SMU Data Science Review

Volume 5 | Number 3 Article 11

2021

Machine Learning Approach to Stability Analysis of Semiconductor Memory Element

Ravindra Thanniru
Southern Methodist University, t.ravindra.naidu@gmail.com

Gautam Kapila
Southern Methodist University, gkapila@gmail.com

Nibhrat Lohia
Southern Methodist University, nlohia@smu.edu

Machine Learning Approach to Stability Analysis of Semiconductor Memory Element

Ravindra Thanniru[1], Gautam Kapila[1], Nibhrat Lohia[1]
[1] Master of Science in Data Science, Southern Methodist University,
Dallas, TX 75275 USA
{rthanniru, gkapila, nlohia}@smu.edu

Abstract. Memory stability analysis traditionally relied heavily on circuit simulation-based approaches that run Monte Carlo (MC) analysis over various manufacturing and use condition parameters. This paper researches application of Machine Learning approaches for memory element failure analysis which could mimic simulation-like accuracy and minimize the need for engineers to rely heavily on simulators for their validations. Both regressor and classifier algorithms are benchmarked for accuracy and recall scores. A high recall score implies fewer escapes of fails to field and is the metric of choice for comparing algorithm. The paper identifies that recall score in excess of 0.97 can be achieved through stack ensemble and logistic regression-based approaches. The high recall score suggests machine learning based approaches can be used for memory failure rate assessments.

1 Introduction

Semiconductor devices or chipsets have a wide variety of on-chip memory requirements [1]. The rapid adoption of Artificial Intelligence (AI) based systems has fueled the need to develop specialized computing hardware to run machine learning algorithms. These AI chips [2] support very high memory bandwidth [3] to perform Deep Neural Network (DNN) computations efficiently and in a short time. Further, ubiquitous Graphical Processing Units (GPU) have dedicated memory to support large input data sets and do massively parallel floating-point computations [4]. Recently, Cerebras's CS-2 claims to be the world's largest AI chip, with 850,000 AI optimized core and 40Gb of on-chip SRAM (Static Random-Access Memory), a type of volatile memory element [5]. A common thread in all of the above is the ever-increasing reliance on larger amounts of on-chip memory. All of the above makes reliability assessment of memory element an important research topic, with business implications.

Reliability of memory elements primarily refers to the stability of memory elements, i.e., their ability to hold on to stored bits of information. Multiple aspects make reliability assessment critical and very difficult. First, larger memory sizes in miniaturized chips are hard to make and suffer from process variation, i.e., each memory element is slightly different, leading to different electrical properties leading to different stability performance. Second, while memory size is increasing, the number of allowed fails can't increase, leading to stricter specifications on memory failure rate. Evaluating the failure probability of memory elements for a given memory array is very

challenging in simulation space and even harder to validate in actual Silicon or product. Any assessment involving comprehending rare fails is computationally intensive, as it invariably consists in running a large number of simulations. Third, larger memory integration leads to higher power consumption. To keep power consumption in check, low voltage operation is desired, making a memory element less reliable. All of these considerations bring home the need to study and develop techniques for memory reliability or stability analysis.

In the current state of the art, memory reliability assessment is done by adopting circuit simulation-based approaches that run Monte Carlo (MC) analysis over a wide variety of manufacturing and use condition parameters. A typical memory element consists of 6 transistors called 6T SRAM cells [6]. While many different SRAM cell constructions have been proposed, this work focuses only on 6T SRAM cells, referred to as SRAM cells from now onwards. The stability of the SRAM cell depends on the strength of each of the individual transistors constituting the cell. By varying strength of each transistor element, per manufacturing process variation data, stability of the memory cell is evaluated in simulation space using a SPICE (Simulation Program with Integrated Circuit Emphasis) circuit simulator. This is done for a specific use voltage and uses temperature. In a typical use voltage condition, the cell failure rate is expected to be less than 1 in a million. To verify this, millions of process variation vectors are generated, where each vector represents a unique SRAM cell from manufacturing and its electrical performance perspective. Cell simulation is performed for every vector, and a stability metric like Static Noise Margin (SNM) is evaluated. Millions of such simulations help provide an estimate of SRAM failure rate. This is done for a specific use temperature and voltage. However, running millions of MC simulations for single voltage and temperature conditions is computationally intensive and requires expert supervision. It also needs to be redone for every new end application use temperature and operating voltage. Lack of availability of user-facing tools that could generate memory failure probability as a function of user-entered voltage and temperature makes estimating reliability at new use conditions tough and time-consuming.

To speed up memory reliability assessment, preliminary work so far has comprised of varying sampling techniques to capture failure region over process variables in a fewer number of simulations [7][8] or use of a surrogate model in place of SPICE simulations to do failure assessment [9][10][11]. A recent work [12] looked at handling data imbalance in the ML approach to classifying memory elements as stable or unstable. Further, a few papers [13] [14][15] have explored the use of algorithms like SVM and Random Forest in assessing the yield of circuit elements like a buffer and DC-DC converter, but not SRAM.

