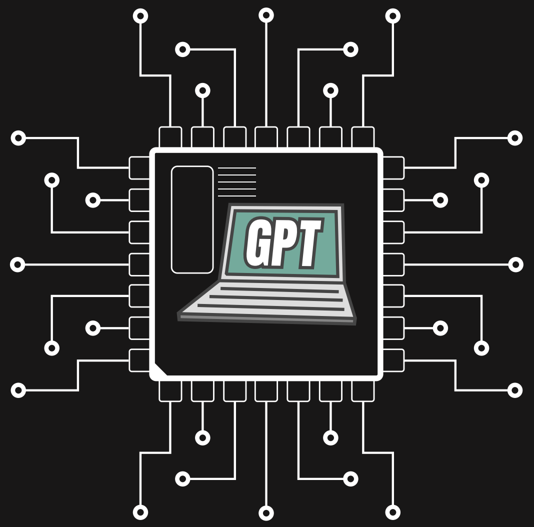

GPT

Stacked Ensemble Learning-Enabled Early Prediction of MBIST Area and Test Time in Complex Memory IPs

SEEL4824 Final Year Project Part II

Presenter:

Chee Jin Teoh

A21EE0295

Main Supervisor:

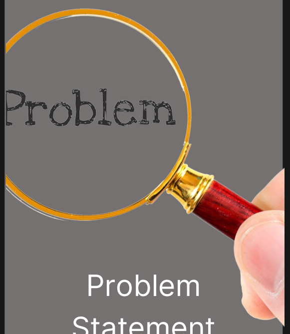
Prof. Madya Ir. Dr. Ab Al-Hadi bin Ab Rahman

Industry Supervisor:

Mr. Johnny, Kee Hui Wong

Mr. Siang Leong Foo

Outlines



Problem Statement

Up to

21.2x

more accurate area prediction

Up to

20x

more accurate test time prediction

Up to

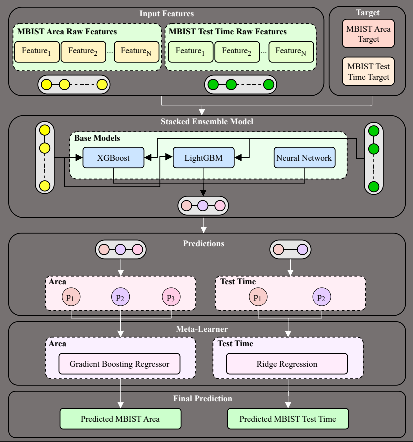
8x

more reliable in ablation study

Conclusion

Ensemble Learning

Stacked Ensemble Architecture



Results & Discussion

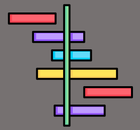
Project Background

References

Objectives

Literature Review

Research Methodology



Gantt Chart

Feature Engineering

Q&A

Project Background

MBIST is essential, but it comes with a COST.

70% memory dominates SoC area

In modern chips, memory blocks like SRAM and register files can take up more than 70% of the total area [1-3].

66% MBIST leads in DFT overhead

MBIST makes up around 66% of the total test logic. It's the biggest contributor to DFT area [4, 5].

Slow traditional metrics estimation

Both MBIST area and test time are only available after RTL is converted into a gate-level netlist [6-9].

Early prediction by machine learning

Machine learning models can estimate MBIST impact early using only RTL-level data, without full synthesis [5, 10].

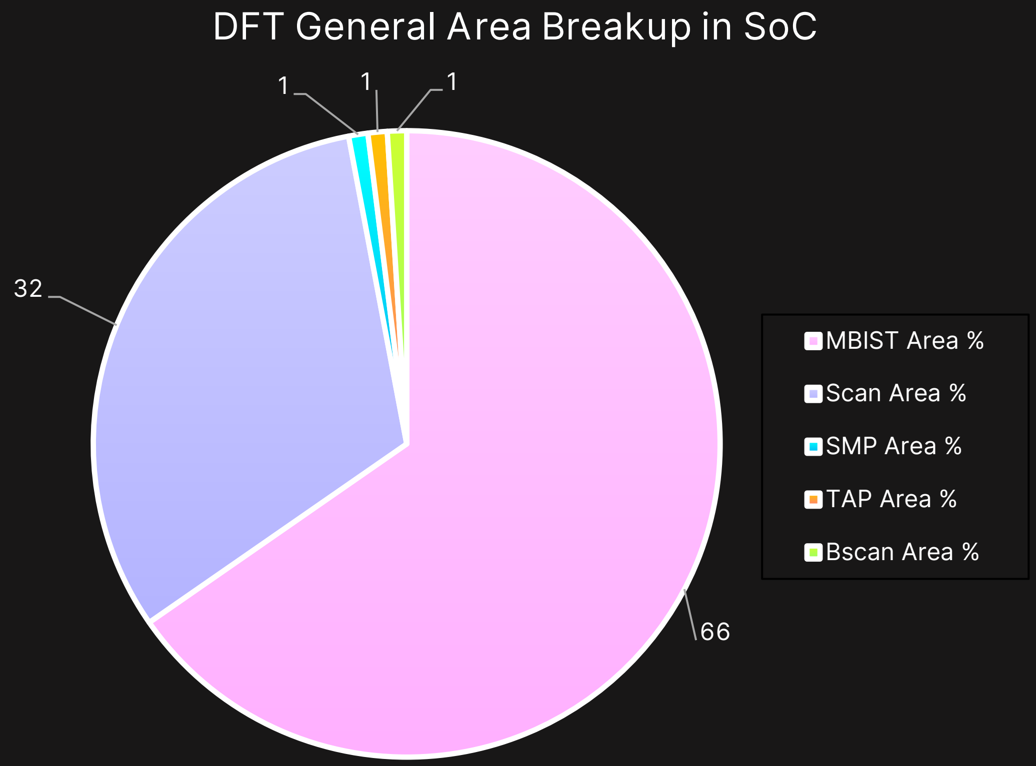


Figure 1: Area distribution of DFT components in an SoC [5].

Problem Statements

Designers still lack of RELIABLE tools for early MBIST estimation.

Late accurate metrics generation

Both MBIST area and test time are measurable after synthesis and pattern generation [11].

Slow design iteration

Changing MBIST settings to meet the constraints means repeating insertion and synthesis. This slows down the design cycle [5, 7, 12].

Hard to make early decisions

Without early estimates, it's hard to compare MBIST configurations or balance area and time [13-16].

Lack of scalable existing models

Most existing ML methods don't handle real SoC designs with complex memory IPs using vital features [14, 15, 17-19].

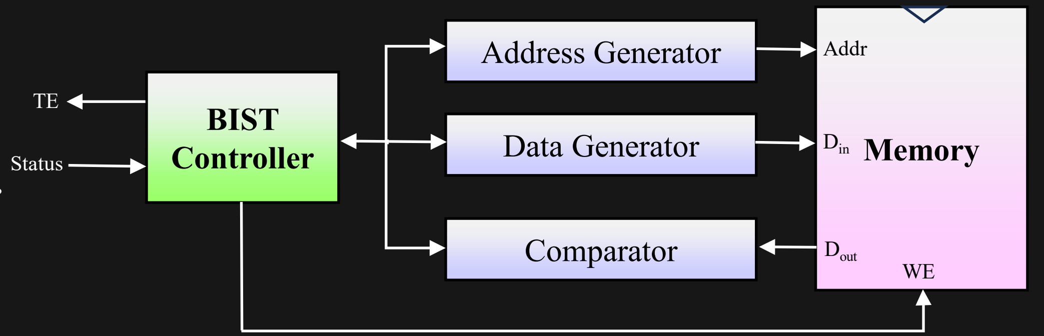


Figure 2: The simplified architecture of an MBIST system [2].

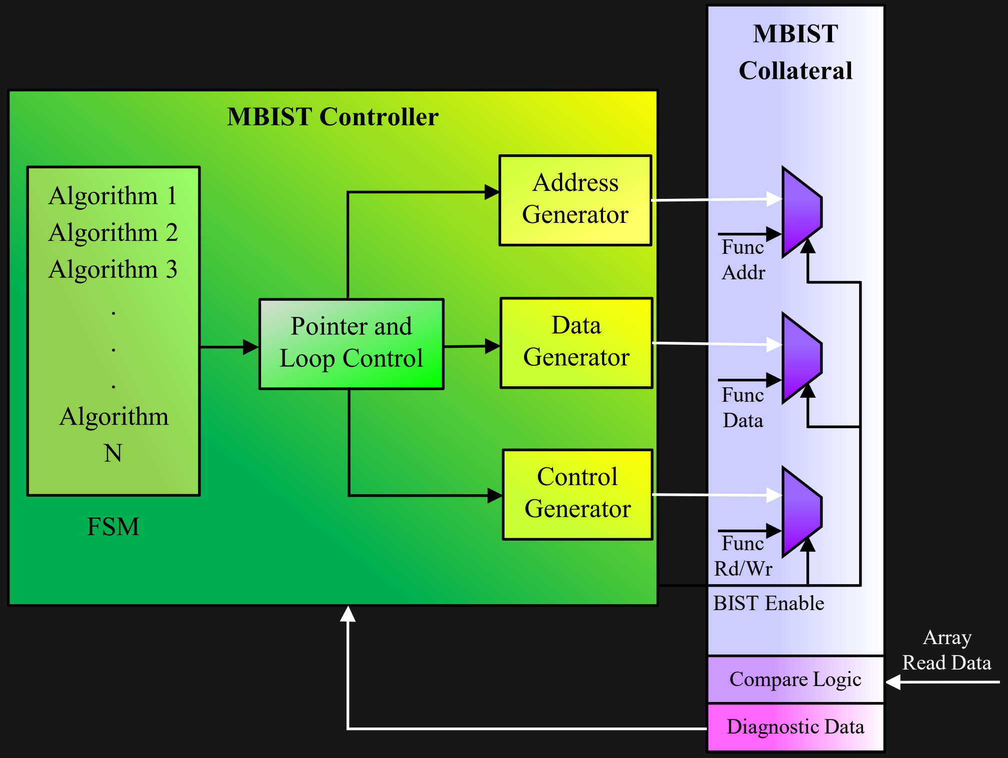


Figure 3: MBIST architecture showing the MBIST Controller with its components and MBIST Collateral [9].

