

Exploring Error Bits for Memory Failure Prediction: An In-Depth Correlative Study

Qiao Yu^{*†}, Wengui Zhang[‡], Jorge Cardoso^{*¶}, and Odej Kao[†]

^{*}Huawei Munich Research Center, Germany, {qiao.yu, jorge.cardoso}@huawei.com

[†]Technical University of Berlin, Germany, odej.kao@tu-berlin.de

[‡]Huawei Technologies Co., Ltd, China, zhangwengui1@huawei.com

[¶]CISUC, University of Coimbra, Portugal

Abstract—In large-scale datacenters, memory failure is a common cause of server crashes, with Uncorrectable Errors (UEs) being a major indicator of Dual Inline Memory Module (DIMM) defects. Existing approaches primarily focus on predicting UEs using Correctable Errors (CEs), without fully considering the information provided by error bits. However, error bit patterns have a strong correlation with the occurrence of UEs. In this paper, we present a comprehensive study on the correlation between CEs and UEs, specifically emphasizing the importance of spatio-temporal error bit information. Our analysis reveals a strong correlation between spatio-temporal error bits and UE occurrence. Through evaluations using real-world datasets, we demonstrate that our approach significantly improves prediction performance by 15% in F1-score compared to the state-of-the-art algorithms. Overall, our approach effectively reduces the number of virtual machine interruptions caused by UEs by approximately 59%.

Index Terms—Memory, Failure prediction, AIOps, Uncorrectable error, Reliability, Machine Learning

I. INTRODUCTION

With the increasing demand for cloud computing and big data storage services, hardware failures [1], [2] can significantly impact the Reliability, Availability, and Serviceability (RAS)¹ of servers. Among hardware failures, DRAM (Dynamic Random Access Memory) failure is a major occurrence, accounting for 37% of total hardware failures in Figure 1. DRAM failure is often accompanied by DRAM errors, i.e., Correctable Error (CE) and Uncorrectable Error (UE). To mitigate DRAM failures, Error Correction Code (ECC) mechanisms such as SEC-DED [3], Chipkill [4] and SDDC [5] are used to detect and correct data corruption errors. For example, Chipkill ECC can correct any erroneous data bits originating from a single DRAM chip. However, when erroneous data bits span across two or more chips, the error correction capability of Chipkill ECC becomes overwhelmed, often resulting in a system crash due to a UE. Moreover, the ECC on contemporary Intel platforms like Skylake and Cascade Lake servers is less robust compared to Chipkill ECC, making it vulnerable to certain error-bit patterns from a single chip [6]. Thus, solely depending on ECC for DRAM reliability proves inadequate, with DRAM failures remaining a significant cause of system failures.

To improve memory reliability, several studies [8]–[13] have investigated the correlations between memory errors and

¹Reliability, Availability, and Serviceability are three key attributes assessing the dependability of computer systems.

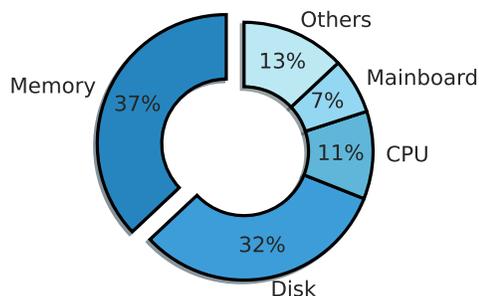


Fig. 1: Distribution of Hardware Failures in Data Centers [7].

failures, which forms the foundation of our work. Machine Learning (ML)-based techniques have been leveraged for DRAM failure prediction [14]–[21], using CEs information from a large-scale datacenter to predict UEs. These studies have effectively utilized the spatial distribution of CEs to enhance DRAM failure prediction. Moreover, system-level workload indicators such as memory utilization, read and write have been applied for DRAM failure prediction in [22]–[24]. The experiments in [24] have demonstrated that the workload metric is relatively less significant compared to other CE related features. In [25], CE storm (numerous CEs occurring in a short period) and UEs are considered for predicting DRAM-caused node unavailability (DCNU), emphasizing the importance of spatio-temporal CE features. Furthermore, in [6], specific error bit patterns are discussed and correlated with DRAM UEs. Rule-based error bit pattern indicators are developed for DRAM failure prediction across different manufacturers and part numbers, aligning with the ECC design of contemporary Intel Skylake and Cascade Lake servers. In addition, HiMFP framework [26] advocates a hierarchical system-level approach to memory failure prediction, using error bits features. *However, the intrinsic distributions of error bits, specifically in Data pins (DQ) and beat, remain unexplored in above literature. Delving into these distributions is crucial for understanding the correlation between CE and UE.*

In this paper, we present an in-depth correlative analysis between CE and UE, specifically focusing on the spatio-temporal distribution of error bits. We also investigate latent patterns of error bits from CE to UE on the ECC of contemporary Intel servers. Our primary goal of analysis is to enhance memory failure prediction based on various DRAM errors and

system configurations. Finally, machine learning models are implemented to leverage spatio-temporal error bits for memory failure prediction.

The key contributions of the paper are as follow:

- We analyze error bits patterns generated from DIMM manufacturer and part number, and construct novel temporal risky CE indicators for UE prediction.
- We conduct the first in-depth correlative analysis between error bits and UE, specifically during DRAM read/write in Data pins (DQ) and beat. In addition, micro-level faults in the memory subsystem and system configurations are further correlated with UE occurrences.
- We design ML-based failure prediction algorithms, based on the statistical insights from our analyses. Through evaluations using real-world data from a large-scale data center, our proposed error bits features have demonstrated the ability to capture latent patterns within the ECC of contemporary Intel servers, significantly improving UE prediction. When compared to the state-of-the-art algorithm [6], our approach achieves up to an 15% improvement in F1-score for UE prediction, resulting in approximately a 59% Virtual Machine Reduction Rate (VIRR) in our data centers.

The remainder of this paper is organized as follows: Section II provides the background of our work. Section III discussed the dataset employed in our data analysis. In Section IV, we formulate the the problem and define the performance measurements. Section V introduces error bit pattern indicators for UE prediction. Section VI presents an correlative study on UE. Section VII demonstrates machine learning techniques for memory failure prediction. Experimental results are shown in Section VIII. Section IX concludes this paper.

II. BACKGROUND

A. Terminology

A *fault* serves as the underlying cause of an error in DRAM, and it can be caused by various factors such as particle impacts, cosmic rays or defects.

