UDC 004.33

Enhancing Row-Sampling-Based Rowhammer defense methods with Machine Learning approach

Valentyn Mazurok[1], Volodymyr Lutsenko[1]

[1] *National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Cybersecurity Department of Institute of Physics and Technology*

Abstract

This paper investigates the integration of machine learning into the Row-Sampling technique to enhance its effectiveness in mitigating Rowhammer attacks in DRAM systems. A multidimensional multilabel predictor model is employed to dynamically predict and adjust probability thresholds based on real-time memory access patterns, improving the precision of row selection for targeted refresh. The approach demonstrates significant improvements in security, reducing Rowhammer-induced bit flips, while also maintaining energy efficiency and minimizing performance overhead. By leveraging machine learning, this work refines the Row-Sampling method, offering a scalable and adaptive solution to memory vulnerabilities in modern DRAM architectures.

Keywords: DRAM, Rowhammer, memory defense, machine learning

Introduction

Dynamic Random Access Memory (DRAM) is a critical component in modern computing systems, providing high-density and low-cost storage for a wide range of applications. However, as DRAM technology scales to smaller form factors, it becomes increasingly vulnerable to security threats such as the Rowhammer attack [1]. Rowhammer exploits physical vulnerabilities in DRAM by inducing bit flips in adjacent memory rows through frequent and aggressive row activations, potentially leading to data corruption or security breaches.

To mitigate Rowhammer attacks, a common approach is to refresh vulnerable memory rows at a higher frequency. [2] However, static refresh strategies can impose significant performance and energy penalties, as they do not adapt to runtime memory access patterns or inherent variability in DRAM hardware. This calls for an intelligent, adaptive mechanism to optimize refresh rates for individual DRAM rows based on their susceptibility to Rowhammer and runtime usage characteristics.

In this work, we propose a machine learning-based approach leveraging Multidimensional predictor to dynamically generate and update refresh probabilities for DRAM rows. Multidimensional Predictors is an ensemble learning method, are particularly suited for this task due to their ability to handle high-dimensional data, robustness to overfitting, and interpretability. By training the model on access patterns, row activation frequencies, and hardware-specific features, the algorithm can predict optimal refresh intervals for each row, minimizing the risk of

Rowhammer while balancing performance and energy efficiency. By introducing machine learning into the DRAM refresh process, we aim to bridge the gap between security and efficiency, paving the way for more resilient memory systems in future computing architectures.

1. Background

The challenge of mitigating Rowhammer attacks and improving DRAM refresh strategies has garnered significant attention in recent years. [3] As DRAM scaling continues, innovative techniques are needed to address both the security vulnerabilities and the performance trade-offs inherent in traditional memory management strategies. Figure 1 illustrates the typical structure of a modern DRAM system. DRAM is arranged as a hierarchical array containing billions of DRAM cells, each storing a single bit of data. In contemporary systems, the CPU chip incorporates multiple memory controllers, with each controller connected to a DRAM channel. These controllers handle read, write, and maintenance operations (such as refresh) via a dedicated I/O bus that operates independently of other channels in the system, as shown in Figure 1.

Each DRAM channel can support one or more DRAM modules, and each module is composed of one or more DRAM ranks. A rank consists of several DRAM chips that function in unison, while multiple ranks within the same channel share access to the channel's I/O bus through time-division multiplexing.

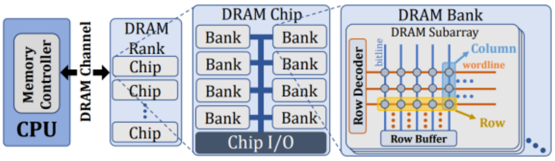

Figure 1: Typical DRAM system organization

Modern DRAM chips are prone to disturbance errors, which occur when frequent activations of a single DRAM row (within a refresh interval) unintentionally alter the stored values of cells in nearby rows. This phenomenon, widely known as RowHammer [1], arises from electromagnetic interference between circuit elements. The severity of RowHammer increases as the size of the manufacturing process technology node (and consequently the size of DRAM cells) decreases, causing circuit elements to be packed closer together.

