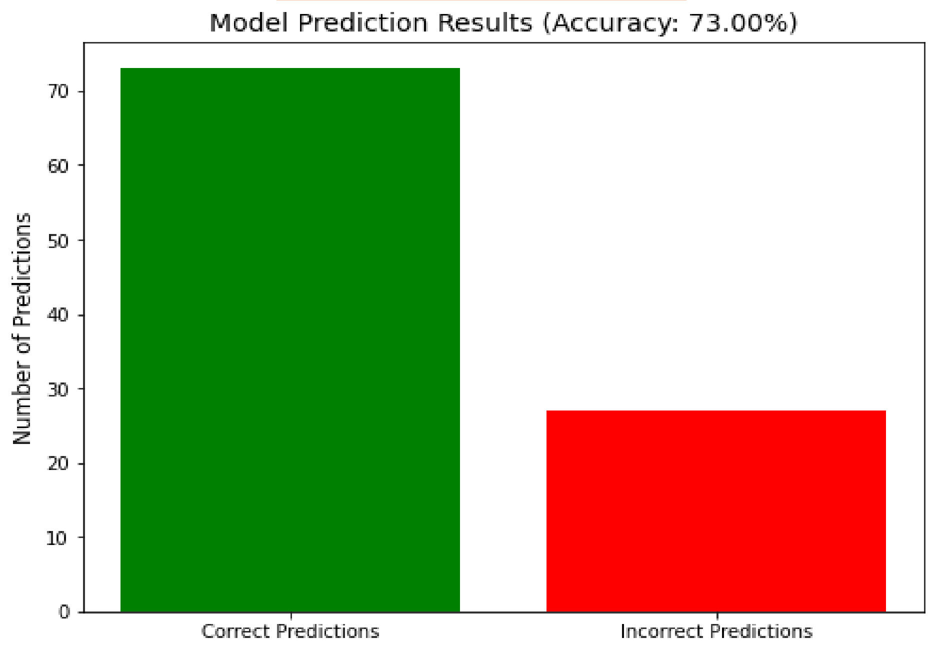
GlobalFoundries © 2024 All Rights Reserved

Final Model

We resolved the over fitting issues by applying regularization and selected the CNN for the final implementation.

Final Model Prediction Results

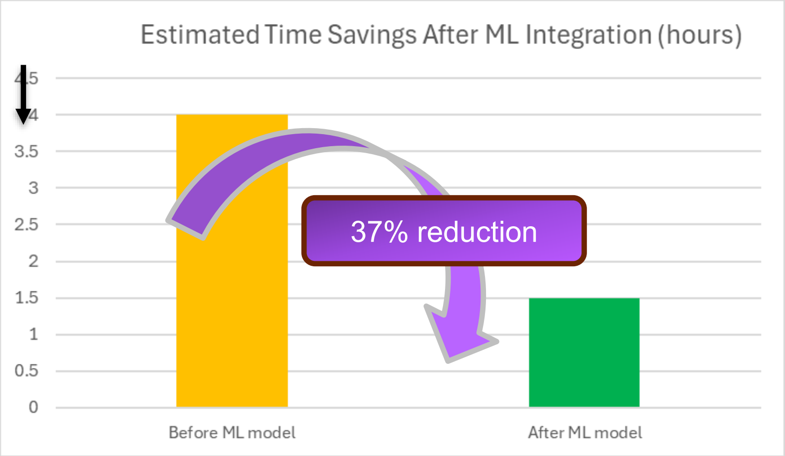
Metric	Result
Good accuracy	Achieved
Classification	Resolved
Over fitting	Resolved



Web integration

Web integration

Description	ML Prediction
Number of shapes on layer:	possible error
Number of shapes on layer:	possible error
dgnfeth_bc_CDNS_731080	no error
dgnfeth_bc_CDNS_731080	no error



Summary

Patent pending

Started with a dataset of 1,500 lines of error data to develop models for classifying XOR errors.

Improved accuracy with sampling techniques, resulting in more balanced and reliable predictions.

Expanded to a multi-class dataset of six devices.

Employed L2 regularization and Dropout to reduce overfitting.

These techniques significantly improved the model, leading to a **73%** correct prediction rate for the error type.

Tested CNNs on multiple datasets to find errors in test cases.

We found indicators of overfitting which resulted in our model struggling to generalize to new data

Successful web integration has been successfully completed.

ML integration reduces error detection time by **37%**, leading to significant efficiency gains.

Future work

Patent pending