In this paper, the use of a machine learning-based approach is being proposed to assess the SRAM memory failure rate. This research analyzes the ability to apply various machine learning approaches in learning the stability of memory circuit elements under manufacturing variability and the electrical use application condition. A key objective here is to evaluate the accuracy of machine learning approaches in replicating the response of a circuit simulator-based approach. This could then be extended to develop a user-facing tool that assesses and outputs an SRAM failure rate at a given use temperature and voltage.

2 Literature Review

There is a proliferation of semiconductor devices in the world around us, whether in personal electronics space, automotive, or industrial. Each market segment requires end application-specific analysis of memory reliability. Relying on traditional Monte Carlo based circuit simulation approaches can be very time-consuming and less adaptable to rapid reassessment needs of memory reliability for various design applications. Machine learning techniques to predict memory fails could be an effective alternative. In this section, the meaning of memory element stability is reviewed along with its associated metric, followed by summarizing traditional approaches in computing memory failure rate, and finally, recent literature on machine learning-based approaches to the problem.

2.1 Memory element stability

Ensuring the stability of a memory element across manufacturing process variations and use conditions is an important design requirement. An analytic and simulation-based framework to assess memory element stability has been previously investigated [6]. The memory element is considered stable if it can hold the data written into it at operating voltage and temperature. Stability is measured in terms of static noise margin (SNM), the maximum amount of either external DC voltage noise or internal transistor parameter offset that can be tolerated without losing stored data [6]. As part of current research, the SNM computation approach discussed above is used to generate a dataset for the purpose of training a machine learning model.

2.2 Traditional stability analysis approaches

Prior works [7][8][9][10] have relied on the Monte Carlo based circuit simulation approach to estimate memory element's failure probability. Memory element failure at use conditions is by design a rare event and involves capturing fail probabilities of a figure of merit metric, e.g., static noise margin or SNM. Number of Monte Carlo simulations 'N' needed to determine the probability of occurrence of failure (P_f), at significance level α, is given by [7]

$$N = \frac{4}{\alpha^2}\frac{(1-P_f)}{P_f} \quad . \tag{1}$$

The above formulation shows that the number of simulations needed is prohibitively large to estimate low fail probabilities reliably. For example, a number of Monte Carlo simulations needed to estimate failure probability of 1E-04, at 95% confidence interval is more than 10 million, requiring more than a week to complete [7].

The reason the Monte Carlo approach is very slow is that many simulation vectors get generated around the mean of the sampling distribution where the circuit does not

fail. The failure region is in the tail of the distribution, where enough samples are not generated to estimate the number of failing samples. The limitation of fewer samples in failure region is overcome using Importance Sampling (IS) [8] and mixture importance sampling approaches, which have shown speed up of simulation time by 100X [7]. Both approaches modify the sampling function to pick points in the failure region to set up Monte Carlo simulations and back-calculate true failure probability post-simulation using mathematical transformations. Mathematically, this is the concept is based on the below transformation [7] [8]

$$E_{p(x)}[\theta] = E_{g(x)}\left[\theta.\frac{p(x)}{g(x)}\right] \quad (2)$$

The above formulation states that the expected value of variable 'θ' when derived using a sampling distribution of p(x), is the same for revised variable 'θ.p(x)/g(x),' over the importance or new distribution g(x). Here p(x)/g(x) is the likely hood ratio that transforms the likelihood of occurrence to original distribution. The idea here is that the revised distribution is chosen. A larger number of simulation samples are generated in the failure region, helping converge robust failure rate estimates in fewer samples. However, since the failure region is not known beforehand, identifying a revised, modified sampling scheme is not straightforward. Importance sampling identifies a method to produce a revised sampling scheme by shifting the original sampling scheme by the center of gravity of the failure region [7]. Mathematically, this means

$$g(x) = p(x - \mu_0) \quad (3)$$

Here, the revised distribution g(x) is shifted by μ_0, so additional failure points are picked for simulation. The choice of μ_0 is to be determined through uniform sampling of parameter space noting locations of failure points, and taking mean of parameters associated with such fails. In a slightly modified approach, called Mixture Importance Sampling (MIS) [7] [8], the revised sampling function is chosen as a mixture of uniform and original Gaussian distribution. This approach is shown to improve speed up by over 1000x as compared to standard Monte Carlo.

Another approach uses "surrogate models" over and above importance sampling approaches to further reduce overall simulation time [9]. In this approach, a surrogate model describes the relationship between process variations and the circuit figure of merit response. This mathematical model helps evaluate the stability of memory elements faster than SPICE simulations. An additional order of magnitude speedup is achieved by combining the improved failure sampling scheme, i.e. importance sampling, and the surrogate model in lieu of SPICE simulations. Yao et al. [9] use radial basis function network-based surrogate model and refer to other approaches to develop such model, e.g., artificial neural network and surface response modeling.