An *error* refers to the situation in which a DIMM provides data to the memory controller that is inconsistent with the ECC [3]–[5], [27], resulting from an active fault. Depending on ECC’s capability to correct them, memory errors can be classified into correctable errors (CEs) and uncorrectable errors (UEs) [12]. Two specific types of UEs are well-studied in prior literature [14]. 1) *sudden UE*: UEs caused by some component faults that instantly corrupt data, and 2) *predictable UE*: UEs that initially manifest as correctable errors but eventually escalate into UEs. A sudden UE typically has no CEs before it occurs, while a predictable UE can be predicted using CEs with failure prediction algorithms.

B. Memory Organization and Access

Figure 2 illustrates a framework of memory subsystem organization, memory access and memory RAS. The memory system is hierarchical in Figure 2(1): A DIMM rank is composed of several DRAM chips that form banks of two-dimensional

arrays. Each bank is organized into rows and columns, and each addressable unit indexed by rows and columns is a memory cell containing a 4-bit word in the x4 DRAM device. Data flow in this architecture is transmitted from the cell to memory controller, which can generally detect and correct CEs via channels. Figure 2(2) depicts the transmission process of x4 DRAM Double Data Rate 4 (DDR4) chips via DQs. Upon initiating a data request, 8 beats each with 72 bits (64 data bits and 8 ECC bits) including ECC error codes are transferred to memory controller via DQ wires. Implementing the contemporary ECC [6], [27], 72-bit data are spread across 18 DRAM chips, allowing the memory controller to detect and correct them with ECC in Figure 2(3). Note that ECC checking bits addresses are decoded to locate specific errors in DQs and beats. Then, all these logs including error detection and correction, events, and memory specifications are archived in Baseboard Management Controller (BMC)² in Figure 2(4). Among previous works [6], [15]–[19], [25], [26], error bits in a cell have not been extensively examined. In our work, we conduct the first in-depth correlative analysis between error bits and UE in the field, to unveil the latent patterns of memory UEs.

C. DRAM RAS Techniques

DRAM subsystems are typically protected by RAS features in Figure 2(6). Proactive early VM live migrations can greatly reduce VM interruptions by moving VMs without service interruption. The CE storm suppressed mechanism helps avoid service degradation caused by CE storm³. Advanced RAS techniques are designed to protect server-grade machines include the avoidance of fault regions. On the hardware technologies, sparing mechanisms are employed, such as bit sparing (e.g., Partial Cache Line Sparing (PCLS) [28]), row/column sparing (e.g., Post Package Repair (PPR) [29]), bank/chip sparing (e.g., Intel’s Adaptive Double Device Data Correction (ADDDC) [30], [31]), etc. On the software-sparing mechanisms, such as the page offlining in operating systems, can also be applied to avoid memory errors [30], [32], [33]. However, these techniques often require higher redundancy and entail additional overhead, which can potentially impact system performance. Hence, these techniques cannot be universally adaptable across all machines. Utilizing memory failure prediction allows for the prediction of UEs and the activation of corresponding mitigation techniques based on specific use cases.

III. DATASET

Our dataset was obtained from the Baseboard Management Controller (BMC) of a large-scale datacenter, which includes system configuration, Machine Check Exception (MCE) log [34], and memory events. We focus on DIMMs with CEs, excluding those with sudden UEs from our datasets due to

²BMC is a dedicated processor integrated into server’s motherboard, tasked with monitoring the physical state of a computer, network server, or other hardware device.

³CE interruptions repeatedly occur multiple times, e.g., 10 times.

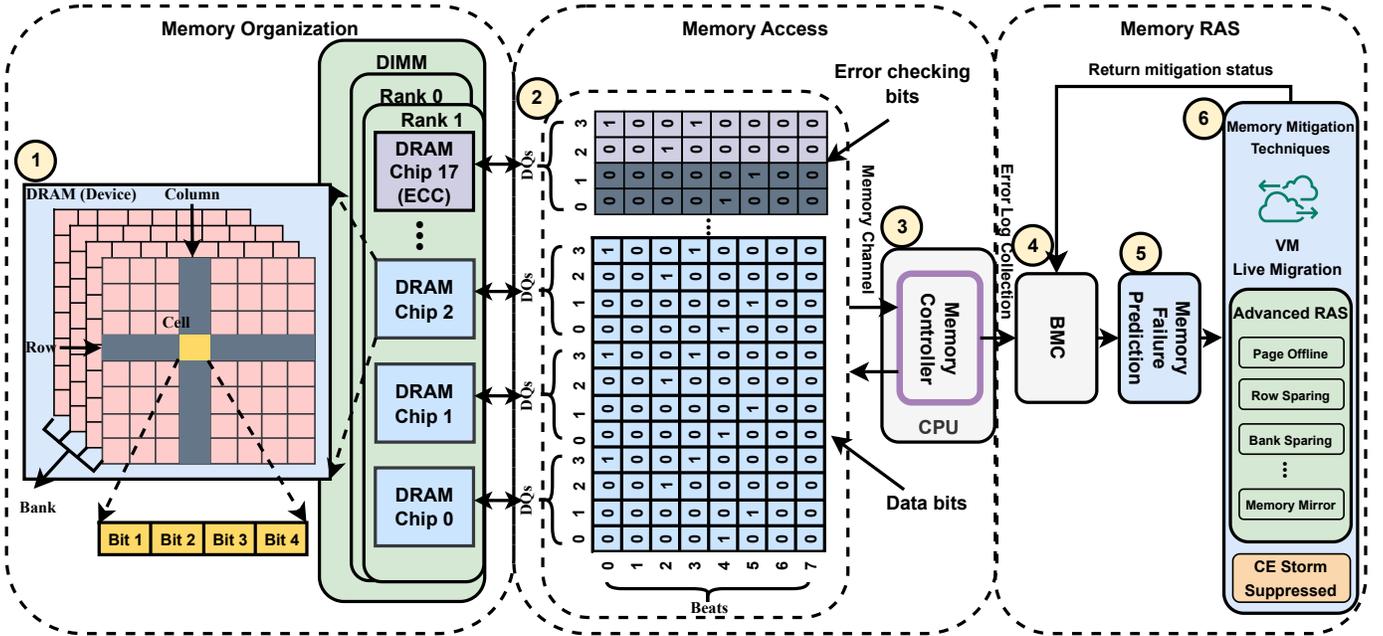


Fig. 2: Illustration of Memory Failure Prediction Framework.

TABLE I: Description of dataset.

Dataset	Timespan	DIMMs with CEs	DIMMs with UEs
Train set	9 months	> 80,000	> 2,000
Test set	3 months	> 30,000	> 1,000

a lack of prediction information. The MCE log records both CE and UE, providing details about memory error addresses (e.g., rank, bank, column) and DIMM specifications (e.g., manufacturer, capacity). We examined error logs from approximately 200,000 servers with Intel Skylake (Launched in 2017), Cascade Lake (Launched in 2019), Cooperlake (Launched in 2020) and Icelake (Launched in 2021) architectures in the datacenter.

