IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 12, 2019, accepted March 25, 2019, date of publication April 10, 2019, date of current version April 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909567

RBER-Aware Lifetime Prediction Scheme for 3D-TLC NAND Flash Memory

RUIXIANG MA[1], FEI WU[1,2], (Member, IEEE), MENG ZHANG[1], ZHONGHAI LU[3], JIGUANG WAN[1], AND CHANGSHENG XIE[1,2], (Member, IEEE)

[1]Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China
[2]Shenzhen Research Institute, Huazhong University of Science and Technology, Shenzhen 518000, China
[3]School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden

Corresponding author: Fei Wu (wufei@hust.edu.cn)

This work was supported in part by the Creative Research Group Project of NSFC under Grant 61821003, in part by the NSFC under Grant 61872413 and Grant U1709220, in part by the Wuhan Science and Technology Project under Grant 2017010201010108, in part by the Shenzhen Basic Research Project under Grant JCYJ20170307160135308 and Grant JCYJ20170818162129916, in part by the National Key Research and Development Program of China under Grant 2018YFB10033005, in part by the Fundamental Research Funds for the Central Universities under Grant 2016YXMS019, in part by the 111 Project under Grant B07038, and in part by the Key Laboratory of Data Storage System, Ministry of Education.

ABSTRACT NAND flash memory is widely used in various computing systems. However, flash blocks can sustain only a limited number of program/erase (P/E) cycles, which are referred to as the endurance. On one hand, in order to ensure data integrity, flash manufacturers often define the maximum P/E cycles of the worst block as the endurance of flash blocks. On the other hand, blocks exhibit large endurance variations, which introduce two serious problems. The first problem is that the error correcting code (ECC) is often over-provisioned, as it has to be designed to tolerate the worst case to ensure data integrity, which causes longer decoding latency. The second problem is the underutilized block's lifespan due to conservatively defined block endurance. Raw bit error rate (RBER) of most blocks have not arrived the allowable RBER based on the nominal endurance point, which implies that the conventional P/E cycle-based block retirement policies may waste large flash storage space. In this paper, to exploit the storage capacity of each flash block, we propose an RBER-aware lifetime prediction scheme based on machine learning technologies. We consider the problem that the model can lose prediction effectiveness over time and use incremental learning to update the model for adapting the changes at different lifetime stages. At run time, trained data will be gradually discarded, which can reduce memory overhead. For evaluating our purpose, four well-known machine learning techniques have been compared in terms of predictive accuracy and time overhead under our proposed lifetime prediction scheme. We also compared the predicted values with the tested values obtained in the real NAND flash-based test platform, and the experimental results show that the support vector machine (SVM) models based on our proposed lifetime prediction scheme can achieve as high as 95% accuracy for flash blocks. We also apply our proposed lifetime prediction scheme to predict the actual endurance of flash blocks at four different retention times, and the experimental results show that it can significantly improve the maximum P/E cycle of flash blocks from 37.5% to 86.3% on average. Therefore, the proposed lifetime prediction scheme can provide a guide for block endurance prediction.

INDEX TERMS NAND flash, P/E cycle, retention time, RBER, machine learning.

I. INTRODUCTION

NAND flash has been widely used for data storage due to its high density, high throughput, and low power. To realize the high storage capacity, the size of 2D planar NAND flash has been scaled down from 9Xnm to 1Xnm, which can also reduce the cost per bit. Nowadays, it has reached the limit in process technology feature size, it is becoming extremely difficult to further increase memory capacity using conventional 2D planar NAND flash memory technology. To solve this problem, 3D NAND flash has been studied and developed to improve the storage density, it can be allowed to use larger

The associate editor coordinating the review of this manuscript and approving it for publication was Lorenzo Ciani.