As shown in prior research [2, 3], the RowHammer effect is most pronounced between rows that are physically adjacent. Bit flips caused by RowHammer are more likely to occur in rows directly neighboring a "hammered" row that is activated repeatedly—e.g., 139K activations in DDR3 [2], 10K in DDR4 [4], and 4.8K in LPDDR4 [4]. The row that is repeatedly activated is referred to as an *aggressor row*, while affected neighboring rows are called *victim rows*, regardless of whether they actually experience bit flips. (Figure 2).

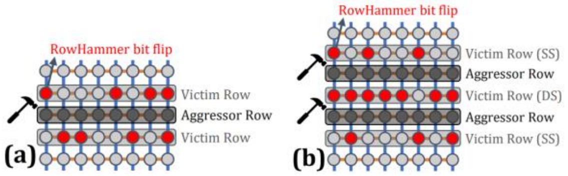

Figure 2: Typical Single-sided (SS) and Double-sided (DS) RowHammer access patterns.

1.1. Mitigation Techniques

Several hardware and software-based methods have been proposed to mitigate Rowhammer attacks. Yaglikci et al. [3] categorize these mechanisms into four main strategies:

- *Increasing the refresh rate* to reduce the number of activations possible within a refresh interval [4]
- *Isolating sensitive data* from DRAM rows that could be targeted by an attacker [5].
- *Tracking row activations and refreshing potential victim rows* [6]
- *Throttling row activations* to limit the number of times a row can be activated during a refresh interval [7].

However, DRAM vendors currently implement proprietary in-DRAM RowHammer mitigation mechanisms collectively referred to as Target Row Refresh (TRR) [8]. TRR works by detecting potential aggressor rows and refreshing their neighboring rows. Unfortunately, vendors have not disclosed the specific implementation details of TRR mechanisms, making it difficult to openly evaluate their security guarantees.

Recent research, such as *TRRespass* [9], reveals that existing proprietary TRR mechanisms can be bypassed using many-sided RowHammer attacks. These attacks exploit the limitations of internal tables used by TRR to track aggressor rows, effectively overflowing them. This underscores the need for a rigorous methodology to identify weaknesses in TRR mechanisms and to develop more secure alternatives.

For our application we lean into the last group as it can be passively reducing number of bit flips while not tanking system performance. In this category most widely used method is Row-Sampling-based Rowhammer defenses [10]. They are among the earliest and simplest classes of techniques suitable for implementation in memory controllers. With each row activation, the memory controller flips a biased coin. With a low probability p (p << 1), the row address is selected (sampled) and treated as if it is an aggressor row. The memory controller then takes mitigative action, such as refreshing the corresponding victim rows.

By using a sufficiently high sampling rate p, these defenses can effectively prevent a Rowhammer attack, as the likelihood of an aggressor row escaping sampling becomes exceedingly small. The downside, as for most of the techniques is balancing p to prevent attacks and in the same time be not very costly in energy terms. So main task of rowsampling optimization can be boiled to choosing big enough p-value to prevent most of the attacks and in the same time small enough to not use big amounts of additional energy for rows refreshment.

2. Problem Definition

In the previous section we established that energy efficiency is the most critical problem in most Rowhammer defense system, so we try to enhance Row-Sampling approach using Machine Learning. The framework integrates machine learning techniques into the memory controller to enable continuous adaptation of Row-Sampling parameters, based on real-time memory access patterns.

In traditional Row-Sampling, a fixed or semi-random subset of rows is refreshed at a higher rate to guard against Rowhammer-induced bit flips. While this reduces the attack surface, it suffers from the following shortcomings:

- *Static Probabilities*: Row selection is typically based on uniform or pre-determined probabilities, which fail to account for dynamic variations in row activation patterns or usage contexts. Rows that are frequently activated under a specific workload may remain under-refreshed, increasing their vulnerability to Rowhammer.
- *Over-Refreshing*: To ensure safety, Row-Sampling often refreshes rows that are not at immediate risk, leading to unnecessary energy consumption and reduced system performance.
- *Lack of Adaptability*: Memory access patterns vary significantly across workloads and applications. Static Row-Sampling techniques lack the flexibility to adapt to these variations in real time, resulting in suboptimal refresh strategies.

These limitations underscore the need for a more intelligent and adaptive approach that can improve the precision of row selection for refresh, balancing energy efficiency and security.