Table I provides an overview of the collected data. For the training set, we gathered over 80,000 Double Data Rate 4 (DDR4) DIMMs, spanning different manufacturers and part numbers, with CEs recorded from January to September 2022. Among them, we observed over 2,000 DIMMs with UEs, with 71% of UE DIMMs having preceding CEs and 29% are sudden UEs. Using a consistent collection approach, we prepared over 30,000 DIMMs for the test set from October to December 2022. This test set included over 1,000 DIMMs with UEs, with 67% of UE DIMMs having preceding CEs and 33% are sudden UEs. We conducted our correlative analysis and algorithm training based on the train set. The test set is reserved for final evaluation in Section VIII.

IV. PROBLEM FORMULATION AND PERFORMANCE MEASURES

The failure prediction problem is formulated as a binary classification problem [26]. As illustrated in Figure 3, at present t , an algorithm observes historical data from an *observation window* Δt_d to predict failures within the prediction period $[t + \Delta t_l, t + \Delta t_l + \Delta t_p]$, where Δt_l is a minimum time interval between the prediction and the failure. Δt_p denotes

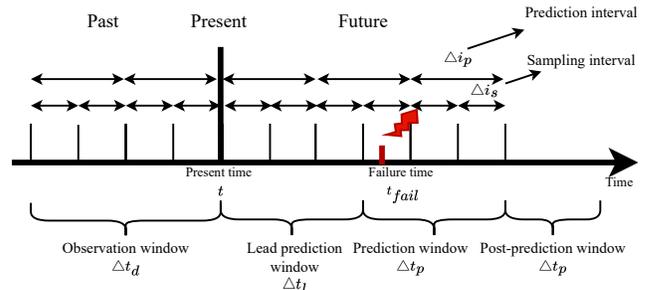


Fig. 3: Failure prediction problem definition [26].

the prediction interval. Online event samples are taken every Δi_s , e.g., CE events are logged every minute. Predictions run at 5-minute intervals Δi_p . Observation and prediction windows are set at 5 days (Δt_d) and 30 days (Δt_p) respectively, enabling proactive measures. Note that these parameter settings were derived from an empirical analysis in the production environment. The *lead prediction window* $\Delta t_l \in (0, 3h]$ varies based on production use cases. A True Positive (TP) is a correctly predicted failure within the prediction window, while a False Positive (FP) is an incorrect prediction. A failure without a prior alarm is a False Negative (FN), and a True Negative (TN) occurs when no failures are predicted or occur. We assess the algorithm using $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$ and $F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$.

VM Interruption Reduction Rate (VIRR). Previous works [6], [17], [19], [25], [26] have proposed cost-aware models to measure the benefits of memory failure prediction. In this work, we focus on *VM Interruption Reduction Rate (VIRR)* [26] as it more accurately reflects the impact on customers.

To understand VIRR, consider V_a as the average number of VMs in a server. In a scenario devoid of prediction, the interruptions are defined as $V = V_a(TP + FN)$. Even though proactive VM live migrations can reduce VM interruptions without service interruption, a notable fraction of

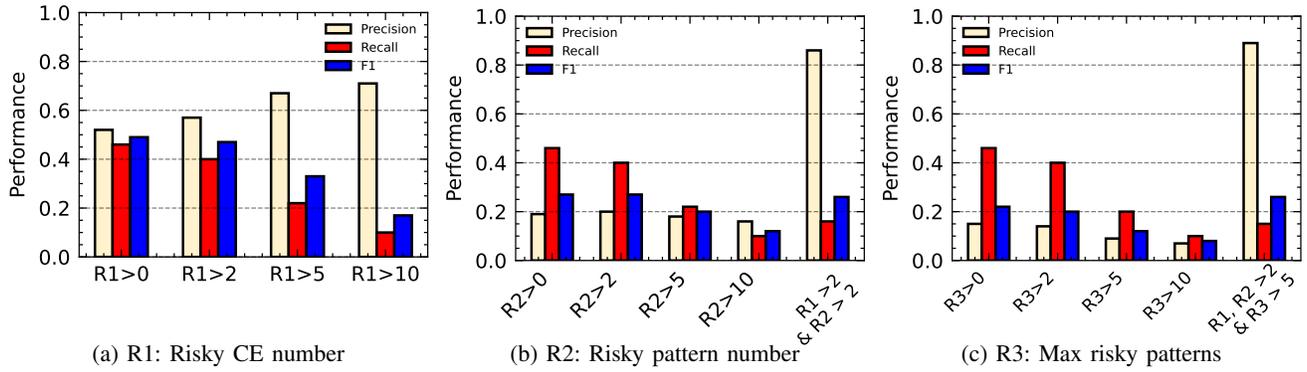


Fig. 4: Performance analyses of risky CE patterns.

VMs may still experience cold migration, which generally interrupts VMs. This cold migration typically ensues when live migrations cannot be applied, either due to a paucity of resources or unforeseen failures. Given that cold migration is a prevalent strategy for both VM relocation and maintenance. The percentage of such migration is represented as y_c . Therefore, we define $V'_1 = V_a \cdot y_c(TP + FP)$ as the number of VM interruptions arising from cold migrations initiated by positive failure predictions (TP + FP). On the other side, any missed failure predictions invariably escalate the interruptions, represented by $V'_2 = V_a \cdot FN$. The overall interruptions after factoring in the prediction algorithm sum up to $V' = V'_1 + V'_2$. The formula to measure VIRR is thus: $VIRR = \frac{V - V'}{V}$. Simplifying this give us $(1 - \frac{y_c}{precision}) \cdot recall$ as derived in [26].

In real-world production environments, y_c retains a positive value as VMs can be cold migrated due to the failure of live migration or memory recovery. If a model's precision dips below the percentage of cold migration ($precision < y_c$), the VIRR becomes negative, indicating an increase in VM interruptions. In contrast, models with high precision consistently yield a positive VIRR, and this is further amplified by the recall. Based on our observations in the production environment, we have defined $y_c = 0.1$ for our evaluation. Note that this value is already pessimistic, as the cloud infrastructure continues to expand, leading to a decrease in y_c over time.

V. TEMPORAL RISKY CE PATTERN INDICATORS

According to a recent study by Intel [6], ECCs in modern Intel server platforms do not fully cover every potential errors from a single chip. Although Intel keeps the exact ECC algorithms confidential and undisclosed, they have provided some general information on error-bit patterns that can be fully correctable, partially correctable and potential risky in [6], [30], [35]. For example, as shown in Figure 2(2), a DIMM with x4 DRAMs provides 32 error checking bits across 4 DQs and 8 beats during memory access. In a specific ECC outlined in [35], if all the actual erroneous bits are bounded within the half of the bitmap (highlighted in gray in the error checking bits), that error is guaranteed to be correctable. Otherwise, it

is risky. More publicly available examples of error bit patterns can be found in [6], [30], [35].