Finally, importance sampling-based schemes considered above become inefficient as the data dimensionality increases [10], and a new scaled sigma sampling (SSS) method is proposed to overcome it. In SSS, random samples are drawn from a distorted probability density function with a 'scaled up' standard deviation. This leads to larger failure points being picked for the same number of circuit simulations. While this

approach helps address the problem of failure rate estimation, it still relies on the use of circuit simulation to determine the stability of the memory element.

2.3 Machine Learning based stability analysis approaches

Dataset associated with memory element failures is highly imbalanced, as very few failures are recorded. The data set for the work is available from the Monte Carlo circuit simulation-based approach with features or parameters representing manufacturing variability and memory use conditions. Building a machine learning approach that could mimic a circuit simulator-based approach to identify unstable memory elements in various use or test conditions requires techniques to handle highly imbalanced datasets. As such, the current paper explores various data imbalance handling approaches [12].

Prior studies using Support Vector Machine (SVM) Surrogate Model (SM) based methods for parametric yield optimization [13] and using Random Forest classifier [14] to detect rare failure events have shown promising results.

The rarity of the failures also meant that these failures could be considered as outliers in the dataset. With advancements in methods, models, and classification techniques in detecting outliers [15], the current paper also explores various outlier detection techniques in building a better machine learning model. Guidelines to manage univariate and multivariate outliers and tools to detect outliers [17] are considered. Recommendations to use the median absolute deviation to detect univariate outliers and use Mahalanobis-MCD distance to detect multivariate outliers [17] are explored.

Considering various approaches, the hypothesis validated in the current paper is that Machine Learning approaches for circuit failure analysis could mimic simulation-like accuracy and minimize the need for engineers to rely heavily on simulators for their validations.

3 Methodology

This section discusses the overview of data, metrics used, and methods and techniques used to detect memory element failures.

3.1 Data

For evaluating various machine learning models in the current paper, dataset is generated from running Monte Carlo SPICE (Simulated Program with Integrated Circuit Emphasis) simulations. It involves instantiating the memory element circuit in a netlist and running Monte Carlo runs, where each run contains a unique input vector representing manufacturing process variation and use conditions.

There are 14 total features, of which 12 represent process variation, and one each for use supply voltage ‘Vdd’ and use temperature ‘T’. These features are independent. The

process variation variables follow the standard normal Gaussian distribution. The voltage values range between maximum and minimum operating voltages. The temperature range in use conditions starts from –40 °C and to a maximum of 200 °C.

The output variable part of the original data set is 'Vdelta', which is a measure of how stable the memory element is. For a given input vector consisting of process variation, voltage and temperature, the value of 'Vdelta' lies between -Vdd, and +Vdd. The more positive 'Vdelta' is above zero, the more stable the memory element is. All memory element with 'Vdelta' ≤ 0 are unstable. For every input supply voltage, a 'Vdelta' value normalized to 'Vdd' is used for modeling purposes. Another output variable derived from Vdelta is 'FAIL' variable, that can take two classes, namely '1' and '0', where '1' represents failure, while '0' represents no failure, i.e. a stable memory element. These two variables provide flexibility to explore modeling as a regression problem, or a binary classification problem. Former is the scenario when 'Vdelta' variable is used, while latter is the scenario when 'Fail' output variable is used.

3.2 Data Analysis

The dataset contains stability assessment for 100,000 sampled instances of process variable at different Vdd, and Temperature. Summary of voltage and temperature combinations present in dataset can be reviewed in Table 1 below.

Table 1. Summary of voltage and temperature combinations present in dataset. For each voltage and temperature pair, 100,000 instances of process variable samples are present. The voltage values are standard normalized.

Normalized Supply Voltage	**Normalized Temperature (T) Range**		
	-40C	30C	150C
-1.39	100,000	100,000	100,000
-0.72	100,000	100,000	NA
0.05	100,000	100,000	NA
0.62	100,000	100,000	100,000
1.29	100,000	100,000	100,000

The twelve process variables are independent of each other, and follow standard normal Gaussian distribution with '0' mean, and standard deviation of '1', refer Fig 1.

An additional aspect of the dataset is the highly imbalanced nature of target variable 'FAIL'. The number of failing memory elements reduce exponentially at higher voltage levels. This is evident from Fig. 2 below. At the highest voltage there is only 1 failure in a sample of 100,000. When modeling the data as a binary classification problem, the highly imbalanced nature of this variable may need to be accounted in modeling efforts to improve classifier performance.