Objectives

Key goals is to improve MBIST area and test time estimation using machine learning and automation

To PROPOSE

better feature engineering techniques

Use RTL-level info like memory size, word width, clock domains, and test settings to capture and model design complexity.

To DESIGN

smarter models – stacked ensemble

Stacked ensemble with XGBoost, LightGBM, Neural Network, and meta-learners to predict MBIST area and test time without synthesis.

To INCORPORATE

AI-driven semantic port classification

Apply AI to classify ports (clock, address, data) from RTL code using semantic analysis, replacing manual tagging.

Scope of Work

From MBIST integration to synthesis-free prediction using machine learning

Integrated

MBIST into RTL designs using Tessent MBIST

Extracted

area using Synopsys Design Compiler and test time using Tessent MBIST

Created

baseline models for comparison (Linear, Polynomial Regression, Gradient Boosting, KNN, Neural Network)

Developed

a stacked ML ensemble combining XGBoost, LightGBM, and Neural Network

Scope of Work

From MBIST integration to synthesis-free prediction using machine learning

Applied

feature engineering using RTL-level memory and port parameters

Generated

over 5,000 IP design variants with varied configurations

Tuned

models' hyperparameter with Optuna for better accuracy

Evaluated

model performance using R^2 score, $MAPE$ and $Accuracy$ with ±10% tolerance

Literature Review

Summary of past MBIST prediction studies and their key limitations

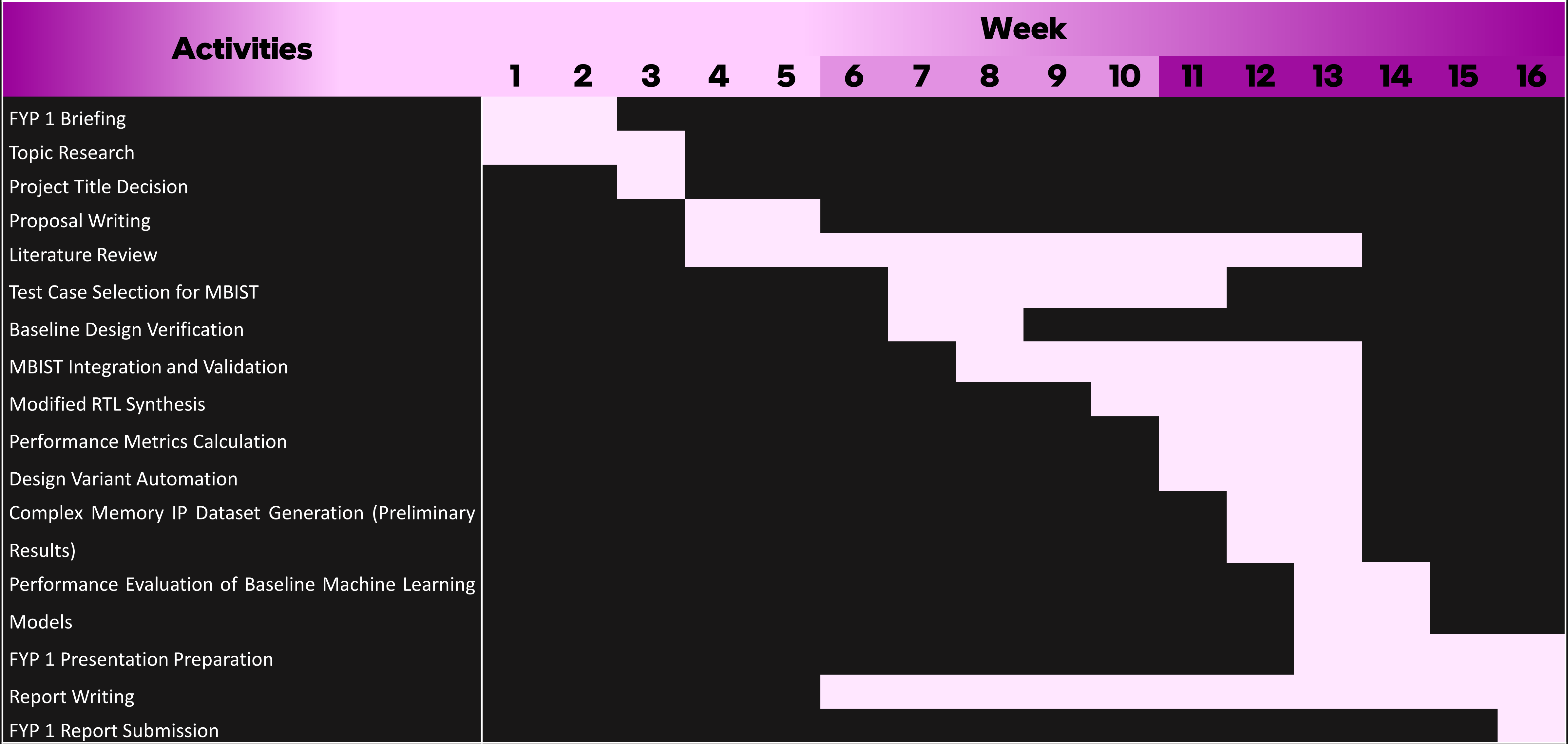
Table 1: Review of related works on machine learning.

Author, Title, Year	Improvement Techniques	Limitations	Key Results
Darakshan Jamal, Ratheesh Thekke Veetil, MBIST Area & Test Time Optimisation Using Machine Learning, 2023 [10]	Logistic, Decision Tree, Polynomial Regression, clustering	Did not include MBIST algorithm type, limiting test time accuracy	40% area reduction, >85% test time accuracy
Puneet Arora et al., "Machine Learning-Based MBIST Area Estimation," 2023 [7]	Random Forest, Neural Network	Basic models, no layout insight or combinational/sequential breakdown	>90% accuracy, faster config evaluation
Darakshan Jamal et al., "Efficient MBIST Area and Test Time Estimator Using Machine Learning Technique," 2023 [5]	Decision Tree, Gradient Boosting, clustering	No clear link between test time and core parameters	>93% area accuracy, >85% test time accuracy
Balwinder Singh et al., "Area Overhead and Power Analysis of March Algorithms for Memory BIST," 2012 [12]	Verilog-based March algorithms	No IP-level MBIST or integration challenges considered	March Y: 2562 cycles, March C-: 4105 cycles

Key Contributions	Aspect	Identified Gaps
Used linear regression, decision trees [5, 6, 36]	**Model Complexity**	No ensemble models. Poor at capturing complex patterns
Used memory size, depth, and clock domains [5, 7, 11, 37]	**Feature Engineering**	Missed algorithm types and feature interactions
Classified memory blocks manually or skipped signal types [5, 6, 9, 38]	**Signal Classification**	No RTL-level port tagging or bit-width detection
Focused on predicting MBIST area and test time [5, 7]	**Optimisation Focus**	No attempt to optimise or improve results after prediction
Applied MBIST at SoC level [5, 7, 10]	**Integration Level**	IP-level MBIST insertion rarely explored
Evaluated < 40 SRAMs [5, 7]	**Memory Coverage**	Limited scalability. No testing on large, mixed memory sets
Selected memory-related inputs [5, 7, 10]	**Feature Selection**	Ignored logical ports that affect test cycles

Gantt Chart (FYP 1)

Table 2: Detailed Gantt chart of FYP 1.



Gantt Chart (FYP 2)

Table 3: Detailed Gantt chart of FYP 2.