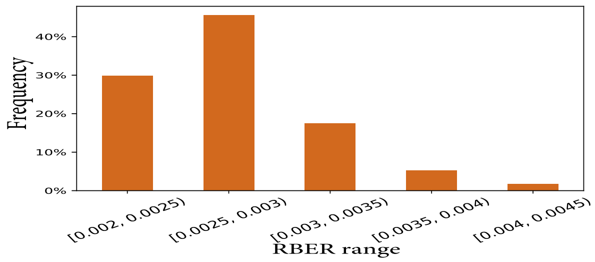


FIGURE 1. **The distribution of RBER when P/E cycle is 6000 for 600 blocks.**

feature sizes and it is becoming an alternative to 2D planar NAND flash. Unfortunately, the reliability of NAND flash is still difficult to guarantee.

Flash cell in 2D planar NAND flash or 3D NAND flash can only undergo a limited number of Program and Erase (P/E) cycles due to the damage caused by the program and erase operation on it, which will eventually lead to its wear-out over time [1]. Endurance is defined as the maximum P/E cycles that block can sustain before its Raw Bit Error Rate (RBER) exceeds the error correction capacity of Error Correcting Code (ECC). Traditionally, NAND flash manufacturers also prescribe a specified endurance for all blocks in the same chip. However, due to process variations in the lithography of flash manufacturing, the RBER varies substantially from block-to-block and page-to-page [2]–[4]. For illustration, we also choose 600 blocks which are evenly distributed in a 3D-TLC NAND flash memory chip tested in our experiment. Figure 1 shows the measured probability density distribution for per-block RBER within the 600 blocks that all have endured 6000 P/E cycles, the bars show the measured per-block RBERs are categorized into 5 bins, the results represent that flash blocks have a great difference in reliability. Thus, the endurance has to be defined as the worst case block with consideration of data integrity across all blocks. Some researches have revealed that endurance of most blocks is higher than the nominal endurance, most blocks can still be used when P/E cycles exceed the specified endurance [3], [5]–[7]. However, in the practical use of NAND flash, the flash-based device controller stops using one block when P/E cycles exceed its nominal endurance, and mark it a bad block. Most blocks are limited by the nominal endurance and don't arrive the allowable RBER limits at the nominal endurance point, NAND flash is not fully utilized, which introduces great waste of storage space.

The lifetime of NAND flash can be improved by the wear leveling algorithms which always regards P/E cycle as the target and make even consumption of P/E cycles among blocks. However, wear leveling algorithms improve service lifetime of NAND flash by minimizing the number of worn-out blocks (Flash will be indicated as bad when the number of worn-out blocks exceeds a given threshold), which actually don't improve the number of P/E cycles that each flash blocks can tolerate. Some reliability enhancement schemes take heuristic solutions to improve the endurance of flash blocks [4], [8]–[15], however, parameters such as ECC strength, program voltage and retention time need to reconfigured at run-time. Our aim is to make full use of the storage capacity of blocks without changing these parameters. In this paper, we will exploit the maximum P/E cycle to extend the lifetime of the individual block and make flash blocks suffer the maximum allowed RBER rather than retiring it at a prescribed P/E threshold at given parameters, and the proposed scheme is also orthogonal with existing works, it means that they are used together to improve the reliability of NAND flash.

Nowadays, machine learning algorithms have been widely used on various applications, it has been proved to be an excellent method for finding the intrinsic relation of data and it needn't any assumption compared with traditional statistical learning method. Furthermore, machine learning models can adapt to the change at run-time. In this paper, block information arrive online and model is established based on their historical data, current data (previously unseen) that need to be predicted have no overlap with training dataset for trained models and the status of flash blocks also change over time, thus, trained models cannot guarantee predictive accuracy at run-time, which is called "model aging". The statistical learning method is also applied at off-line learning and model is typically updated by retraining accumulative data, which can need to preserve and train accumulated data and cause great memory space and time overhead. Therefore, in the paper, we could take machine learning technology to establish flash lifetime prediction models and apply models to predict endurance of flash blocks in advance, which can make full use of flash blocks.