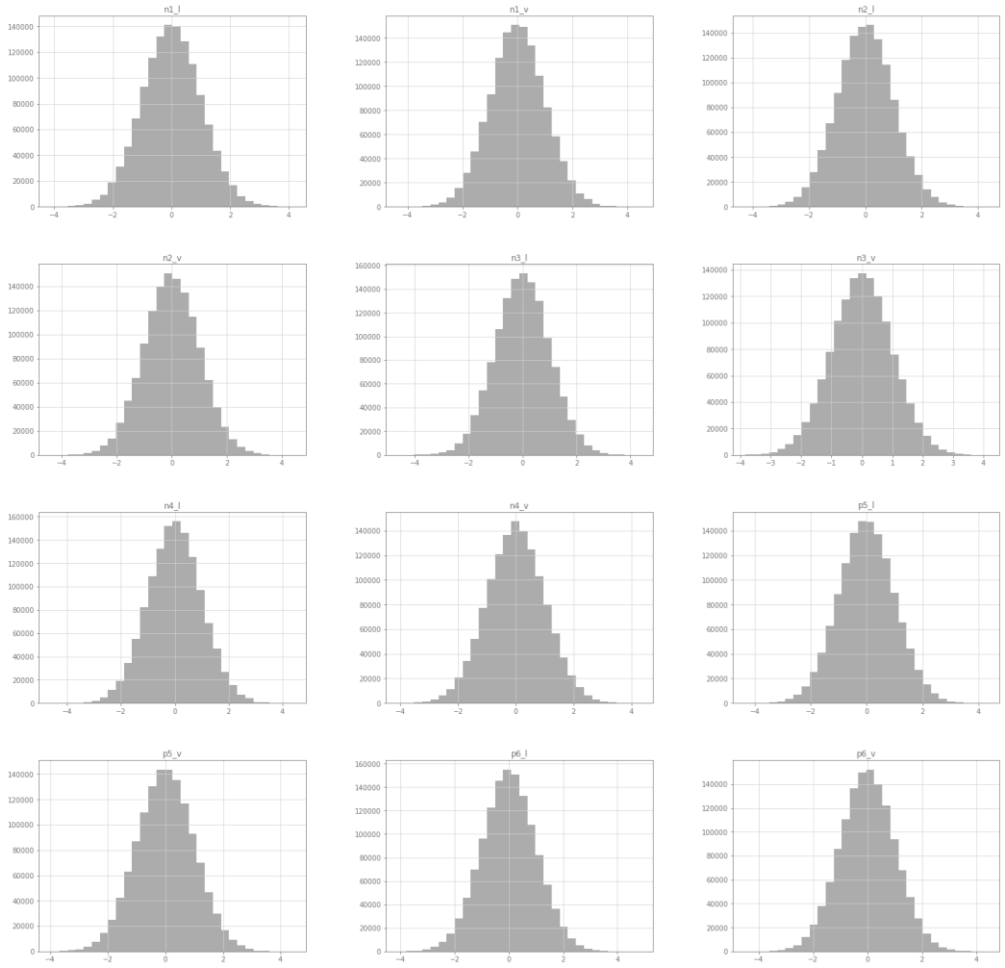

Fig. 1. Figure showing feature distributions of all the variables in the dataset. The first 12 histograms show that the process variation variables follow the standard normal Gaussian distribution.

Correlation analysis between the target variable, and input features is used to determine features that can be leveraged to build robust machine learning models. Table 2 summarizes the correlation values, along with a categorization of input features into highly correlated, and poorly correlated buckets.

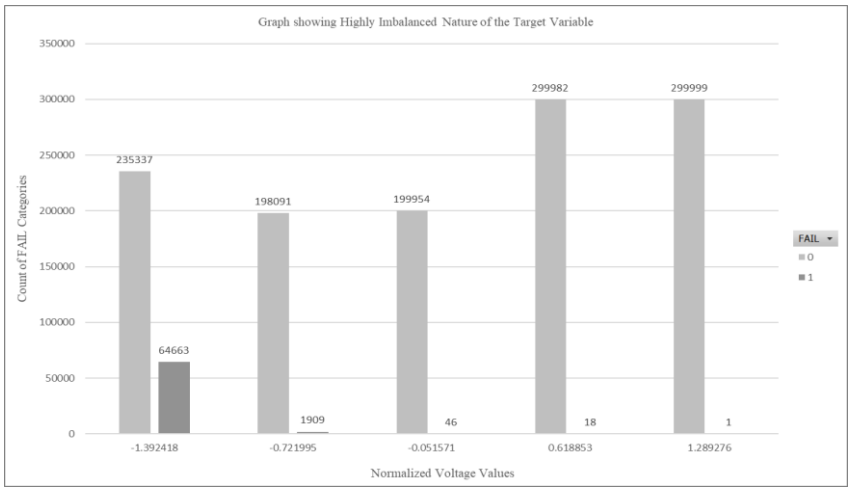

Fig. 2. Graph showing the highly imbalanced nature of the target variable. The X-axis represents the normalized voltage values, and the y-axis counts FAIL categories (target variable). The two FAIL categories are 1 and 0. A memory element failure is indicated by 1, and a stable memory element is indicated by 0.