Activities	Week															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Machine Learning Model Training																
Advanced Feature Engineering Implementation																
Ensemble Model Design and Deployment																
FYP 2 Briefing																
Model Prediction Validation																
Ablation Study Design and Comparison																
Conference Paper Preparation and Submission																
Overall Results Analysis																
Model Migration to RTL																
Script Documentation																
FYP 2 Presentation Preparation																
Report Writing																
FYP 2 Report Submission																

System Architecture

Architectural flow of the proposed ML-based MBIST estimation system

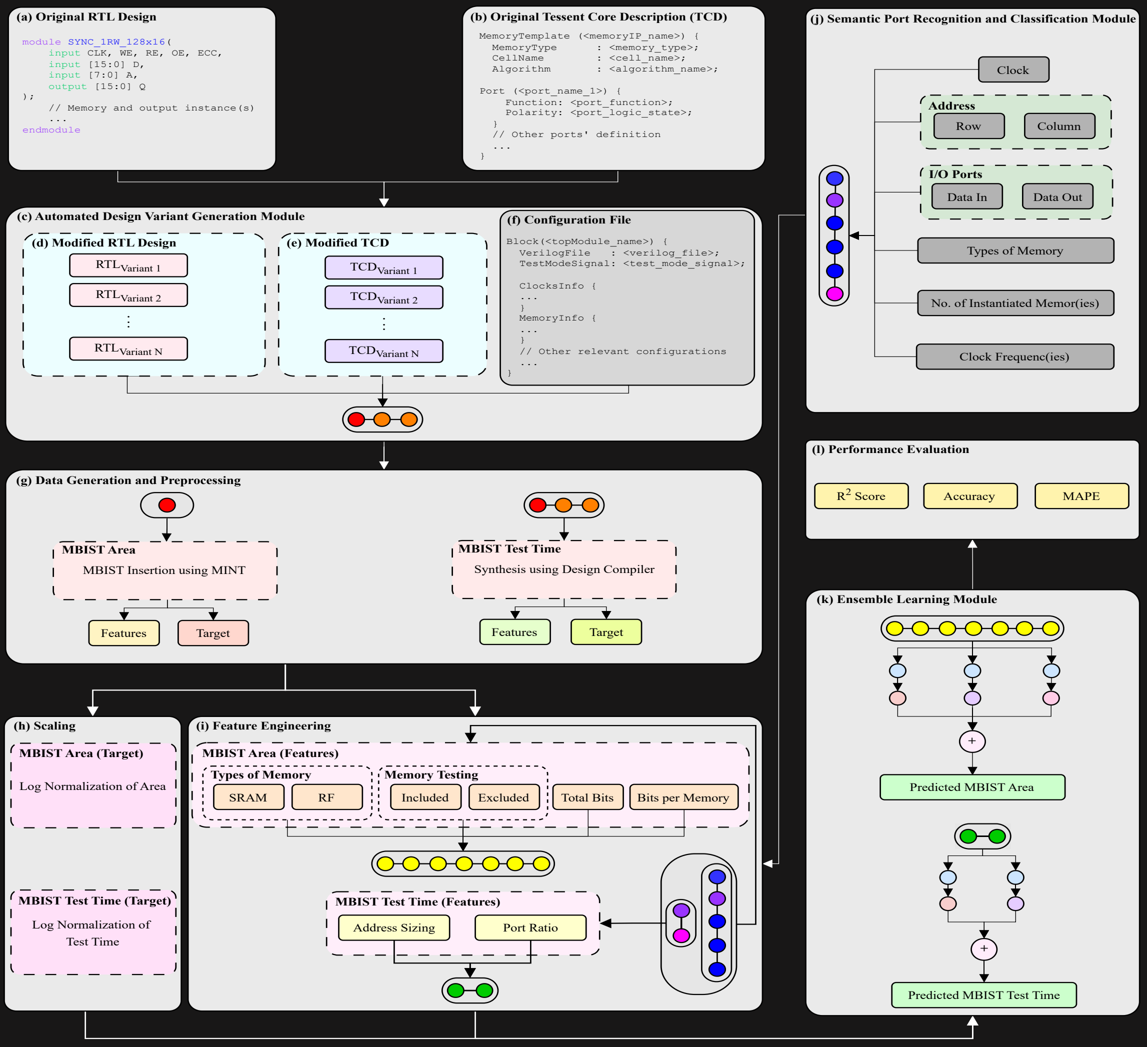


Figure 4: Proposed framework for MBIST prediction overview.

Research Methodology (Part 1)

From RTL design to feature-rich dataset for MBIST estimation

Inserted

MBIST at RTL designs using Tessent MBIST [8, 9, 27]

Generated

over 5,000 variants by changing memory configs, ports, clocks and more

Extracted

area using Synopsys Design Compiler and test time using Tessent MBIST [9, 27]

Identified

essential features for both MBIST area and test time to predict target(s) [14, 31]

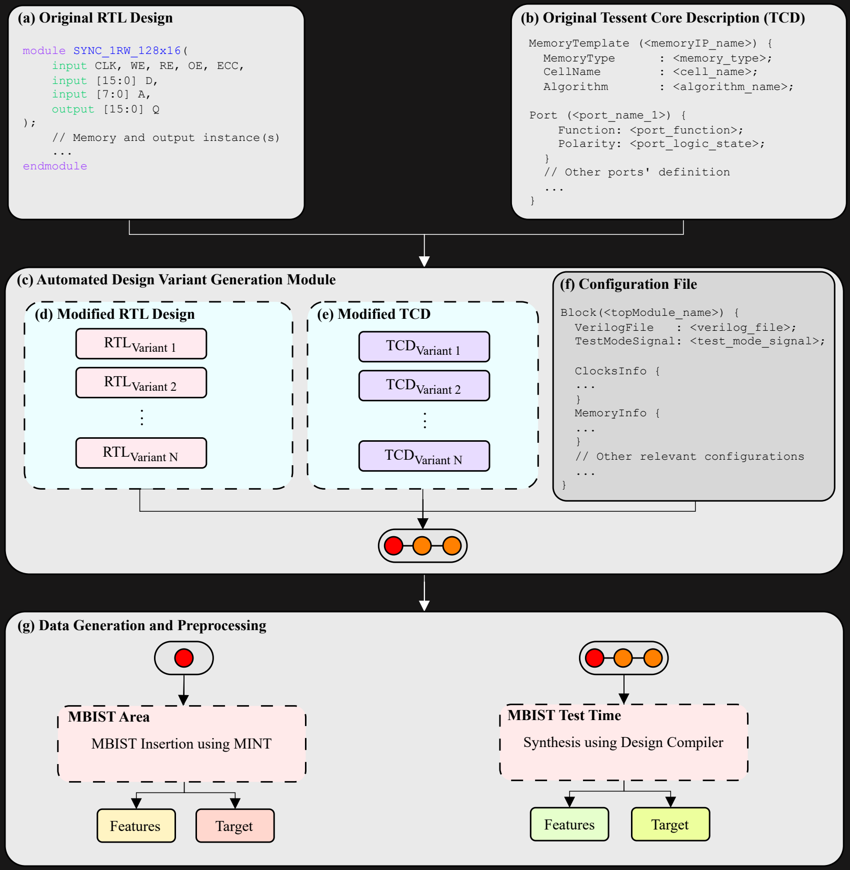


Figure 5: Proposed framework for dataset generation.

Research Methodology (Part 2)

Ensemble machine learning pipeline for accurate and scalable prediction

Reengineered

the raw features to new features such as included bit usage, address sizing, port ratios

Normalised

both MBIST area and test time to improve learning

Optimised

hyperparameter tuning using Optuna (100 trials) [30, 63]

Developed

stacked ensemble model for area: XGBoost + LightGBM + Neural Network → Gradient Boosting (meta) [14, 17, 47]

stacked ensemble model for test time: XGBoost + LightGBM → Ridge Regression (meta) [30, 63]