In this paper, we also obtain coarse-grained error-bit patterns, such as risky error bit patterns that are more likely to encounter UEs on contemporary Intel servers. CEs with risky error bit patterns are prone to evolve to UEs that cannot be corrected by the modern ECC algorithm [6]. We introduce three temporal risky pattern indicators as follows:

- **R1: Risky_CE_Cnt:** The number of unique CEs that match at least one risky error-bit pattern in a 24-hour period;
- **R2: Risky_Pattern_Cnt:** Total number of matched risky error-bit patterns in a 24-hour period;
- **R3: Max_Risky_Pattern_Cnt:** Maximum number of unique matched risky error-bit patterns counted in a 24-hour period;

While R1 is similar to the indicator in [6], R2 and R3 are novel pattern indicators proposed in this work. We compare the performance of these three indicators in Figure 4. As shown in Figure 4(a), when the count of risky CEs is greater than 0 (indicating at least one risky CE), it achieves 52% precision, 46% recall and 49% F1-score on the training set. As the count of risky CEs increases, the precision also increases accordingly. However, the recall drops, indicating that most DIMMs with UE originate from a small number of risky CEs. This is intuitive that We evaluate the performance of R2 indicator in Figure 4(b) and observe that its performance does not increase linearly as the count of matched risky patterns increases. However, when combining $R1 > 2$ and $R2 > 2$, the precision increases significantly to 86%. On the other hand, individual R3 does not perform well on its own in Figure 4(c). However, when combined with R1 and R2, it improves precision to the highest value of 89%. Therefore, combining different pattern indicators can effectively enhance performance, which motivates us to use machine learning to integrate all indicators and correlated features, aiming to further improve UE prediction. Additionally, the risky patterns originate from the distribution of error bits in Data pins (DQs) and beats. We delve deeper into investigating the spatial and temporal distribution of error bits in DQs and beats in Section VI-A.

Finding 1. *The performance of an individual risky CE pattern is limited. However, the proper combination of risky CE pattern indicators can significantly improve the results, particularly precision.*

VI. CORRELATIVE ANALYSIS BETWEEN UNCORRECTABLE ERROR AND VARIOUS FACTORS

We start with the high-level of correlative study between UE and various factors. Specifically, we investigate the relationship among error bits, DRAM faults, and system configurations to gain insights into their influence on UE occurrences. This analysis is essential for identifying relevant features that can be used for model training and failure prediction as outlined in Section VII. Our methodology follows a similar approach to previous studies [9], [11], [20]. We employ a calculation method named as relative UE rate, where DIMMs are grouped based on specific characteristics (e.g., server age), and the fraction of DIMMs experiencing UEs is determined. The relative UE rates are normalized within the range [0, 1], enabling us to observe trends, compare rates, and finally extract important features for UE prediction.

A. Correlative Analysis Between Error Bits and UE

We first examine the relative UE rate based on characteristic of error bits. To quantify this, we first calculate the total number of error bits and the number of adjacent error bits within a single CE event. For a specific memory access, Figure 5 visualizes bitmap of error bits occurring in four DQs and four beats. In this example, there are total six error bits and one pair of adjacent error bits. Figure 6 illustrates the correlation between the total number of error bits and the UE rate. As the count of error bits increases, the UE rate generally rises. However, the overall relative UE rates remain relatively low.

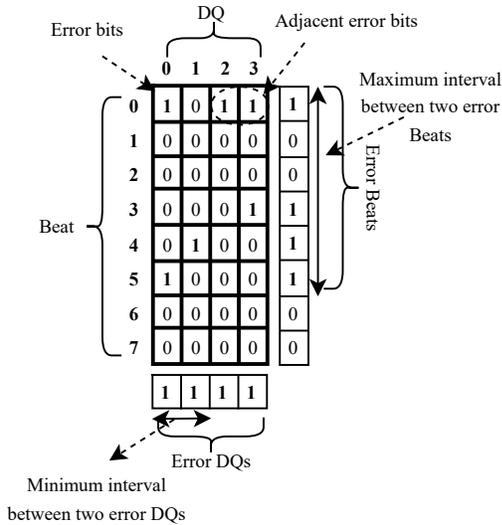


Fig. 5: Spatial correlations of error bits in DQs and Beats.

A finding emerges when comparing the relevance of adjacent error bits with the total number of error bits. The occurrence of adjacent error bits within a specific range, such as greater than 0 or 5, is more strongly associated with the

occurrence of UEs. This implies that even a small number of adjacent bits can have a risk for UE occurrence.

Finding 2. *In terms of UE occurrence, the total number of error bits exhibits weaker correlation compared to adjacent error bits. Even a small number of adjacent bits can lead to UE occurrence.*

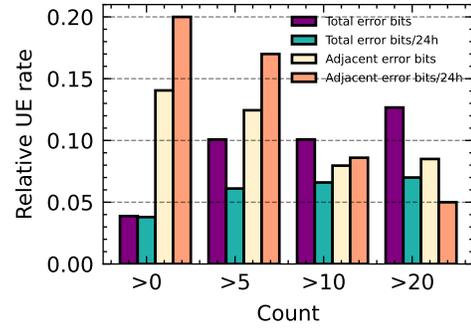


Fig. 6: Error bits analysis.

We then investigate the spatial distribution of error bits in DQs and beats. As shown in Figure 5, we have calculated the number of error DQs and beats, which yields four error DQs and four error beats. We also examine other key features such as the interval between error DQs and the interval between error beats.

The correlative analysis of these spatial features is presented in Figure 7. In Figure 7(a), error DQs with two, three, or four generally exhibit higher UE rates compared to those CEs with only one error DQ. Similarly, Figure 7(b) indicates that multiple error beats have higher UE rates compared to one error beat. Error bits occurring in more than one DQ and beat are more likely to encounter UEs. Additionally, our analysis reveals an important observation regarding the interval between error DQs and beats. Specifically, we found that the error DQs interval of three exhibit a relatively lower UE rate compared to other intervals. On the other hand, beats interval of four have the highest UE rate compared to other intervals. These insights highlight the importance of considering the specific intervals between error DQs and beats in understanding the occurrence of UE.