Table 2. Summary of correlations between target variable (Vdelta) and input features (process variable, supply voltage, and temperature)

Input variable	Feature type	Correlation Value	Assessment
n3_v	Process variables	0.25	Highly Correlated ($\|r\| \geq 0.1$)
n2_v		0.13	
n1_v		-0.21	
p6_v		0.13	
Vdd	Supply Voltage	0.34	
Tj	Temperature	-0.19	
n1_l	Process variables	0.0096	Poorly correlated ($\|r\| < 0.04$)
n2_l		-0.011	
n3_l		-0.0049	
n4_l		0.00066	
n4_v		-0.038	
p5_l		-0.0025	
p5_v		0.011	
p6_l		0.01	

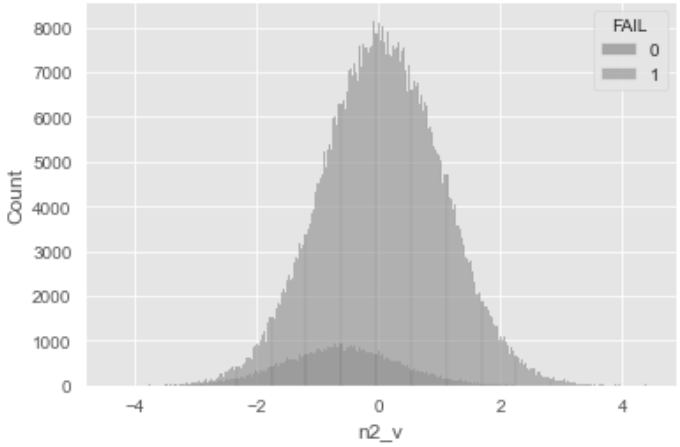

Fig. 3a. Histogram of process variable 'n2_v' as a function of memory element stability condition. It contains all voltage, and temperature points. The FAIL=1 distribution is towards left of FAIL=0. An important observation is that FAIL=1 region is not localized in a small range of values. It is possible to have a few failures even when n2_v is positive and close to 2σ point.

The highly correlated variables are

1. Process variables – n1_v, n2_v, n3_v, p6_v
2. Supply voltage & temperature.

Variables with very low correlation are:

1. Process variables – n4_v, n1_l, n2_l, n3_l, n4_l, p5_v, p5_l, p6_l

Semiconductor circuit theory, and functioning of memory element supports the correlations noted above. Some observations in this regard are:

1. when n3_v is larger, the corresponding transistor in memory element is weaker, and it's harder for stored charge to be lost; so, the internal node voltage level is preserved.
2. For n2_v variable, however, the effect is weaker which reflects in smaller correlation number.
3. Supply voltage is the most strongly correlated variable, as larger supply voltage leads to more stable memory element, i.e. larger Vdelta
4. Higher temperature values lead to larger fails, i.e. smaller Vdelta which is reflected in negative correlation coefficient.

Another key aspect of interrelationship between stability of memory element, and individual process variables is visualized by looking at histogram plots as a function of memory element being stable or unstable, refer Fig 3(a) and 3(b). Two important observations are:

1. Mean of n2_v distribution for unstable memory elements is lower than stable memory elements. The scenario is reversed for n1_v
2. Both n2_v and n1_v process variables have a wide range over which memory element can fail. This is almost 4σ, as can be visually observed.

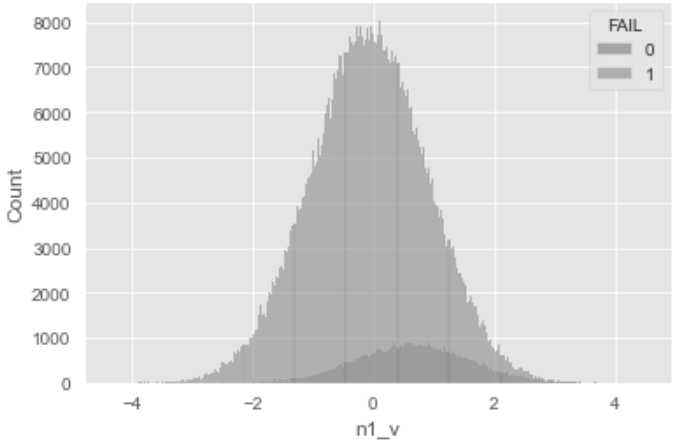

Fig. 3b. Histogram of process variable 'n1_v' as a function of memory element stability condition. It contains all voltage, and temperature points. For n1_v, FAIL = 1 is to right of FAIL= 0. Further a very important observation is that FAIL=1 region is not localized in a small range of values. It is possible to have a few failures even when n1_v is negative and close to -2σ point.