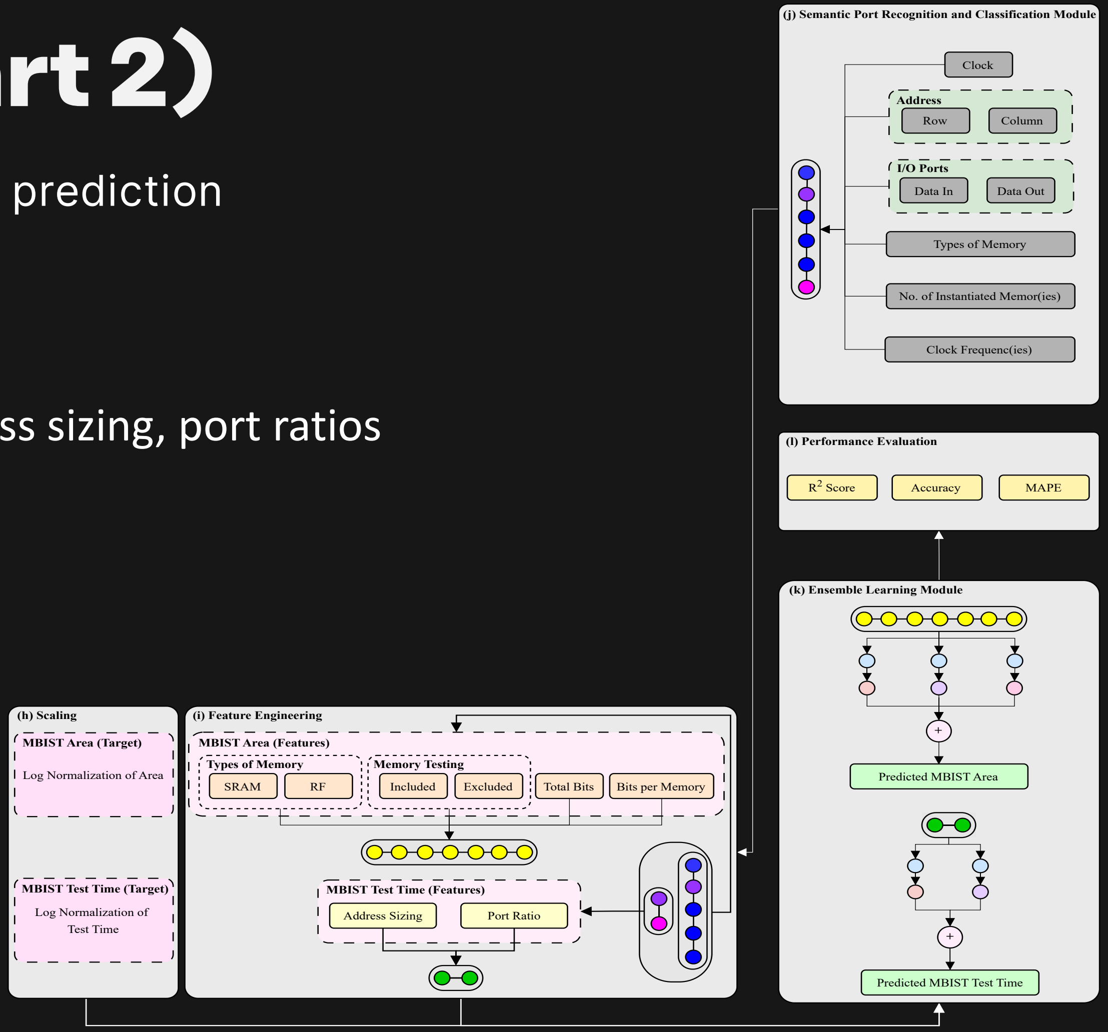


Figure 6: Proposed framework for ensemble pipeline with feature engineering.

Dataset Generation for MBIST Prediction

Creating > 5,000 memory IP variants through automated RTL and MBIST flows

4,470
dataset for area prediction

624
dataset for test time prediction

Tool Flow

Makefile automation → Tessent (MBIST insertion) → Design Compiler (synthesis) → Python (structured CSV with raw features)

Target / Output

$$MBIST\ Area = Modified\ Design\ Area - Original\ Design\ Area$$

$$Test\ Time = Clock\ Period \times Test\ Cycles = \frac{Test\ Cycles}{Clock\ Rate}$$

Table 4: Raw features and dataset range for MBIST metrics prediction.

Feature	Description	Range
MBIST area engineered features		
SRAM	Number of embedded SRAM instances	0 – 80
RF	Number of register file (RF) instances	0 – 80
Total Memories	Total number of memories (SRAM + RF + excluded memories)	2 – 103
Clock Domains	Number of clock domains connected to memory blocks	1 – 3
Address Depth	Bit-width of address used to access memory	64 – 512
Data Width	Bit-width of data bus used for read/write access	8 – 32
MBIST test time engineered features		
Row Count	Bit-width of row number in the memory library	2 – 4096
Column Count	Bit-width of column number in the memory library	2 – 16384
Write Port	Total number of memory write ports	1 – 2
Read Port	Total number of memory read ports	1 – 2

Feature Engineering and Input Transformation

Enriching raw RTL features to boost MBIST prediction accuracy

Extracted

raw design attributes from RTL

Applied

polynomial and logarithmic transformations

Normalised

targets using logarithm with z-score scaling

Outcome?

Domain-specific transformations led to better convergence, reduced overfitting, and improved generalisation

Table 5: Engineered features for MBIST metrics prediction.

Feature	Description
MBIST area engineered features	
Included	Indicates whether a memory is configured for MBIST insertion
Included Ratio	Ratio of included memories to total memory instances
Included per Clock Domain	Average number of included memories per clock domain
Memory Type	Encoded memory type (0 for SRAM, 1 for Register File)
Total Bits	Total memory capacity calculated as depth multiplied by width
Excluded Ratio	Ratio of excluded memories within the design
Bits per Memory	Average bit size per memory instance
Log Area	Log-transformed value of area to stabilise variance
MBIST test time engineered features	
Address Sizing	Estimated number of addressable memory cells calculated from row and column
Port Ratio	Ratio of write to read ports with a constant added to maintain stability
Log Test Cycles	Log-transformed value of test cycles used as the prediction target

Proposed Stacked Ensemble Architecture

Leveraging hybrid learning models for accurate and scalable MBIST estimation

Area	Aspect	Test Time
Included Included Ratio Included per Clock Domain Memory Type Total Bits Excluded Ratio Bits per Memory	**Feature**	Address Sizing Port Ratio
XGBoost LightGBM Neural Network	**Base Models**	XGBoost LightGBM
Gradient Boosting Regressor	**Meta Learner**	Ridge Regression
Adopted from existing works	**Hyperparameter Tuning**	Optuna search (100 trials)

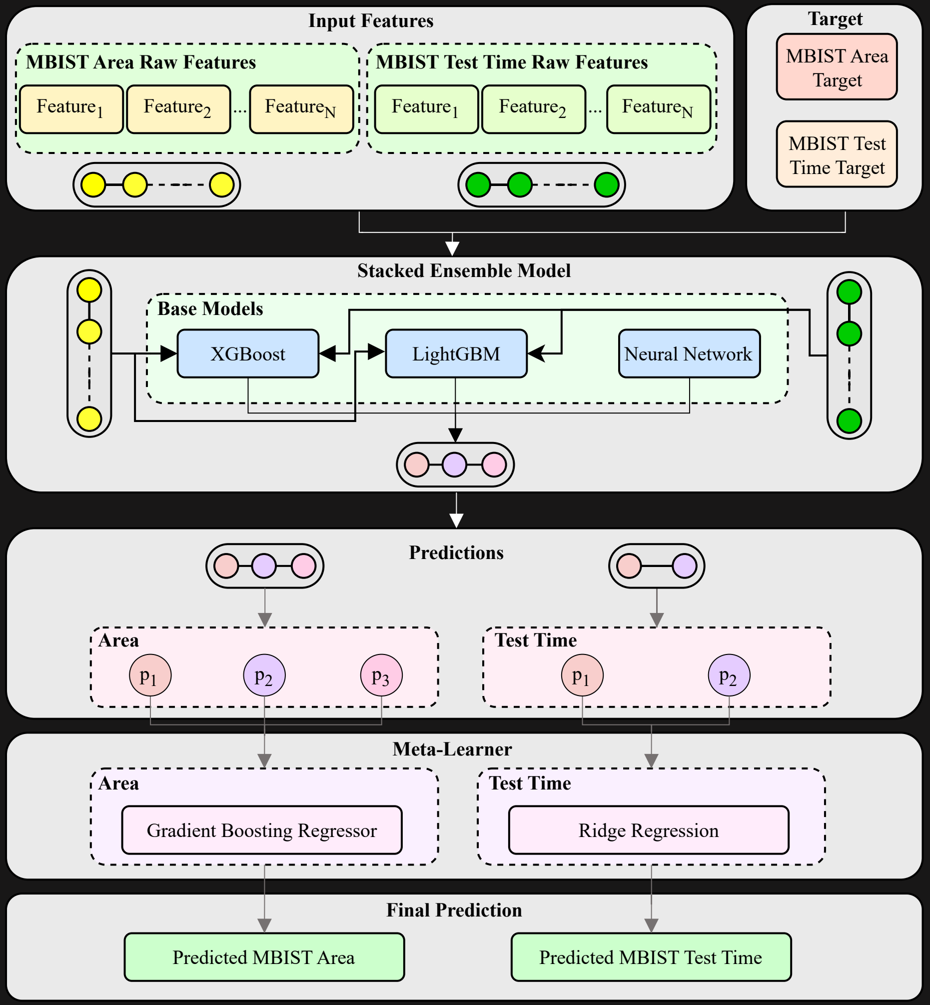


Figure 7: Illustration of the proposed stacked ensemble architecture.

Proposed Algorithms

Leveraging hybrid learning models for accurate and scalable MBIST estimation

Table 6: Algorithm of the proposed stacked ensemble learning for MBIST area prediction.

Algorithm 1 Proposed stacked ensemble learning for area prediction

Input: Dataset = $\{X, y\}$ with engineered features, y log-scaled
polynomial degree $p = 2$;
base learners $M_1(XGBoost), M_2(LightGBM), M_3(NN)$;
meta-learner M_{meta} (Gradient Boosting Regressor);
Convergence threshold ε
Output: Trained meta-learner M_{meta}
$\tilde{X} \leftarrow$ poly_expand(X, p)
$\tilde{X} \leftarrow$ standardise$(\tilde{X})$
for $M_i \in \{M_1, M_2, M_3\}$ do
    Train M_i on $(\tilde{X}, y)$
Z $\leftarrow [M_1(\tilde{X}), M_2(\tilde{X}), M_3(\tilde{X})]$
$\text{err}_{\text{prev}} \leftarrow \infty$
repeat
    Train M_{meta} on (Z, y)
    $\hat{y} \leftarrow M_{meta}(Z)$
    $\text{err}_{\text{curr}} \leftarrow MAPE(y, \hat{y})$
    if $|\text{err}_{\text{prev}}\_\text{err}_{\text{curr}}| < \varepsilon$ then
        break
    else
        Adjust hyperparameters of M_{meta}
        $\text{err}_{\text{prev}} \leftarrow \text{err}_{\text{curr}}$
until convergence return M_{meta}

Table 7: Algorithm of the proposed stacked ensemble learning for MBIST test time prediction.