In addition to spatial correlative analysis of error bits in DQs and beats, we incorporate temporal information into these features. As shown in Figure 8, error bits are propagated through DQs and beats over time in t , which can be increased from a single one in t_0 to multiple bits spanning across DQs and beats, eventually lead to UE. Note that error bits of UEs are typically unknown, since they are not correctable, typically leading to service down without logging error addresses. In terms of spatial distribution, a single CE event in t_{i-1} may involve 2 error DQs within the same beat. However, in the case of multiple CEs within an interval Δt_i , there could be three error DQs spanning across 2 beats. To capture these spatio-temporal bits patterns, we calculate statistical features such as *Sum*, *Maximum*, *Minimum*, *Average* and *Standard deviation* of error bits in DQs and beats based on all CE events within the aggregation window Δt_i . Additionally, we analyze spatio-temporal features of DQ and beat counts and

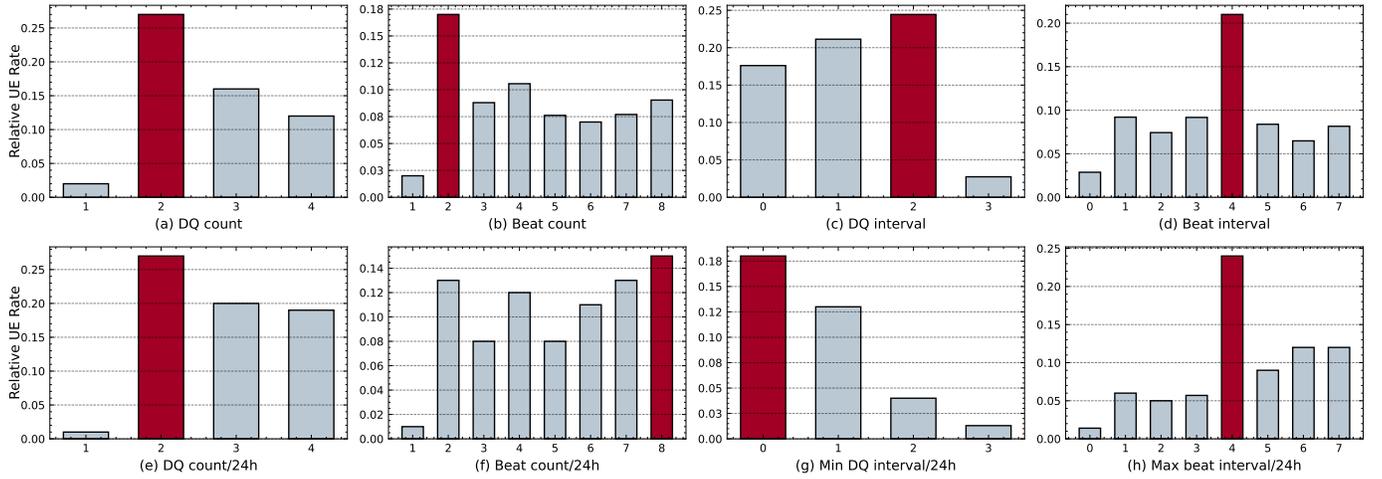


Fig. 7: Analyses of spatial and temporal error bits: Highlighting the highest rate with red bar.

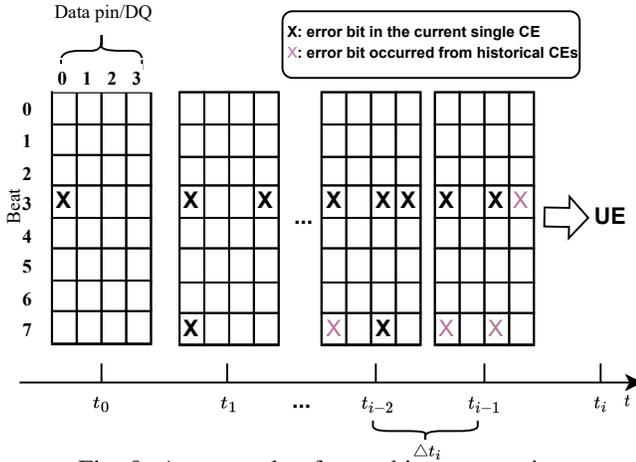


Fig. 8: An example of error bits propagation.

intervals within a 24-hour aggregation window in Figure 7(e)-(h). While the relative UE rates for temporal error DQs and beats in Figure 7(e) and (f) are vary with Figure 7(a) and (b) respectively, the consistent trend remains that one error DQ or beat has a lower relative UE rate compared to multiple error DQs and beats. Furthermore, the minimum error DQ interval and the maximum error beat interval within 24 hours exhibit different relative UE rates in Figure 7 (g) and (h). Among all error DQ intervals, the interval of 3 consistently exhibits the lowest UE rate. On the other hand, among all beat intervals, the interval of 4 demonstrates the highest UE rate.

Finding 3. *Our analyses reveal that both spatial and temporal error bits in DQs and beats play a significant role in distinguishing UE occurrences. This finding suggests that these features can serve as important indicators for UE prediction.*

Therefore, we generate both spatial error bits features in a single CE and spatio-temporal error bits features across multiple CE events for UE prediction. Even features with relatively low UE rates may still contribute significantly when utilized in conjunction with machine learning techniques for UE prediction. We conduct feature selection and UE prediction based on machine learning in Section VII.

B. Correlative Analysis Between DRAM Faults and UE

CEs can originate from various components within the memory subsystem, as depicted in Figure 2(1). To examine the impact of different component faults on memory failure, which ultimately leads to the generation of error bits during memory access as shown in Figure 2(2). We consider DIMM-level of components' faults from cell, column, row, bank, device and rank respectively. If the number of CEs repeated in the same cell reaches a predefined threshold θ_{cell} , it refers to **Cell fault**. If CEs scattered along in a row and a column reaches θ_{row} and θ_{column} , they are **Row fault** and **Column fault** respectively. **Bank fault** refers to the case where row faults and column faults both are greater than θ_{bank} in the same bank. More than θ_{device} of unique bank faults occurred in a device indicates **Device fault**. **Rank fault** represents that device faults reach a predefined threshold θ_{rank} in the same rank. We defined $\{\theta_{cell}, \theta_{row}, \theta_{column}, \theta_{Device}, \theta_{Rank}\} = 2$ and $\theta_{bank} = 3$ in our analyses.