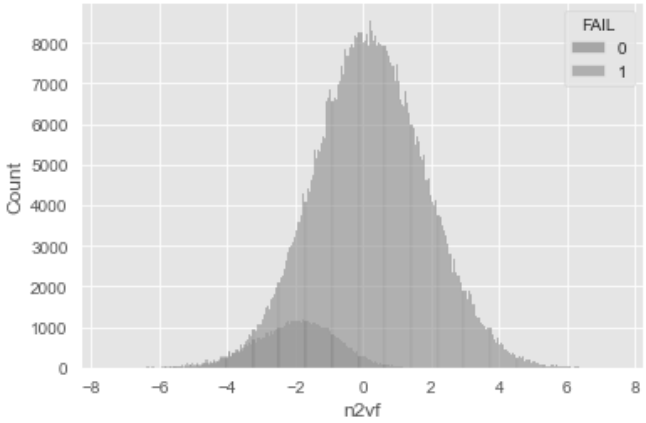

Fig. 4. Histogram of derived process variable feature n2vf given by n2vf = n2_v - n1_v + n3_v as a function of memory element stability condition. It contains all voltage, and temperature points. As compared to n2_v, the engineered process variable feature appears to distinguish failing and not failing memory elements more sharply.

Above observations are consistent with correlation coefficient sign for both n2_v and n1_v. Fundamentally it also suggests that there is likely no narrow region in 12-dimensional process variable space, which is exclusively a failure region.

Further, engineering a new process variable 'n2vf' given by n2vf = n2_v – n1_v + n3_v appears to failing and non-failing distributions that are more separable or distinct, refer Fig 4.

3.2 Metrics

When building and evaluating various machine learning models, metrics such as precision and recall are used to compare results. Definitions for the metrics are listed below.

To detect memory element failures, identify a failure is considered *positive*. Results from a machine learning model are captured in a tabular layout referred to as a Confusion Matrix; see Table 3 below.

Table 3. Confusion Matrix for a binary classification problem.

Classes		**Actual**	
		Positive	**Negative**
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Precision. The measure of the number of *predicted* positives that are true positives and is shown by the formula below.

$$\text{Precision} = \frac{TP}{TP+FP} \quad . \tag{1}$$

Recall. The measure of the number of *actual* positives that are true positives and is shown by the formula below.

$$\text{Recall} = \frac{TP}{TP+FN} \quad . \tag{2}$$

Precision serves an important measure when the cost of a False positive is higher. However, when determining stability of memory elements having a higher Recall rate becomes crucial, as the tangible and intangible cost associated with shipping a faulty memory element to the field is much higher when compared to discarding a good memory element. Hence, when comparing various machine learning models, the current paper focuses on achieving a higher Recall rate.

3.3 Modeling Approach

In the development of modeling approach, following insights from data analysis are considered:

1. Only six out of fourteen input variables have high correlation with output.
2. The target variable for modeling perspective can be a numeric variable, or a binary class variable
3. Failing memory elements are not restricted to a narrow range of input variables, but can spread over a wide range of values (4σ) for highly correlated process variables

In the first set of experiments, baseline recall performance of Random Forest algorithm as a binary classifier vs as a regressor is benchmarked. When target variable is numeric, it is hypothesized that algorithm should perform better in identifying the unstable memory elements. This is experiment (1) vs experiment (3) in Table 4. Next, impact of choosing only highly correlated features is compared to all the features is assessed, see experiment (1) vs experiment (2) and experiment (3) vs experiment (4).

Table 4. Summary of baseline ML model development experiments. Results are reviewed separately below.

Exp. No.	Algorithm	Input feature set used	Target class	Problem construction
1	Random Forest Classifier	All	Binary	Binary classification
2		Highly correlated only		
3	Random Forest Regressor	All	Numeric	Regression
4		Highly correlated only		

Later, multiple regression and classification algorithms are assessed for recall score. In regression, these are stack ensembles and voting ensemble of XGBoost and LightGBM, Extreme Random trees, XGBoost, LightGbM etc. In classification, Logistic regression, XGBoostClassifier, Extreme Random trees etc. The results are summarized in next section.

4 Results

Results from baseline experiments using Random Forest Classifier, and Random Forest Regressor are tabulated in Table 5 and Table 6 respectively.

Table 5. Summary of baseline Random Forest Classifier model performance results – accuracy, recall and precision scores.

Exp. No.	Description	Accuracy Score	Recall Score	Precision Score
1	Dataset with all feature variables	96.65%	33.44%	99.60%
2	Dataset with only highly correlated feature variables	98.05%	66.99%	98.05%

Table 6. Summary of baseline Random Forest Regressor model performance results – mean RMSE, recall and precision scores.