Algorithm 2 Proposed stacked ensemble learning for test time prediction

Input: Dataset = $\{X, y\}$ with engineered features, y log-scaled
base learners $M_1(XGBoost), M_2(LightGBM)$;
meta-learner M_{meta} (Ridge Regression);
Optuna search with $T = 100$ trials; convergence threshold $\mathcal{E}$
Output: Optimised meta-learner M_{meta}
$X \leftarrow$ standardise (X, p)
$M_1 \leftarrow$ optuna_tune$(XGBoost, X, y, T)$
Train M_2 on (X, y)
Z$\leftarrow [M_1(X), M_2(X))]$
$\text{err}_{\text{prev}} \leftarrow \infty$
repeat
    Train M_{meta} on (Z, y)
    $\hat{y} \leftarrow M_{meta}(Z)$
    $\text{err}_{\text{curr}} \leftarrow MAPE(y, \hat{y})$
    if $|\text{err}_{\text{prev}}\_\text{err}_{\text{curr}}| < \varepsilon$ then
        break
    else
        Adjust hyperparameters of M_{meta}
        $\text{err}_{\text{prev}} \leftarrow \text{err}_{\text{curr}}$
until convergence return M_{meta}

Results and Discussion

Results show significant improvement in MBIST estimation using proposed ML

Hyperparameter Tuning

Refining ensemble models using Optuna to maximise MBIST prediction accuracy

Table 10: Summary of tuned hyperparameters used in stacked ensemble model for MBIST area prediction.

Hyperparameter	XGBoost	LightGBM	Neural Network
Max Depth	7	7	N/A
Learning Rate	0.0135	0.0130	0.0015
N Estimators	850	850	N/A
Subsample	0.92	0.92	N/A
Colsample Bytree	0.87	0.87	N/A
Reg Alpha	0.15	0.15	N/A
Reg Lambda	0.35	0.35	N/A
Dropout Rate	N/A	N/A	0.3 / 0.2
Batch Size	N/A	N/A	128

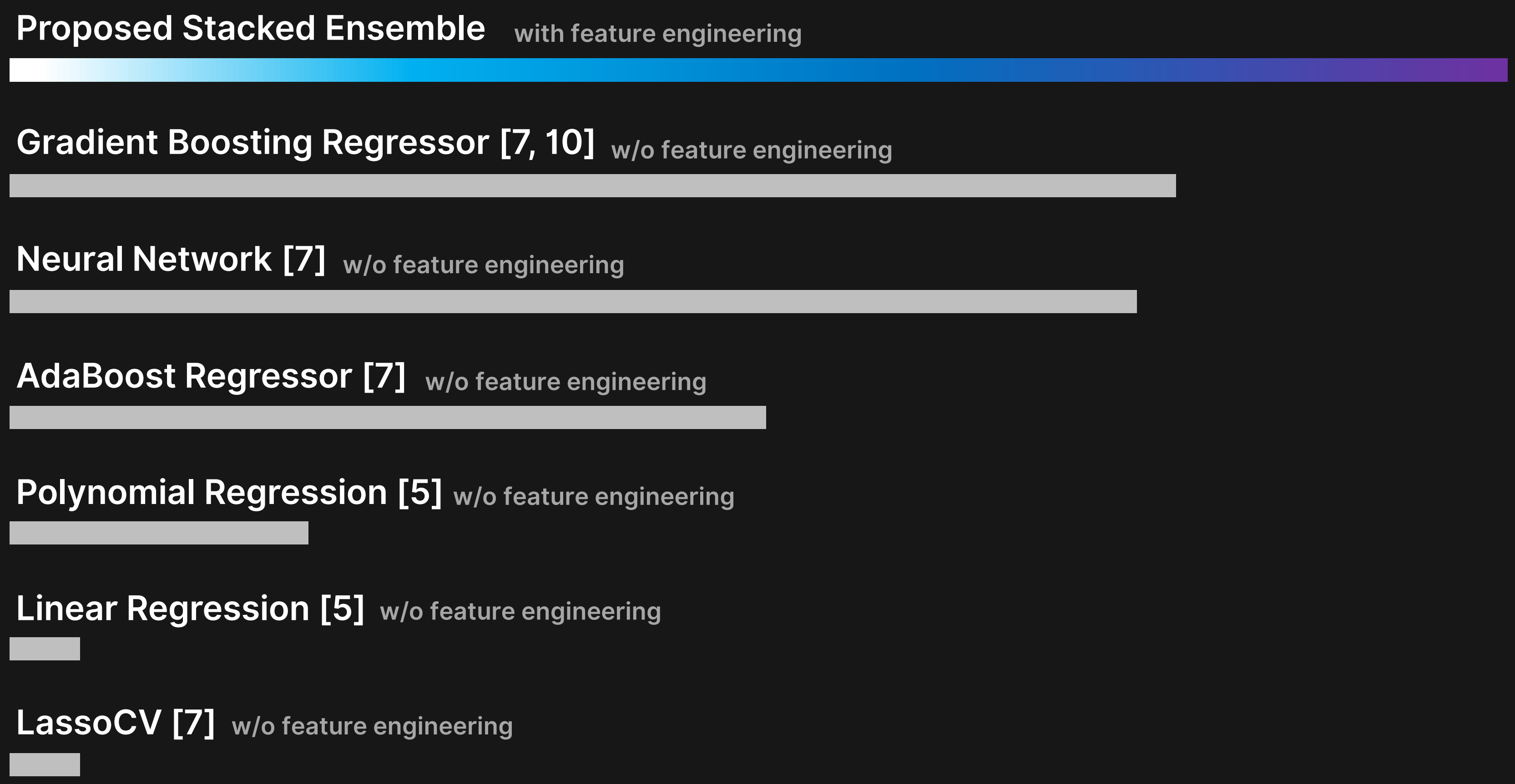
Table 11: Summary of tuned hyperparameters used in stacked ensemble model for MBIST test time prediction.

Hyperparameter	XGBoost	LightGBM
Max Depth	4	10
Learning Rate	0.0278	0.015
N Estimators	719	1500
Subsample	0.65	0.80
Colsample Bytree	0.85	0.85
Gamma	1.083e-4	N/A
Reg Alpha	7.397e-4	0.05
Reg Lambda	1.40e-4	0.5

Optuna search (100-trials) + **Fine**-tuned parameters + **Meta**-learner = **BETTER Generalisation**

MBIST Area Prediction Accuracy

Feature engineering significantly improves area prediction performance



Up to

21.2x

more accurate than the models WITHOUT feature engineering

At least

8.48%

better than the existing models WITH feature engineering

MBIST Area Prediction Accuracy

Feature engineering significantly improves area prediction performance

Table 8: Comparison of MBIST area prediction metrics with and without feature engineering.

Feature Engineering	Model	Training Set			Test Set		
		R^2 *Score*	*Accuracy* (%)	*MAPE* (%)	R^2 *Score*	*Accuracy* (%)	*MAPE* (%)
No	LR [5]	0.4074	6.10	2979.99	0.3756	4.26	2871.76
	PolyReg [5]	0.8254	19.41	1973.07	0.7927	18.07	2066.40
	LassoCV [7]	0.4074	6.12	2982.90	0.3757	4.26	2874.53
	AdaBoostReg [7]	0.9886	46.97	137.83	0.9882	45.79	137.44
	GBR [7, 10]	0.9999	71.32	22.65	0.9998	70.59	21.07
	NN [7]	0.9943	68.06	20.69	0.9947	68.24	16.70
	Proposed Work	0.9971	75.64	73.33	0.9946	69.13	121.00
Yes	LR [5]	0.9854	51.83	36.62	0.9875	50.51	34.65
	PolyReg [5]	0.9289	46.74	17.31	0.9327	45.90	15.78
	LassoCV [7]	0.8855	23.40	2319.90	0.8740	21.21	2190.96
	AdaBoostReg [7]	0.9887	47.39	117.50	0.9884	46.35	108.76
	GBR [7, 10]	0.9994	80.25	20.07	0.9994	79.91	22.18
	NN [7]	0.9990	83.43	10.37	0.9992	82.15	8.95
	Proposed Work	1.0000	91.60	5.54	0.9999	90.68	5.14