In addition to the P/E cycle, data retention time is another important factor related to data reliability. The lifetime of NAND flash also is defined as the P/E cycles with which data can be reliably stored in flash cells while avoiding data loss for a minimum data retention period as guaranteed by manufacturers. NAND flash devices traditionally have retention times which are expected to hold data for one or more years. According to the JEDEC standard [16], retention time is required for at least 10 years when P/E reaches 10% of the given endurance. When P/E is 100% of a given endurance, retention time is at least 1 year. Enterprise SSD must meet the requirement of placing at 40°C for 3 months and Customer-level SSD requirements can be stored at 40°C for 1 year. In practice, retention time is often inversely proportional to the P/E cycle in NAND flash, the higher the number of P/E cycles, the worse the retention issue is.

To fully exploit a more accurate reflection of a block's endurance, it is very necessary to take into account the effect of data retention. In this paper, by exploiting a key challenge about how to perform lifetime prediction flexibly and efficiently, we propose an RBER-aware lifetime prediction scheme based on machine learning techniques to fully exploit the storage capacity of flash blocks.

In summary, the main contributions of this paper are summarized as follows:

- We investigate the problem that most flash blocks do not arrive their actual endurance based on current endurance definition and take RBER as the reliability measure.
- To establish the optimal machine learning models for flash blocks, four well-known machine learning algorithms are compared in terms of generalization ability and time overhead.
- We find that there exists a model aging problem in this lifetime prediction process and propose a model updating strategy to construct adaptive run-time models.
- We apply SVM models based on our proposed lifetime prediction scheme to predict endurance of flash blocks, the experimental results show it can significantly improve the maximum number of P/E cycle from 37.5% to 86.3% at four different data retention time. We also discuss several different use cases of block lifetime prediction model.

The rest of paper is organized as follows: Section II shows the background of 3D-TLC NAND flash and motivation. Related works are discussed in Section III. The design of the proposed lifetime prediction scheme is presented in Section IV. Experiment and evaluation results are described in Section V, and the conclusions are presented in Section VI.

II. BACKGROUND AND MOTIVATION

A. CHARACTERISTICS OF 3D NAND FLASH MEMORY

3D NAND flash, where multiple layers are vertically stacked to increase the density and improve the scalability of memory, consists of multiple Logical Units (LUNs). A Logical Unit (LUN) consists of multiple planes. A plane consists of multiple blocks. A block consists of multiple pages. Each LUN has at least one page register and cache register, page register is used to transfer data from NAND flash array, cache register is used to transfer data from the host. Each page is divided into the user area and spare area. The user area is used to store written data, the spare area is used to store ECC or other metadata. 3D NAND flash's basic operations include read, program, and erase. A page is the smallest unit of read and program operation, the smallest unit of erase operation is a block, an erase operation resets the data to value "1" in all pages of a block. NAND flash does not support in-place update, the block must be erased before programming data.

In 3D-TLC NAND flash, most existing 3D NAND flash memory designs use a charge trap (CT) transistor for each cell, each cell can be programmed to eight distinct states, which each state correspond to the 3-bit value, the state of a flash cell is determined by the number of electrons present in a cell. As shown in Figure 2, each state corresponds to a non-overlapping threshold voltage window, the non-overlap space between adjacent distributions is called the distribution margin. To read the value stored in a cell, the flash memory applies a read reference voltage to it, seven predefined read voltage levels are used to distinguish the eight states in 3D-TLC NAND flash. The threshold voltage is usually distorted by various sources, such as P/E error, disturb error, retention time [2], [17]. Thus, the cell can be misread when we apply the read reference voltage in it, which leads to raw bit errors.

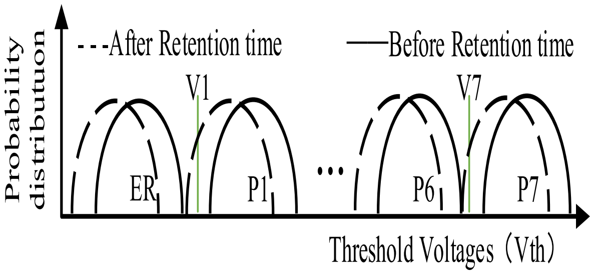


FIGURE 2. Threshold voltage shifts induced by retention time.