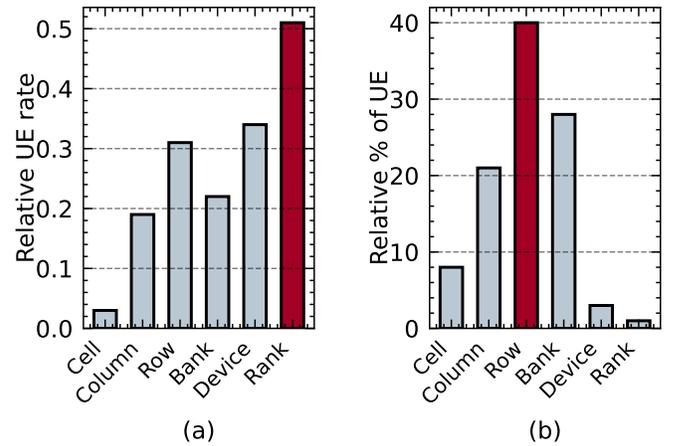


Fig. 9: Micro-level components' fault analysis.

We first examine each component fault by excluding the higher-level faults. For example, As shown in Figure 9, Cell faults exhibit a UE rate of less than 0.2. However, when cell faults accumulate and propagate to higher levels of DRAM

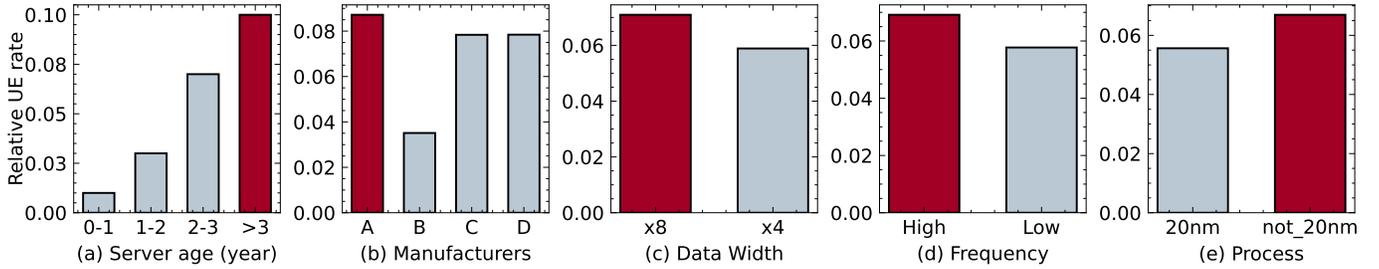


Fig. 10: System configurations analyses.

components, relatively 0.31 of UE rate associated with row faults (excluding column and higher-level faults, such as bank, device and rank), and 0.22 of UE rate associated with bank faults (excluding device and rank faults). We also visit relative percentage of UE in each component fault. Although Device and Rank faults have a higher relative UE rate, the proportion of UEs associated with these faults is relatively small compared to Row and Bank faults.

Finding 4. *While higher-level faults may have a higher likelihood of causing UEs, Row and Bank faults account for the majority of UEs in the system. This emphasizes the importance of addressing and mitigating Row and Bank faults to improve the overall reliability and performance of the memory subsystem.*

C. Correlative Analysis Between System Configuration and UE

In our study, we examine the correlation between system configurations and UEs. We first analyze the server age, and our findings align with our conjecture that older servers are more likely to experience memory failures. Figure 10(a) shows that servers with more than 2 years of age have a higher UE rate.

Furthermore, we investigated various DRAM hardware configurations, including manufacturer, capacity, device data width, frequency, and process in Figure 10. To protect the confidentiality of manufacturer names, we anonymized them as manufacturers A to D, representing the four major DIMM manufacturers in our data centers. Different manufacturers exhibited varying UE rates, potentially due to differences in DIMM processes.

We also observe that DIMMs with x8 bit width have a higher relative UE rate compared to those with x4 bit width. This difference may be attributed to variations in memory access and ECC correction. Additionally, higher DRAM frequency generally correlates with higher relative UE rates. We further examine the DRAM process, categorizing them as either 20nm or not (as the exact processes of 1nm, 1xnm, and 1znm are proprietary information). The not 20nm process category shows a higher relative UE rate.

The capacity of the DIMM did not significantly impact the UE rate in our study.

Finding 5. *The UE rate varies across server age, manufacturers, data width, frequency and process, while we did not*

observe significant differences in the UE rate based on the capacity of the DIMM.

These attributes, including server age, manufacturer, data width, frequency, and process, can be valuable for failure prediction in Section VII.

VII. FAILURE PREDICTION

In this section, we design memory failure prediction based on pattern indicators (Section V) and correlative analysis between UE and various factors (Section VI).

We develop failure prediction mainly using machine learning techniques, e.g., Random Forest [25], XGboost [25], LightGBM [20] and AdaUboost [19], since these ensemble learning techniques have been widely used in previous memory failure prediction literature [14], [17]–[19], [24] due to their fast learning and good performance. The experimental results of these models are presented in Section VIII.

Labeling method: Our prediction framework categorizes samples into two classes: Positive and Negative. DIMMs expected to encounter at least one UE within the prediction window are categorized as **Positive**, whereas those not expected to experience any UE are termed **Negative**.

Positive samples are labeled based on the time interval t_i between a CE and its subsequent UE. Selected intervals for t_i include 6 hours, 24 hours, 72 hours, 120 hours, 1 month, and a DIMM’s entire lifetime. CEs that fall within the 0 to t_i interval preceding a UE are marked as **Positive samples**. CE events outside this period are excluded to prevent mislabeling. All CE events from healthy DIMMs are labeled as **Negative samples**. However, our training data experiences from class imbalance, we employ over-sampling strategies for positive samples, ensuring models adequately address both classes.

Feature generation. We categorize features into six groups including:

- *Static Features* describe DIMM characteristics studied in Section VI-C including server age, manufacturers, data width, frequency and chip process.
- *CE error rate* refer to the number of CEs and their occurrence frequency, e.g., error counts of all CEs within the predefined time.
- *DQ-Beat Error Bits* features refer to the spatial and temporal distribution of error bits in DQs and beats, as discussed in Section VI-A.
- *Error bit Patterns* features are derived from three risky CE pattern indicators described in Section V.

- *Fault Counts* refers to the cumulative number of components’ faults (cell, row, column, bank, device and rank) within t_i , derived from study in Section VI-B.
- *Memory Events* refers to CE storm³, CE overflow⁴, CE storm suppressed notification⁵, etc, which indicate the unhealthy status of memory.

Totally, six groups of features are constructed as input of machine learning approaches. We select the best features using *Pearson correlation*, *Random Forest* and *LightGBM* in Section VIII.

VIII. RESULT

After empirical experiments on the training set, we explored the parameter t_i ranging from 1 minute to 5 days. The output probability threshold was set to 0.3, since it can achieve the best VIRR with a predefined $y_c = 0.1$. To evaluate the importance of designed features, three feature selection approaches are implemented in Table II. Among the top five important features identified by these approaches, four out of five are related to error bits, highlighting the significance of error bits in predicting UEs. Notably, *Minnum error DQ interval* consistently ranked as the most important feature across all approaches. To determine the best feature set for algorithm training, we employed recursive feature elimination and feature importance ranking. Table III displays the results, demonstrating that LightGBM outperformed other machine learning techniques with a F1-score of 0.64 on the test set. Consequently, we selected LightGBM for further analysis in our study.