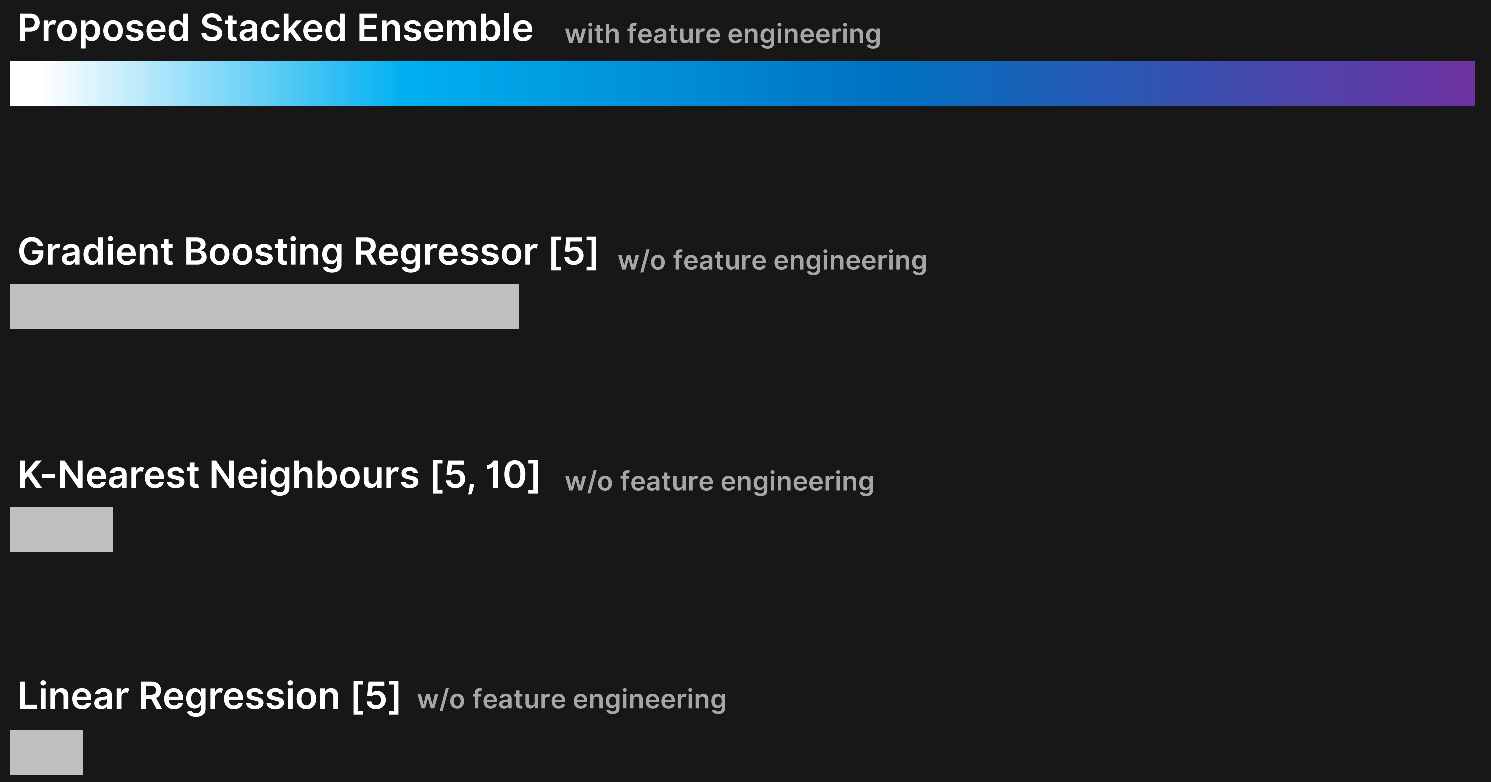
Excellent
model fit

High
prediction confidence

Low
relative error

MBIST Test Time Prediction Accuracy

Feature engineering and model stacking drastically reduce test time error



Up to

20x

more accurate than the models WITHOUT feature engineering

At least

48.8%

better than the existing models WITH feature engineering

MBIST Test Time Prediction Accuracy

Feature engineering and model stacking drastically reduce test time error

Table 9: Comparison of MBIST test time prediction metrics with and without feature engineering.

Feature Engineering	Model	Training Set			Test Set		
		R^2 *Score*	*Accuracy* (%)	*MAPE* (%)	R^2 *Score*	*Accuracy* (%)	*MAPE* (%)
No	LR [5]	0.2761	4.68	2795.41	0.1302	6.78	1909.91
	GBR [5]	0.8327	39.48	16.49	0.7818	33.60	19.40
	KNN [5, 10]	0.4412	4.80	88.15	0.4412	4.80	88.15
	Proposed Work	0.9936	35.87	532.88	0.9360	22.40	788.14
Yes	LR [5]	0.9053	7.89	2062.17	0.8989	7.02	1138.62
	GBR [5]	0.9520	50.50	12.21	0.9095	48.00	14.18
	KNN [5, 10]	0.9072	19.84	23.76	0.8315	19.20	35.26
	Proposed Work	0.9997	100.00	1.06	0.9978	96.80	2.10

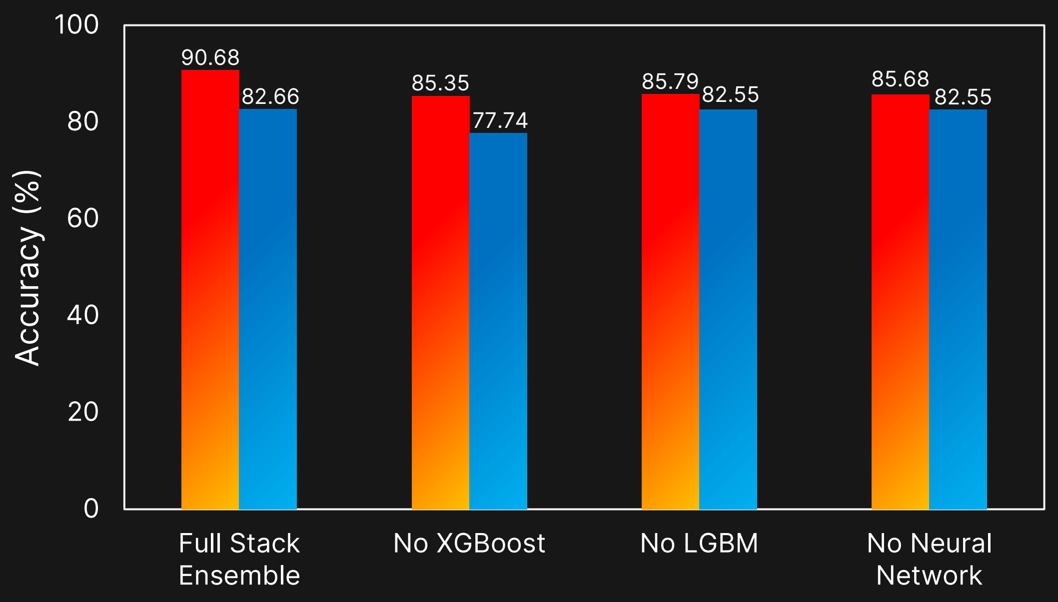
Robust
model fit

Reliable
prediction confidence

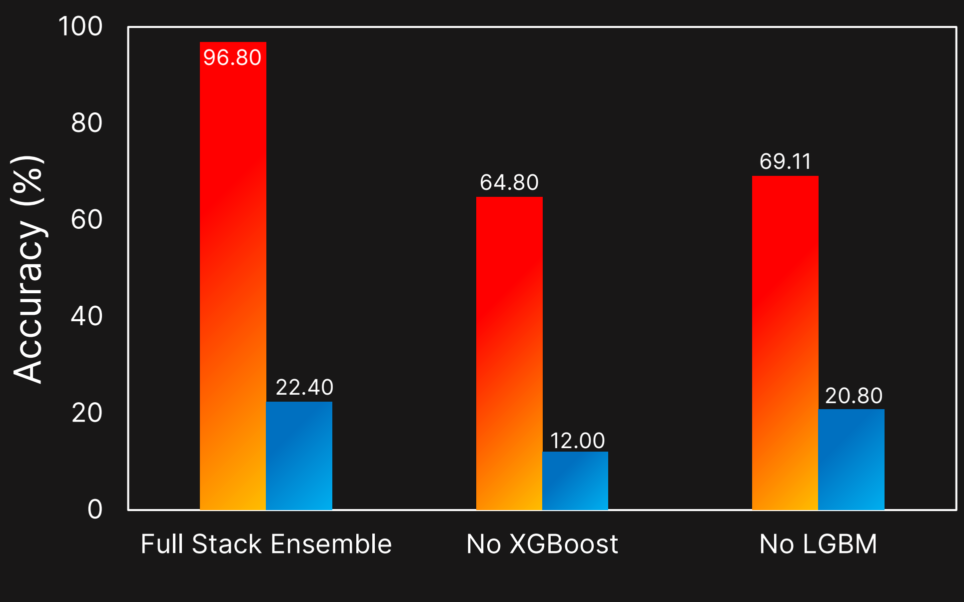
Reduced
relative error

Ablation Study – Model Component Impact

Evaluating the role of ensemble learners and feature engineering in prediction performance



(a)



(b)

Figure 8: Accuracy comparison of (a) area and (b) test time on ablation study.

Improved

learning capability via stacked ensemble

Each base learner (XGBoost, LightGBM, NN) contributes unique strengths, i.e., removing one lowers accuracy.

Critical

domain-specific feature engineering

Lack of feature engineering leads to the largest drop in performance, especially for test time.