Exp. No.	Description	Mean RMSE	Recall Score	Precision Score
3	Dataset with all feature variables	0.17671	76.91%	86.88%
4	Dataset with only highly correlated feature variables	0.18105	76.24%	87.73%

Comparing the results Random Forest Regressor performs much better on recall metric. By providing a full numeric range of stability values, through target variable 'Vdelta' the model performs better in detecting True Positives. However, the model performance does not improve when highly correlated input features are used with Random Forest Regressor, (experiment (3) vs experiment (4) in results table 6). The best baseline model performance is by Random Forest Regressor that uses all input features and provides an RMSE score of 0.17671. Note that the numeric Vdelta outcome predicted by regressor algorithm is converted to memory element stable, or unstable class by comparing against threshold of '0V'.

Finally, more complex ensemble and stacked machine learning models are run, and their performance tabulated in Table 7 and Table 8 respectively for classification and regressor based approaches. Clearly Logistic Regressor and Stacked Ensemble with XGBOOST, and LightGBM provide best model performance, which is also much better than baseline model performance. The best model performance here has a recall of 0.97. Recall of 0.97 indicates that among all the fails noted, about only ~3% are due to False Negatives. Low False Negatives ensure that there would be few or no instances of memory element being modeled as passing, while failing in customer hands. Note that when producing memory elements by millions or in large quantities, the cost of discarding few good memory elements is always going to be lesser than incurring business loss due to customer's product fails.

Table 7. Summary of final ensemble, and stacked ML classification model performance results – recall score

Experiment Number	Model Description	Recall Score

1	MaxAbsScaler, LogisticRegression	97.26%
2	StandardScalerWrapper, LogisticRegression	97.25%
3	MaxAbsScaler, SGD	97.09%
4	SparseNormalizer, LogisticRegression	96.88%
5	StandardScalerWrapper, XGBoostClassifier	96.40%

Table 8. Summary of final ensemble, and stacked ML regressor model performance results– mean RMSE score.

Experiment Number	Model Description	Normalized RMSE
1	StackEnsemble (XGBR, LGBM)	0.07127
2	VotingEnsemble (XGBR, LGBM)	0.07228
3	StandardScalerWrapper, XGBoostRegressor	0.07245
4	MaxAbsScaler, LightGBM	0.07528
5	MaxAbsScaler, ExtremeRandomTrees	0.12781

5 Discussion and Ethics

The research done as part of this work looked at application of machine learning techniques in accurately modeling failure statistics of semiconductor memory element. The results summarized so far indicates that Machine Learning models can be trained to learn electrical response of semiconductor memory element, for different process variation and use conditions to a very good accuracy and high recall score.

Adoption of Machine learning based technique in semiconductor design space is at nascent stage. The work presents proof of concept of an application in semiconductor technology that can be steppingstone to wider deployment of Machine learning based techniques. Current production electronic design automation tools and methodologies do not leverage machine learning techniques. At a fundamental tool and methodology level, machine learning techniques are not part of validation exercise to compare response of circuit to actual silicon data. In fact, in the research attempted as part of this work, the machine learning technique is validated against simulator output and not actual silicon data. Since existing production tools and flows themselves have limitations on accuracy in modeling actual silicon behavior, it needs to be understood

further how tool outcome, that uses machine learning, directly compares to silicon. It is possible that additional inaccuracy in silicon modeling gets introduced. Further rigorous studies and careful assessment may be needed to understand this further and identify ways to mitigate model performance gap to silicon, if any.

In the existing standard methodology, Monte Carlo simulations have to be setup and used and require subject matter expertise every time stability assessment needs to be done. The results from the research suggest it is possible for a machine learning algorithm-based analysis framework to be packaged and deployed for assessing memory element failure rate. In a machine learning model-based framework, a model developed using existing data, or one-time generated data, can be re-used in customer's hand as a software tool. A machine learning model, once developed in consultation with subject matter expert, can then be re-used and deployed as a back engine to provide analysis support to wide range of teams. However, requires a detailed review of model assumptions and validations with users of the models, and other key stake holders in product marketing and quality space.

On one hand no machine learning technique can 100% replicate simulator results, it has potential to get widely deployed across teams if embedded in a software package that is easy to use. In such scenario, implications of additional false negatives or positives needs to be well understood. Various stakeholders need to align on consequences for an additional yield loss or an additional failure in customer hands. It may be possible to factor additional fails in field to be mitigated by building in additional redundancy in memories, but that may add a little to system cost. It is possible existing fault mitigation techniques may suffice to address them, and there may be no additional cost, but all of these discussions need to happen and get addressed before deploying AI/ML based system.

Further, there can be some applications where reliability is very critical, e.g. for medical devices or in space application. Here, even a single incorrect assessment can be of high consequence. In such use cases, additional review of ML based approach should be undertaken and supplemented with some additional testing approaches. Hence, it may not be possible to have one model that fits all application requirements, and additional model development and validation exercises will be needed to safely use the technique.

ML based approach significantly reduces the analysis time and so it could be very tempting to leverage this approach when time-to-market becomes a crucial aspect of business. However, it is very important to establish organization wide principles to put quality and safety in front of profits.