TABLE II: Rankings of the top five important features.

Rank	Pearson	Random Forest	LightGBM
1	Min_DQ_interval	Min_DQ_interval	Min_DQ_interval
2	Max_beat_interval	Error_DQ_counts_24h	Fault(Cell)
3	Risky_CE_Cnt	CE overflow	Risky_CE_Cnt
4	Risky_Pattern_Cnt	Max_adjacent_bits_24h	Risky_Pattern_Cnt
5	Fault(Row)	Error_beat_Cnt	Error_DQ_counts_24h

Comparison with existing approaches. We further evaluate the significant of our proposed error bits features by comparing with existing the state-of-the-art approaches presented in [6]. Specifically, we reproduced their rule-based approaches as discussed in Section V and apply the same experimental setup on our dataset. Note that the approaches in [6] are designed with various part numbers of manufacturers, but the detail was not disclosed in their work. We evaluated their approaches without differentiating the part numbers. The results in Table IV demonstrate that our approach significantly achieves higher F1-score of 0.64 by including all features. In addition, our algorithm still achieves relatively better performance by excluding the error bits patterns features, which indicates the superior of error bits features in UE prediction. By excluding both error bits and pattern features, algorithm cannot perform well, which further prove the significant of error bits information for UE prediction.

⁴CE counts reach an initial overflow threshold.

⁵The mechanism suppresses and notifies if a CE storm occurs several times in the same DIMM.

TABLE III: Performance of ML algorithms.

Algorithms	Precision	Recall	F1-Score
Random Forest	0.63	0.62	0.63
XGBoost	0.54	0.67	0.59
AdaUboost	0.54	0.78	0.64
LightGBM	0.53	0.82	0.64

TABLE IV: Comparison with existing approaches.

Algorithms	Precision	Recall	F1-Score
Risky_CE Pattern	0.53	0.46	0.49
Risky_CE Pattern \wedge Column	0.68	0.10	0.17
Risky_CE Pattern \wedge Bank	0.84	0.11	0.19
Ours (Excluding error bits and patterns)	0.30	0.51	0.38
Ours (Excluding patterns)	0.45	0.74	0.56
Ours (All features)	0.53	0.82	0.64

Finding 6. *The inclusion of error bits features significantly enhances UE prediction performance, even without knowledge of the error bits patterns. This alludes that the latent patterns of error bits can be predicted using spatio-temporal error bits features.*

Lead time. In Table V, we also examine the prediction results for three lead times. The lead time refers to the duration between the prediction time and the expected occurrence of a failure. Depending on the memory mitigation techniques, these lead times can vary. For instance, in a 15-minute lead time allows VM migration to a backup system and the deployment of advanced RAS techniques to prevent UE incidents. With a 1-hour lead time, VM migration may span up to an hour due to the workload involved, and failing machines can be localized and replaced with the corresponding DIMM. The VM Interruption Reduction Rate (VIRR) discussed in Section IV is estimated for these lead times. In our datacenters, we take into consideration a 15-minute lead time, which results in a reduction of approximately 59% in VM interruptions caused by UEs.

TABLE V: Performance in different lead times.

Lead time	Precision	Recall	F1-Score	VIRR
1s	0.53	0.82	0.64	0.67
15m	0.46	0.75	0.57	0.59
1h	0.36	0.45	0.40	0.33

IX. CONCLUSION

We present an in-depth correlative analysis on uncorrectable errors with various factors, particularly focusing on spatio-temporal error bits information of CEs. We report 6 findings from our analyses and failure prediction studies. Through evaluations using real-world datasets, we demonstrate that our approach significantly improves prediction performance by 15% in F1-score compared to the state-of-the-art algorithms. Overall, it can reduce VM interruptions by around 59% VIRR in the datacenter. In the future, we plan to extend our algorithm to include servers from different manufacturers’ platforms, particularly focusing on the comparisons of Chipkill and non-Chipkill ECC servers.

ACKNOWLEDGEMENT

We thank the anonymous reviewers from ICCAD’23 for their great comments.