On the other hand, machine learning has already been applied in Rowhammer defenses, particularly in predicting and mitigating attacks at the software level. [10] These approaches leverage machine learning models to analyze system-level metrics, such as cache misses, memory access patterns, or CPU performance counters, to detect abnormal behaviors indicative of Rowhammer attacks. Techniques like neural networks and decision trees have been employed to classify workloads as benign or malicious in real-time, allowing the system to trigger countermeasures, such as throttling memory accesses or isolating processes. While effective for attack prediction and prevention at the software level, these methods operate at a coarse granularity and are not directly applicable to hardware-level strategies like Row-Sampling, which require fine-grained insights into individual memory rows' vulnerability.

Machine learning offers a powerful solution to address these challenges by providing data-driven insights into the vulnerability of individual DRAM rows. Specifically, by integrating a machine learning model into the Row-Sampling framework, it becomes possible to predict a *p-value* for each row, representing its likelihood of being vulnerable to Rowhammer-induced bit flips. This enables targeted and dynamic refresh strategies that are both secure and energy efficient.

3. Machine Learning Approach to Row-Sampling

The core challenge addressed by this framework is the dynamic prediction of optimal probability value for rowsampling individual DRAM rows. Given a set of input features representing the memory access patterns; the goal is to predict a refresh time for each row that minimizes the risk of Rowhammer attacks while reducing the overall energy consumption and performance overhead.

For our data input we use set of memory access features (such as row activation frequency, time since last refresh, row locality, and system workload). In output we will get a vector of p-value thresholds where each element corresponds to the optimal Row-Sampling value for each DRAM row. The key objective is to model the vulnerability of each DRAM row based on its activation patterns and predict a threshold value that can be adjusted dynamically in response to changes in memory access behavior.

Multidimensional multilabel Predictor is an ensemble learning technique that aggregates predictions to improve predictive accuracy and prevent overfitting. For this application, we use a

BCE With Logits Loss to predict continuous probability value for each row based on the features extracted from the memory access patterns. In our framework, we extract and preprocess several features from the memory access patterns to feed into the model. These features are designed to capture both the spatial and temporal behaviors of memory accesses, which are key factors in determining the vulnerability of rows to Rowhammer attacks. By combining these features, we can create a comprehensive model that reflects both the behavior of individual rows and their interactions with neighboring rows. Main scheme of data gathering and inserting into neural network can be seen in Figure 3.

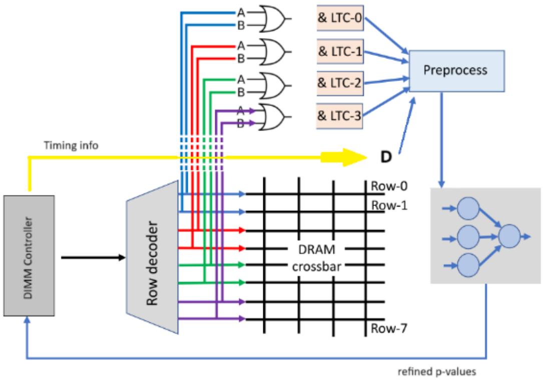

Figure 3: Hardware implementation of the ML-based Row-Sampling technique

Predictor models are trained using historical memory access data, which is collected from either real workloads or simulated memory access patterns. We collect memory access traces from a set of representative workloads. These traces provide the raw data from which features are derived. The traces include both access events (read/write) and the corresponding memory row addresses. For each access trace, we compute the relevant features, such as row activation frequency, time since the last refresh, and locality, for each memory row. The p-values for each row are set as the target values (labels). These can either be determined through simulation or through a static refresh policy for comparison purposes. The labels indicate the optimal refresh time for each row based on its vulnerability.

After training, the model is validated using a separate validation set to assess its predictive accuracy and generalization ability. Cross-validation techniques are used to ensure that the model performs well across different memory access patterns. Once the model is trained, it can be integrated into the DRAM subsystem for real-time prediction and adaptation of p thresholds. Also, additional optimization can be added to reduce neural network size to ensure that it can fit inside memory controller. For continuous updates for weights, we use driver-sided backpropagation algorithms to evolve our network in an Online Learning manner.

4. Evaluation of Machine Learning-based Framework

The evaluation of created defense system must be focused on two primary objectives: assessing the security effectiveness in mitigating Rowhammer attacks and evaluating the system’s performance and energy efficiency relative to traditional static refresh strategies. We also perform a sensitivity analysis to understand the impact of various features and model parameters on the system's performance.