Conclusion

Stacked ensemble and feature engineering deliver accurate MBIST estimation from RTL

Full Stack = Best Generalisation

Combining all learners with engineered features gives highest accuracy across both MBIST metrics.

(a) (b) (c)

Figure 9: Summary of performance comparison of (a) area, (b) test time and (c) ablation model.

90.68% accuracy achieved to predict MBIST area overhead

96.80% accuracy achieved to predict MBIST test time

Developed a stacked ensemble model (XGBoost + LightGBM + Neural Network) with advanced domain-specific feature engineering

Eliminated the need for full synthesis during early-stage design

Enabled fast, accurate MBIST estimation directly from RTL

Thank You

“ If complex IPs can be accurately predicted, then our future shouldn’t just be a guess. ”

- Chee Jin (2025)

QnA Session

“ True optimisation isn’t solely about reducing area and time — ”

it’s about conserving energy for what truly matters in life.

- Chee Jin (2025)

References

[1] M. V. J. Kinseher, I. Polian, "Improving testability and reliability of advanced SRAM architectures," *IEEE Transactions on Emerging Topics in Computing,* vol. 7, pp. 456-467, 2017.

[2] T. Q. Bui, L. D. Pham, H. M. Nguyen, V. T. Nguyen, T. C. Le, and T. Hoang, "An effective architecture of memory built-in self-test for wide range of SRAM," presented at the 2016 International Conference on Advanced Computing and Applications (ACOMP), 2016.

[3] T. C. Jin, "Stacked ensemble learning-enabled early prediction of MBIST area and test time in complex memory IPs," Universiti Teknologi Malaysia, 2025.

[4] S. Zhang, Z. Wang, and Y. Wang, "The MBIST implementation of SoC chip," presented at the 2024 9th International Conference on Integrated Circuits and Microsystems (ICICM), 2024.

[5] D. Jamal and R. Veetil, "Efficient MBIST area and test time estimator using machine learning technique," presented at the 2023 36th International Conference on VLSI Design and 2023 22nd International Conference on Embedded Systems (VLSID), 2023.

[6] T. McLaurin and R. Knoth, "The challenges of implementing an MBIST interface: A practical application," presented at the 2019 IEEE International Test Conference (ITC), 2019.

[7] P. Arora, V. Jain, T. Goyal, and N. Card, "Machine learning based MBIST area estimation," presented at the 2023 IEEE International Test Conference India (ITC India), 2023.

[8] S. Inc., *Tessent ™MemoryBIST user's manual for use with Tessent Shell*. Siemens Inc., 2023, p. 912.

[9] S. Ramesh, M. Junxia, J. Sharada, and K. Neelima, *Memory-BIST Insertion, Validation and Test Generation Environment (MINT): MINT External Product Specification*. 2024, p. 521.

[10] D. Jamal and R. T. Veetil, "MBIST area & test time optimization using machine learning," presented at the 2023 IEEE International Test Conference India (ITC India), 2023.

[11] A. Z. Jidin, R. Hussin, W. F. Lee, and M. S. Mispan, "MBIST implementation and evaluation in FPGA based on low-complexity March algorithms," *Journal of Circuits, Systems and Computers,* vol. 33, no. 08, p. 2450152, 2024.

[12] B. Singh, A. Khosla, and S. B. Narang, "Area overhead and power analysis of March algorithms for memory BIST," presented at the International Conference on Communication Technology and System Design 2011, 2012.

[13] S. Raschka and V. Mirjalili, *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt publishing ltd, 2019.

[14] V. Kore and V. Khadse, "Progressive heterogeneous ensemble learning for cancer gene expression classification," presented at the 2022 International Conference on Machine Learning, Computer Systems and Security (MLCSS), 2022, 149.

[15] S. Ning, K. Nguyen, S. Bagchi, and Y. Park, "The study of feature engineering in machine learning and deep learning for network intrusion detection systems," presented at the 2024 Silicon Valley Cybersecurity Conference (SVCC), 2024.

[16] M. Jimenez, C. Maxime, Y. L. Traon, and M. Papadakis, "On the impact of tokenizer and parameters on n-gram based code analysis," presented at the 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2018, 437.

[17] T. G. Jang, S. Y. Park, J. A. Lee, H. Y. Park, T. Dorjmaa, and D. Kang, "Ensemble learning approaches for automatic detection of chronic kidney disease stages during sleep," presented at the 2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2024.

[18] N. Ramteke and R. Maidamwar, "Cardiac patient data classification using ensemble machine learning technique," presented at the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2023.

[19] A. Fatima, R. Maurya, M. K. Dutta, R. Burget, and J. Masek, "Android malware detection using genetic algorithm based optimized feature selection and machine learning," *2019 42nd International Conference on Telecommunications and Signal Processing (TSP),* p. 4, 2019.

[20] M. Ashour, C. Salama, and M. W. El-Kharashi, "Detecting spam tweets using character n-gram features," presented at the 2018 13th International Conference on Computer Engineering and Systems (ICCES), 2018, 190.

[21] R. Yang, Z. Wang, and M. Shen, "An efficient grouping method for large-scale MBIST," presented at the 2024 2nd International Symposium of Electronics Design Automation (ISEDA), 2024.

[22] Y. J. M. Shirur, H. Lakshmi, and V. S. Chakravarthi, "Implementation of area efficient hybrid MBIST for memory clusters in asynchronous SoC," presented at the 2014 Fifth International Symposium on Electronic System Design, 2014.

[23] K. H. Ng, N. E. Alias, A. Hamzah, M. L. P. Tan, U. U. Sheikh, and Y. A. Wahab, "A March 5n FSM-based Memory Built-In Self-Test (MBIST) architecture with diagnosis capabilities," presented at the 2022 IEEE International Conference on Semiconductor Electronics (ICSE), 2022.

[24] T. N. Kong *et al.*, "An efficient March (5N) FSM-based memory built-in self test (MBIST) architecture," presented at the 2021 IEEE Regional Symposium on Micro and Nanoelectronics (RSM), 2021.

[25] K. Sun, Y. Shen, Y. Lao, Z. Zhang, X. You, and C. Zhang, "A new error correction scheme for physical unclonable function," presented at the 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2018.

[26] N. Sridevi, K. Jamal, and K. Mannem, "Implementation of error correction techniques in memory applications," presented at the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021.

[27] S. SolvNetPlus, *Design Compiler user guide*. Synopsys Inc., 2024, p. 905.

[28] G. Biagetti, P. Crippa, L. Falaschetti, and C. Turchetti, "Machine learning regression based on particle Bernstein polynomials for nonlinear system identification," presented at the 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP), 2017.

[29] D. Maulud and A. M. Abdulazeez, "A review on linear regression comprehensive in machine learning," *Journal of Applied Science and Technology Trends,* vol. 1, no. 2, pp. 140-147, 2020.

[30] S. Kumari, V. C. Ranganayaki, K. Sathyamoorthy, K. Selvi, T. A. Mohanaprakash, and C. Tamilselvi, "Optuna-optimized machine learning technique for accurate diabetes prediction and classification," presented at the 2024 4th International Conference on Sustainable Expert Systems (ICSES), Kaski, Nepal, 2024, 1478.

[31] C. C. Tsai and C. C. Chan, "Ensemble and unsupervised machine learning applied on laser ablation quality study of Silicon Nitride during CMOS-MEMS post processing," presented at the 2022 IEEE 4th Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2022.

[32] J. Kinseher, M. Voelker, and I. Polian, "Improving testability and reliability of advanced SRAM architectures," *IEEE Transactions on Emerging Topics in Computing,* vol. 7, no. 3, pp. 456-467, 2017.

[33] L. Massaron and A. Boschetti, *Regression analysis with Python*. Packt Publishing Ltd, 2016.

[34] E. Ilhan, M. Y. Turali, and S. S. Kozat, "Gradient boosting with moving-average terms for nonlinear sequential regression," *IEEE Signal Processing Letters,* vol. 30, p. 5, 2023.

[35] L. Aththanayake, A. Mahmud, N. Hosseinzadeh, and A. Gargoom, "Performance analysis of regression and Artificial Neural Network schemes for dynamic model reduction of power systems," presented at the 2021 3rd International Conference on Smart Power & Internet Energy Systems (SPIES), 2021.

[36] Y. Lu, Y. Zhu, and M. Li, "A MBIST controller based on JTAG interface applied in power line chip," presented at the 2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), 2016.

[37] M. Peng, K. Ouyang, F. Pan, G. Zhou, and L. Chen, "A novel method to achieve high efficient iteration of MBIST pattern," presented at the 2023 China Semiconductor Technology International Conference (CSTIC), 2023.

[38] K. S. Das and P. Prakash, "Automatic MBIST scheduling engine," presented at the 2019 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2019.

[39] J. Kinseher, M. Richter, and I. Polian, "On the automated verification of user-defined MBIST algorithms," presented at the ZuE 2015; 8. GMM/ITG/GI-Symposium Reliability by Design, Siegen, Germany, 2015.

[40] N. Maneshinde, P. Hegade, R. Mittal, N. Palecha, and M. Suma, "Programmable FSM based built-in-self-test for memory," presented at the 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2016.