6 Conclusion

The research undertaken shows that Machine Learning based approach can successfully learn the semiconductor element's electrical response for stability analysis. This is accomplished by training algorithm on a dataset generated from output of Monte Carlo SPICE simulations. The algorithms have achieved a very high degree of accuracy in replicating simulator outcome and achieved a recall score of 0.97. This work lays

foundation for future efforts to further improve the accuracy of models and consider usage and deployment strategies to mitigate impact of inaccuracies introduced.

Acknowledgments. The authors would like to thank Texas Instruments Inc for support in providing dataset, and especially to Anand Seshadri for useful discussions, and mentoring on the subject matter. Special thanks to also Jacquelyn Cheun for guidance and review of paper drafts through the duration of the project.

References

1. Shukla, P. (n.d.). Types of memories in computing system-on-Chips. Design And Reuse. https://www.design-reuse.com/articles/43464/types-of-memories-in-computing-system-on-chips.html
2. Anadiotis, G. (2020, May 21). AI chips in 2020: Nvidia and the challengers. ZDNet. https://www.zdnet.com/article/ai-chips-in-2020-nvidia-and-the-challengers/
3. Hanlon, J. (n.d.). Why is so much memory needed for deep neural networks? Graphcore: Accelerating machine learning for a world of intelligent machines. https://www.graphcore.ai/posts/why-is-so-much-memory-needed-for-deep-neural-networks
4. Dsouza, J. (2020, December 26). What is a GPU and do you need one in deep learning? Medium. https://towardsdatascience.com/what-is-a-gpu-and-do-you-need-one-in-deep-learning-718b9597aa0d
5. Cerebras. (2021, April 20). Product. https://cerebras.net/product/
6. Seevinck, E., List, F., & Lohstroh, J. (1987). Static-noise margin analysis of MOS SRAM cells. IEEE Journal of Solid-State Circuits, 22(5), 748–754. https://doi.org/10.1109/JSSC.1987.1052809
7. Kanj, R., Joshi, R., & Nassif, S. (2006). Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events. Proceedings of the 43rd Annual Design Automation Conference, 69–72. https://doi.org/10.1145/1146909.1146930
8. Bayrakci, A. A., Demir, A., & Tasiran, S. (2010). Fast Monte Carlo Estimation of Timing Yield With Importance Sampling and Transistor-Level Circuit Simulation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 29(9), 1328–1341. https://doi.org/10.1109/TCAD.2010.2049042
9. Yao, J., Ye, Z., & Wang, Y. (2015). An Efficient SRAM Yield Analysis and Optimization Method With Adaptive Online Surrogate Modeling. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 23(7), 1245–1253. https://doi.org/10.1109/TVLSI.2014.2336851
10. Sun, S., Li, X., Liu, H., Luo, K., & Gu, B. (2015). Fast Statistical Analysis of Rare Circuit Failure Events via Scaled-Sigma Sampling for High-Dimensional Variation Space. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34(7), 1096–1109. https://doi.org/10.1109/TCAD.2015.2404895
11. Singhee, A., & Rutenbar, R. (2009). Statistical Blockade: Very Fast Statistical Simulation and Modeling of Rare Circuit Events and Its Application to Memory Design. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 28(8), 1176–1189. https://doi.org/10.1109/TCAD.2009.2020721
12. Shaer, L., Kanj, R., & Joshi, R. (2019). Data Imbalance Handling Approaches for Accurate Statistical Modeling and Yield Analysis of Memory Designs. 2019 IEEE International Symposium on Circuits and Systems (ISCAS), 1–5. https://doi.org/10.1109/ISCAS.2019.8702731

13. Ciccazzo, A., Di Pillo, G., & Latorre, V. (2016). A SVM Surrogate Model-Based Method for Parametric Yield Optimization. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 35(7), 1224–1228. https://doi.org/10.1109/TCAD.2015.2501307
14. El-Adawi, R., & Dessouky, M. (2017). Monte Carlo General Sample Classification for Rare Circuit Events using Random Forest. 14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design, 1-4. https://doi.org/10.1109/SMACD.2017.7981599
15. Kimmel, R., Li, T., & Winston, D. (2020). An Enhanced Machine Learning Model for Adaptive Monte Carlo Yield Analysis. Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD, 89-94. https://doi.org/10.1145/3380446.3430635
16. Boukerche, A., Zheng, L., & Alfandi, O. (2020). Outlier Detection: Methods, Models, and Classification. ACM Computing Surveys, 53(3), 1–37. https://doi.org/10.1145/3381028
17. Leys, C., Delacre, M., Mora, Y. L., Lakens, D., & Ley, C. (2019). How to classify, detect, and manage univariate and multivariate outliers, with emphasis on pre-registration. https://doi.org/10.5334/irsp.289