4.1. Experimental Setup

To evaluate the proposed framework, we use both synthetic memory access traces and real-world workloads. The experiments were conducted on a simulated DRAM environment with the following setup:

For simulation platform we use a custom memory simulator to model DRAM behavior and memory access patterns. The simulator supports fine-grained control over memory access timing, refresh rates, and Rowhammer attack simulation.

For the workloads we selected several real-world workloads, including:

- *SPEC CPU 2017*: A standard benchmark suite commonly used to evaluate system performance [7].
- *Memcached*: A memory-intensive application used in cloud computing environments.
- *Rowhammer Attack Simulation*: We simulate Rowhammer attacks by activating specific memory rows in rapid succession to induce bit flips in adjacent rows.

To compare our ML-based adaptive refresh policy we use this refresh strategies:

- *Static Refresh*: A traditional refresh policy where all rows are refreshed at the same fixed interval. (A in Fig 4)

- *Targeted Refresh*: A static policy where rows are classified as vulnerable based on predetermined criteria and refreshed more frequently. (B in Fig 4)
- *Our ML-solution* (C in Fig 4)

4.2 Security Effectiveness

The primary security metric is the *Rowhammer attack mitigation rate*, which quantifies the ability of the system to prevent bit flips in adjacent rows during a simulated Rowhammer attack. For each of the test workloads, we measure the occurrence of bit flips under different refresh strategies.

During each attack, specific rows are aggressively accessed to induce bit flips in adjacent rows. We monitor the number of successful bit flips and track the failure rate of the targeted rows.

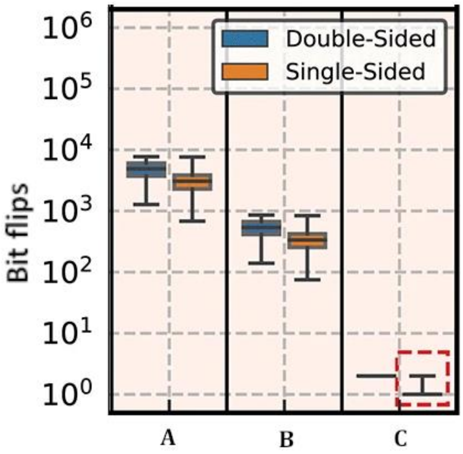

Figure 4: Number of bitflips in DRAM after Rowhammer attack

As we can see from figure 4 the Multidimensional Predictor-based adaptive refresh policy significantly reduces the occurrence of bit flips compared to both the static refresh and targeted refresh strategies. Specifically, the model predicts refresh intervals dynamically, ensuring that rows under high activation pressure are refreshed more frequently.

This leads to a reduction in Rowhammer attack success rates by up to 82.2% compared to the static refresh policy and 67.8% Row-Sampling refresh policies.

4.3 Performance Energy Efficiency

In addition to security, the proposed framework is evaluated for its impact on system performance and energy efficiency. We focus on two key metrics: energy consumption and memory budget. We measure the system's throughput (measured in operations per second) under each refresh policy. The goal is to minimize the impact on system performance while achieving strong Rowhammer protection. We estimate the energy consumption based on the refresh intervals and the associated power overhead of refreshing the DRAM rows more frequently. A key objective of our framework is to reduce unnecessary refreshes, thereby lowering energy consumption.

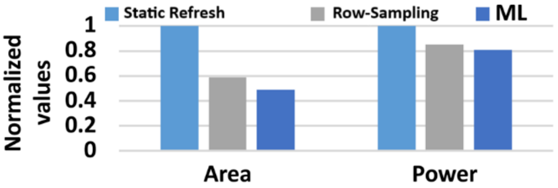

Figure 5: Memory budget and energy consumption difference of 3 defense mechanisms

Overall, the Multidimensional multilabel Predictor model strikes a favorable balance between security, performance, and energy efficiency, outperforming static and targeted refresh strategies in all evaluated workloads.

5. Future Work

In this paper, we have proposed a novel approach for mitigating Rowhammer attacks by dynamically predicting and adjusting RowSampling p-threshold using machine learning model. Our approach offers a dynamic, data-driven solution to the Rowhammer problem, improving the security of DRAM while minimizing the performance and energy costs typically associated with traditional refresh strategies.