B. FLASH ENDURANCE AND DATA RETENTION TIME

There are two major sources of errors in NAND flash: the error caused by the program or erase operation and the error caused by charge leakage during the retention time.

Program and erase operations are accomplished via the Fowler-Nordheim (FN) tunneling mechanism. When NAND flash is programmed and erased, the tunnel oxide layer is conductive with a high voltage between the control gate and the substrate layer. However, high voltage can cause the oxide layer to become weak and electrons are trapped in it. When the number of accumulated charges trapped in the oxide layer, it can cause thresholds voltage shift and ultimately change the program and erase levels of a cell. Thus, when we read data from the cell, the read operation can no longer return valid data.

Retention time is the measure of how long the integrity of data can be guaranteed after being written to flash without suffering from data corruption, and NAND flash also has a limited retention time. Retention errors are caused by charge leakage from a flash cell that contains valid data over time while the cell is idle in data retention process [18], [19], it is the dominant source of flash memory error [1], [20]. Due to charge leakage, the threshold voltage of a flash cell decreases over time, the threshold voltage can shift from higher voltage states to lower voltage states, which is shown in Figure 2. Data read from flash memory can be incorrect when the written data is read after a certain retention time.

C. LIFETIME OF FLASH BLOCKS

1) RELIABILITY METRIC

Reliability is the critical problem in flash-based devices, it related to the lifetime of NAND flash. ECC is also adopted to guarantee the reliability of NAND flash, it can correct data with a high RBER and return data at an error rate called Uncorrectable Bit Error Rate (UBER) that must meet the requirement. The strength of ECC needed is a function of the RBER and the acceptable UBER, the stronger the ECC, the longer the usable life of the flash memory cells. However, stronger ECC also causes higher logic complexity and power consumption, which degrades the overall performance of

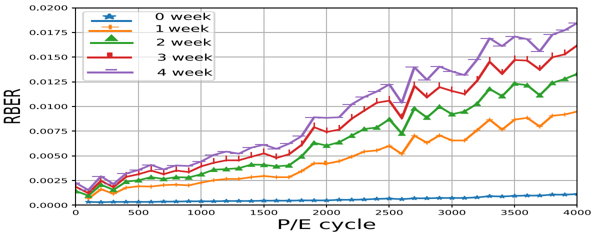


FIGURE 3. **RBER increases with P/E cycle rises at different retention time.**

the system. In the true scenario, the flash manufacturers specify the ECC requirements for their individual flash devices, it is only able to meet output bit error rate requirements with up to a certain RBER, and when the cells deteriorate beyond that point, there is an unacceptable UBER. RBER and UBER are calculated by Equation (1)(2). Therefore, UBER is a function of RBER and determined by it. RBER can be used to be the metric of flash reliability and must be monitored to ensure that it falls in the region of ECC's correction capability.

$$RBER_i = \frac{Bit\ error\ count_i}{Bit\ count\ of\ page} \tag{1}$$

$$UBER_i(n, t) = \sum_{j=t+1}^{n} (RBER_i^j)(1 - RBER_i)^{n-j} \tag{2}$$

where i is the page number, bit count of page is defined as the length of codeword, bit error count is the number of raw bit errors per page, N is the number of the page in a block, $0 < i <= N$, n is the number of bits per page, t is the number of bits that ECC can correct in a page.

2) LIFETIME MODELING OF FLASH BLOCKS

In NAND flash, due to process variations, flash blocks deteriorate at different speeds, which means that their RBER have a huge difference when they arrived the same P/E cycle. Thus, the traditional endurance definition method which takes a conservative P/E cycle as the endurance of flash blocks caused great waste of flash blocks. In order to solve the problem, we must take measure to define 'True' endurance of each individual block. In 3D CT NAND flash, RBER rises as P/E cycle increases and RBER increases with a different rate under different retention time [2]. Figure 3 shows that the higher the retention time, the faster the RBER change, thus, there is the trade-off between the P/E cycle and data retention time.