[41] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, "Machine learning basics," *Deep learning,* pp. 98-164, 2016.

[42] S. Ray, "A quick review of machine learning algorithms," presented at the 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon), 2019.

[43] C.-S. M. Wu, P. Patil, and S. Gunaseelan, "Comparison of different machine learning algorithms for multiple regression on black friday sales data," presented at the 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), 2018.

[44] V. Nasteski, "An overview of the supervised machine learning methods," presented at the 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), 2017.

[45] W.-M. Lee, *Python machine learning*. John Wiley & Sons, 2019.

[46] M. A. Saleem, A. Javeed, W. Benjapolakul, W. Srisiri, S. Chaitusaney, and P. Kaewplung, "Neural-XGBoost: A hybrid approach for disaster prediction and management using machine learning," *IEEE Access,* vol. 13, p. 13, 2025, doi: 10.1109/ACCESS.2025.3569499.

[47] Z. M. Omer and H. Shareef, "Adaptive boosting and bootstrapped aggregation based ensemble machine learning methods for photovoltaic systems output current prediction," presented at the 2019 29th Australasian Universities Power Engineering Conference (AUPEC), Nadi, Fiji, 2019.

[48] W. E. Elhola and K. Bozed, "Performance analysis of brain tumor detection based on Gradient Boosting Machine - CNN model," presented at the 2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), Sabratha, Libya, 2022.

[49] N. Kumar and A. Agarwal, "Real-time modelling and controlling of boost converter using supervised learning," presented at the 2022 IEEE 10th Power India International Conference (PIICON), New Delhi, India, 2022.

[50] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review,* vol. 54, p. 31, 2021.

[51] M. Babu, M. Arun, S. Pushparani, A. Muthukrishnan, and N. Poongavanam, "Exploring quantum machine learning algorithms for enhanced data classification and clustering in complex systems - Integration with Gradient Boosting and K-means for improved performance," presented at the 2024 International Conference on Cybernation and Computation (CYBERCOM), Dehradun, India, 2024.

[52] M. Xu, P. Baraldi, X. Lu, and E. Zio, "Generative adversarial networks with AdaBoost ensemble learning for anomaly detection in high-speed train automatic doors," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 12, p. 14, 2022.

[53] S. Sankar, A. Potti, G. N. Chandrika, and S. Ramasubbareddy, "Thyroid disease prediction using XGBoost algorithms," *Journal of Mobile Multimedia,* vol. 18, no. 3, p. 17, 2022.

[54] Y. Zhou, X. Son, and M. Zhou, "Supply chain fraud prediction based on XGBoost method," presented at the 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Nanchang, China, 2021.

[55] GeeksforGeeks. "Boosting in machine learning | Boosting and AdaBoost." https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/ (accessed 27 May 2025, 2025).

[56] K. Wanjari and P. Verma, "A review on the applications of machine learning and deep learning algorithms for image recognition," presented at the 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL), Bhimdatta, Nepal, 2025.

[57] I. Korkmaz and F. Soygazi, "Gastrointestinal image classification using deep learning architectures via transfer learning," presented at the 2024 Medical Technologies Congress (TIPTEKNO), Mugla, Turkiye, 2024.

[58] J. S. Varela, P. Almasan, M. F. Galmés, K. Rusek, F. Geyer, and X. Cheng. (2023) Graph Neural Networks for communication networks: Context, use cses and oportunities. *IEEE Network*. 8.

[59] Manika. "8 deep learning architectures data scientists must master." https://www.projectpro.io/article/deep-learning-architectures/996 (accessed 25 May 2025, 2025).

[60] A. Pandit, A. Gupta, M. Bhatia, and S. C. Gupta, "Filter based feature selection anticipation of automobile price prediction in Azure machine learning," presented at the 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), Faridabad, India, 2022.

[61] N. Zhantileuov and S. Ospanov, "A comparative study of supervised machine learning and deep learning techniques with feature selection methods for classifying Parkinson's disease based on speech impairments," presented at the 2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 2024.

[62] S. M. Abdelouahed, R. Abla, E. Asmae, and A. Abdellah, "Harnessing feature engineering to improve machine learning: A review of different data processing techniques," presented at the 2024 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 2024.

[63] H. Li, "A SoC estimation method for lithium-ion batteries based on Optuna optimized long short-term memory Neural Network," presented at the 2024 4th International Conference on Electronic Information Engineering and Computer Communication (EIECC), Wuhan, China, 2024, 26.

[64] Q. Song, X. Cheng, and P. Cao, "Critical paths prediction under multiple corners based on BiLSTM network," presented at the 2023 60th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2023.

[65] S. Jose, A. T. Maliackal, A. Sukumaran, J. Cyriac, and R. M. Varkey, "Machine learning technique to analyze the health condition of athletes and predict their performance," presented at the 2023 International Conference on Circuit Power and Computing Technologies (ICCPCT), Kollam, India, 2023.

[66] A. A. Masud, S. Hossain, M. Rifa, F. Akter, A. Zaman, and D. M. Farid, "Meta-learning in supervised machine learning," presented at the 2022 14th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Phnom Penh, Cambodia, 2022

[67] T. Hartmann, A. Moawad, C. Schockaert, F. Fouquet, and Y. L. Traon, "Meta-Modelling Meta-Learning," presented at the 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS), Munich, Germany, 2019, 6.

[68] A. Doke and M. Gaikwad, "Survey on automated machine learning (AutoML) and meta learning," presented at the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021.

[69] X. Liu, Y. Deng, A. Nallanathan, and M. Bennis, "Federated learning and meta learning: Approaches, applications, and directions," *IEEE Communications Surveys & Tutorials,* vol. 26, no. 1, p. 48, 2023, doi: https://doi.org/10.1109/COMST.2023.3330910.

[70] Y. Lai, "Application and effectiveness evaluation of Bayesian optimization algorithm in hyperparameter tuning of machine learning models," presented at the 2024 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC), Athens, Greece, 2024.

[71] Monica and P. Agrawal, "A Survey on Hyperparameter Optimization of Machine Learning Models," presented at the 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2024.

[72] K. M. Sujon, R. Hassan, Z. T. Towshi, M. A. Othman, M. A. Samad, and K. Choi, "When to use standardization and normalization: Empirical evidence from machine learning models and XAI," *IEEE Access,* vol. 12, p. 15, 2024, doi: https://doi.org/10.1109/ACCESS.2024.3462434.

[73] S. SolvNetPlus, *VCS® User Guide*. Synopsys Inc., 2023, p. 2367.

[74] S. SolvNetPlus, *Verdi® User uide and Tutorial*. Synopsys Inc., 2023, p. 571.

[75] K. Choi, S. Kim, J. Kim, and I. C. Park, "Hardware-friendly approximation for Swish activation and its implementation," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 71, no. 10, p. 5, 2024, doi: https://doi.org/10.1109/TCSII.2024.3394806.

[76] H. Moayed and E. G. Mansoori, "Skipout: An adaptive layer-level regularization framework for deep Neural Networks," *IEEE Access,* vol. 10, p. 11, 2022, doi: https://doi.org/10.1109/ACCESS.2022.3178091.

[77] X. Yang and W. Wen, "Ridge and Lasso regression models for cross-version defect prediction," *IEEE Transactions on Reliability,* vol. 67, no. 3, p. 12, 2018, doi: https://doi.org/10.1109/TR.2018.2847353.

[78] L. Laporte, R. Flamary, S. Canu, S. Déjean, and J. Mothe, "Nonconvex regularizations for feature selection in ranking with sparse SVM," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 25, no. 6, p. 13, 2014, doi: https://doi.org/10.1109/TNNLS.2013.2286696.

[79] P. K. Sahu and T. Fatma, "Optimized breast cancer classification using PCA-Lasso feature selection and ensemble learning strategies with Optuna optimization," *IEEE Access,* vol. 13, p. 17, 2025, doi: https://doi.org/10.1109/ACCESS.2025.3539746.

[80] H. Wang, J. He, X. Zhang, and S. Liu, "A short text classification method based on n-gram and CNN," *Chinese Journal of Electronics,* vol. 29, no. 2, p. 7, 2020, doi: https://doi.org/10.1049/cje.2020.01.001.

[81] H. Tsai, S. Ambrogio, P. Narayanan, R. M. Shelby, and G. W. Burr, "Recent progress in analog memory-based accelerators for deep learning," *Journal of Physics D: Applied Physics,* vol. 51, no. 28, p. 283001, 2018.

[82] R. C. Lozoya, B. Berte, H. Cochet, P. Jaïs, N. Ayache, and M. Sermesant, "Model-based feature augmentation for cardiac ablation target learning from images," *IEEE Transactions on Biomedical Engineering,* vol. 66, no. 1, p. 11, 2018, doi: https://doi.org/10.1109/TBME.2018.2818300.

[83] J. Zhao, X. Huang, Y. Gao, J. Zhang, B. Su, and Z. Dong, "Research on machine learning-based correlation analysis method for power equipment alarms," presented at the 2022 International Conference on Informatics, Networking and Computing (ICINC), Nanjing, China, 2022.