The evaluation results demonstrate the effectiveness of our framework across various real-world workloads. The Multidimensional Predictor model provides superior Rowhammer attack mitigation compared to static and targeted refresh approaches, reducing the occurrence of bit flips by more than 80%. In terms of performance, our model incurs minimal overhead—just 2%—and reduces energy consumption by approximately 5%. This highlights the ability of our framework to strike a balance between security, performance, and energy efficiency, making it a promising solution for future memory systems.

The growing vulnerability of DRAM to attacks like Rowhammer necessitates innovative solutions that balance security with performance and energy efficiency. Our proposed framework represents a step forward in adaptive memory management, showing how machine learning can be used to improve memory system security without sacrificing system efficiency. By continuously updating refresh rates in response to real-time memory access patterns, our approach opens the door for more resilient and efficient memory architectures in future computing systems.

Conclusions

As DRAM technology continues to evolve, the need for adaptive, intelligent systems will only increase. Machine learning techniques, particularly ensemble models like Multidimensional Predictors, offer significant potential to enhance the security and performance of these systems, providing a path toward more secure and energy-efficient memory management in the face of increasingly sophisticated attacks.

As the technological process of embedded memory systems becomes smaller, we need to adapt our defense strategies. This paper presents new ways to deal with new threads using Multidimensional multilabel Predictor approach to enhance RowSampling defense from RowHammer. This shows that defense strategies can always be improved and enhanced with new technologies and with optimization. Also, this paper opens new ways for applying machine learning into software and hardware in the future.

References

[1] S. Bhattacharya and D. Mukhopadhyay, “Curious Case of Rowhammer: Flipping Secret Exponent Bits Using Timing Analysis,” in CHES, 2016, doi: 10.1007/978-3-662-53140-2_29.

[2] I. Bhati, Z. Chishti, S.-L. Lu, and B. Jacob, “Flexible Auto-Refresh: Enabling Scalable and Energy-Efficient DRAM Refresh Reductions,” in ISCA, 2015 pp. 235-246, doi: 10.1145/2749469.2750408.

[3] Jiang, H. Zhu, D. Sullivan, X. Guo, X. Zhang, and Y. Jin, “Quantifying Rowhammer Vulnerability for DRAM Security.” In DAC, San Francisco, 2021, pp. 73-78, doi: 10.1109/DAC18074.2021.9586119.

[4] A. G. Yağlikçi, M. Patel, J. Kim, R. Azizi, A. Olgun, L. Orosa, H. Hassan, J. Park, K. Kanellopoulos, T. Shahroodi, S. Ghose, and O. Mutlu, “BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows,” in HPCA, 2021, pp. 345-358, doi: 10.1109/HPCA51647.2021.00037.

[5] M. Kaczmarski, “Thoughts on Intel Xeon E5-2600 v2 Product Performance Optimisation,” 2014, url: https://slideplayer.com/slide/12161905.

[6] E. Bosman, K. Razavi, H. Bos, and C. Giuffrida, “Dedup Est Machina: Memory Deduplication as an Advanced Exploitation Vector,” in S&P, 2016, pp. 987-1004, doi: 10.1109/SP.2016.63.

[7] F. Brasser, L. Davi, D. Gens, C. Liebchen, and A.-R. Sadeghi, “Can’t Touch This: Practical and Generic Software-only Defenses Against RowHammer Attacks,” USENIX Security, 2017, doi: 10.48550/arXiv.1611.08396.

[8] L. Cojocar, K. Razavi, C. Giuffrida, and H. Bos, “Exploiting Correcting Codes: On the Effectiveness of ECC Memory Against RowHammer Attacks,” in S&P, 2019, pp. 55-71, doi: 10.1109/SP.2019.00089.

[9] P. Frigo, E. Vannacci, H. Hassan, V. van der Veen, O. Mutlu, C. Giuffrida, H. Bos, and K. Razavi, “TRRespass: Exploiting the Many Sides of Target Row Refresh,” in S&P, 2020, doi: 10.1109/SP40000.2020.00090.

[10] H. Hassa, Y. C. Tuğrul, J. S. Kim, V. V. K. Razavi, O. Mutlu “Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications” in ISCA, 2020, doi: 10.48550/arXiv.2110.10603.