REFERENCES

- [1] G. Wang, L. Zhang, and W. Xu, "What can we learn from four years of data center hardware failures?" in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2017, pp. 25–36.
- [2] P. Notaro, Q. Yu, S. Haeri, J. Cardoso, and M. Gerndt, "An optical transceiver reliability study based on sfp monitoring and os-level metric data," in *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, 2023, pp. 1–12.
- [3] M. Y. Hsiao, "A class of optimal minimum odd-weight-column sec-ded codes," *IBM Journal of Research and Development*, 1970.
- [4] T. J. Dell, "A white paper on the benefits of chipkill correct ecc for pserver main memory," in *Computer Science*, 1997. [Online]. Available: <https://asset-pdf.scinapse.io/prod/48011110/48011110.pdf>
- [5] "Intel® e7500 chipset mch intel® x4 single device data correction (x4 sdcc) implementation and validation." [Online]. Available: <https://www.intel.com/content/dam/doc/application-note/e7500-chipset-mch-x4-single-device-data-correction-note.pdf>
- [6] C. Li, Y. Zhang, J. Wang, H. Chen, X. Liu, T. Huang, L. Peng, S. Zhou, L. Wang, and S. Ge, "From correctable memory errors to uncorrectable memory errors: What error bits tell," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '22. IEEE Press, 2022.
- [7] "Intel MCA+MFP Helps JD Stable and Efficient Cloud Services." [Online]. Available: <https://www.intel.com/content/dam/www/central-libraries/us/en/documents/memory-failure-prediction-at-jd-cloud-us.pdf>
- [8] B. Schroeder, E. Pinheiro, and W.-D. Weber, "Dram errors in the wild: A large-scale field study," in *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 193–204. [Online]. Available: <https://doi.org/10.1145/1555349.1555372>
- [9] J. Meza, Q. Wu, S. Kumar, and O. Mutlu, "Revisiting memory errors in large-scale production data centers: Analysis and modeling of new trends from the field," in *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. Rio de Janeiro, Brazil: IEEE, 2015, pp. 415–426.
- [10] V. Sridharan and D. Liberty, "A study of dram failures in the field," in *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2012, pp. 1–11.
- [11] V. Sridharan, N. DeBardeleben, S. Blanchard, K. B. Ferreira, J. Stearley, J. Shalf, and S. Gurumurthi, "Memory errors in modern systems: The good, the bad, and the ugly," in *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 297–310. [Online]. Available: <https://doi.org/10.1145/2694344.2694348>
- [12] M. V. Beigi, Y. Cao, S. Gurumurthi, C. Recchia, A. Walton, and V. Sridharan, "A systematic study of ddr4 dram faults in the field," in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2023, pp. 991–1002.
- [13] M. Patel, T. Shahroodi, A. Manglik, A. G. Yaglikci, A. Olgun, H. Luo, and O. Mutlu, "A case for transparent reliability in dram systems," 2022.
- [14] I. Giurgiu, J. Szabo, D. Wiesmann, and J. Bird, "Predicting DRAM reliability in the field with machine learning," in *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Industrial Track*, ser. Middleware '17. New York, NY, USA: Association for Computing Machinery, Dec. 2017, pp. 15–21, 00026. [Online]. Available: <https://doi.org/10.1145/3154448.3154451>
- [15] X. Du and C. Li, "Memory failure prediction using online learning," in *Proceedings of the International Symposium on Memory Systems*, ser. MEMSYS '18. New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 38–49, 00000. [Online]. Available: <https://doi.org/10.1145/3240302.3240309>
- [16] X. Du, C. Li, S. Zhou, M. Ye, and J. Li, "Predicting Uncorrectable Memory Errors for Proactive Replacement: An Empirical Study on Large-Scale Field Data," in *2020 16th European Dependable Computing Conference (EDCC)*, Sep. 2020, pp. 41–46, 00003 ISSN: 2641-810X.
- [17] I. Boixaderas, D. Zivanovic, S. Moré, J. Bartolome, D. Vicente, M. Casas, P. M. Carpenter, P. Radojković, and E. Ayguadé, "Cost-aware prediction of uncorrected DRAM errors in the field," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '20. Atlanta, Georgia: IEEE Press, Nov. 2020, pp. 1–15, 00005.
- [18] F. Yu, H. Xu, S. Jian, C. Huang, Y. Wang, and Z. Wu, "Dram failure prediction in large-scale data centers," in *2021 IEEE International Conference on Joint Cloud Computing (JCC)*. Los Alamitos, CA, USA: IEEE Computer Society, aug 2021, pp. 1–8. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/JCC53141.2021.00012>
- [19] X. Du and C. Li, "Predicting uncorrectable memory errors from the correctable error history: No free predictors in the field," in *The International Symposium on Memory Systems*, ser. MEMSYS 2021. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: <https://doi.org/10.1145/3488423.3519316>
- [20] Z. Cheng, S. Han, P. P. C. Lee, X. Li, J. Liu, and Z. Li, "An in-depth correlative study between dram errors and server failures in production data centers," in *2022 41st International Symposium on Reliable Distributed Systems (SRDS)*, 2022, pp. 262–272.
- [21] J. Bogatinovski, O. Kao, Q. Yu, and J. Cardoso, "First ce matters: On the importance of long term properties on memory failure prediction," in *2022 IEEE International Conference on Big Data (Big Data)*, 2022, pp. 4733–4736.
- [22] X. Sun, K. Chakrabarty, R. Huang, Y. Chen, B. Zhao, H. Cao, Y. Han, X. Liang, and L. Jiang, "System-level hardware failure prediction using deep learning," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, Jun. 2019, pp. 1–6, iSSN: 0738-100X.
- [23] L. Mukhanov, K. Tovletoglou, H. Vandierendonck, D. S. Nikolopoulos, and G. Karakonstantis, "Workload-aware dram error prediction using machine learning," in *2019 IEEE International Symposium on Workload Characterization (IISWC)*, 2019, pp. 106–118.
- [24] X. Wang, Y. Li, Y. Chen, S. Wang, Y. Du, C. He, Y. Zhang, P. Chen, X. Li, W. Song, Q. xu, and L. Jiang, "On workload-aware dram failure prediction in large-scale data centers," in *2021 IEEE 39th VLSI Test Symposium (VTS)*, 2021, pp. 1–6.
- [25] P. Zhang, Y. Wang, X. Ma, Y. Xu, B. Yao, X. Zheng, and L. Jiang, "Predicting dram-caused node unavailability in hyper-scale clouds," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2022, pp. 275–286.
- [26] Q. Yu, W. Zhang, P. Notaro, S. Haeri, J. Cardoso, and O. Kao, "Himfp: Hierarchical intelligent memory failure prediction for cloud service reliability," in *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2023, pp. 216–228.
- [27] "Memory ras configuration user's guide." [Online]. Available: https://www.supermicro.com/manuals/other/Memory_RAS_Configuration_User_Guide.pdf
- [28] X. Du and C. Li, "Dpcls: Improving partial cache line sparing with dynamics for memory error prevention," in *2020 IEEE 38th International Conference on Computer Design (ICCD)*, 2020, pp. 197–204.
- [29] C.-S. Hou, Y.-X. Chen, J.-F. Li, C.-Y. Lo, D.-M. Kwai, and Y.-F. Chou, "A built-in self-repair scheme for drams with spare rows, columns, and bits," in *2016 IEEE International Test Conference (ITC)*, 2016, pp. 1–7.
- [30] X. Du, C. Li, S. Zhou, X. Liu, X. Xu, T. Wang, and S. Ge, "Fault-aware prediction-guided page offlining for uncorrectable memory error prevention," in *2021 IEEE 39th International Conference on Computer Design (ICCD)*, 2021, pp. 456–463.
- [31] X. Jian and R. Kumar, "Adaptive reliability chipkill correct (arcc)," in *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, 2013, pp. 270–281.
- [32] X. Du and C. Li, "Combining error statistics with failure prediction in memory page offlining," in *Proceedings of the International Symposium on Memory Systems*, ser. MEMSYS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 127–132. [Online]. Available: <https://doi.org/10.1145/3357526.3357527>
- [33] D. Tang, P. Carruthers, Z. Totari, and M. Shapiro, "Assessment of the effect of memory page retirement on system ras against hardware faults," in *International Conference on Dependable Systems and Networks (DSN'06)*, 2006, pp. 365–370.
- [34] A. Kleen, "mcelog : memory error handling in user space," 2010.
- [35] K. Criss, K. Bains, R. Agarwal, T. Bennett, T. Grunzke, J. K. Kim, H. Chung, and M. Jang, "Improving memory reliability by bounding dram faults: Ddr5 improved reliability features," in *The International Symposium on Memory Systems*, ser. MEMSYS 2020. New York, NY, USA: Association for Computing Machinery, 2021, p. 317–322. [Online]. Available: <https://doi.org/10.1145/3422575.3422803>