In practice, the lifetime of NAND flash is also defined as the P/E cycles with which data can be reliably stored in flash cells while avoiding data loss for a minimum data retention period as guaranteed by manufacturers [1], [5], [6], [16], in other word, the lifetime of flash blocks consists of P/E cycle which correspond to the device lifetime and retention time which correspond to data lifetime, it motivates us that we can establish lifetime prediction models of each flash blocks based on the relationship of P/E cycle, retention time and RBER. Hence, with the help of lifetime prediction model, we can predict the actual endurance of each block at the given retention time and the allowable RBER in advance, which can make full use of flash blocks.

III. RELEATED WORK

NAND flash-based devices (e.g., Solid State Disk (SSD)) and Hard Disk Driver (HDD) are two main storage devices for data storage. Unfortunately, their reliability will deteriorate over time, which can cause data loss and have catastrophic effects for individual or enterprise. With the development of machine learning and statistical learning methods, they have attracted more interest to assist the storage system security from the industry and academy, some works have taken machine learning and statistical learning models to improve the reliability of HDD and NAND flash-based device, where researches of flash-based device have mainly focused on NAND flash memory rather than flash-based device.

A. MODEL-BASED TECHNIQUES FOR OPTIMIZING RELIABILITY OF HARD DISK DRIVE

Queiroz *et al.* [21] introduce a failure detection model methodology. In this paper, Recursive Feature Elimination is used to find a subset of SMART attributes that best represents the input data The proposed model is built upon semi-parametric and nonparametric methods, which uses a semi-parametric model (Gaussian Mixture Model) to build a statistical model of the SMART attributes of healthy HDDs and uses a nonparametric procedure to detect faults in HDD.

Li *et al.* [22] employ Classification Tree algorithm to establish failure model, they also proposed a health degree model based on the regression model, which give the drive a health assessment rather a simple classification result. This paper simulates the practical use of the proposed scheme in real-world data centers and develops a Markov model for RAID-6 systems to evaluate how their prediction models benefit the reliability of large-scale systems.

Mahdisoltani *et al.* [23] take a variety of machine learning techniques to predict sector errors instead of disk failure, results show that in that even smaller training data sets are sufficient for successful training and that predictors trained on one drive model can be used to predict errors on a different drive model. This paper also proposes a number of different use cases for error prediction.

Xiao *et al.* [24] introduce a disk failure prediction model using Online Random Forests, which can automatically evolve with the sequential arrival of data. This paper simulates the long-term use of Online Random Forests based prediction models and demonstrates the effectiveness and adaptivity of their method in real-world data centers.

B. MODEL-BASED TECHNIQUES FOR OPTIMIZING RELIABILITY OF NAND FLASH MEMORY

Carlo *et al.* [9] propose to establish a flash RBER prediction model to solve the problem which ECCs are designed for the worse-case reliability design, flash controller can adapt the ECC correction of each page based on the model, this paper considers the combined effect of P/E cycle and retention time.

Bertozzi *et al.* [10] observe that designing NAND flash-based systems based on worst-case scenarios leads to a waste of resources in terms of performance, power consumption, and storage capacity, thus, they exploit runtime reconfigurability to support differentiated access modes in flash memory controller, this paper proposes to combine an adaptable memory programming algorithms and adaptable ECC for providing trade-off between performance, reliability, and power.

Gherman *et al.* [11] also investigate that data refresh scheme is based on worst-case scenarios, which can cause unnecessary data refresh operation. Thus, the authors propose to establish a prediction model to predict the data retention age, data refresh operation will be triggered if the predicted remaining retention time is smaller than the time to the next read operation, otherwise, data will be refreshed, this adaptive scheme can reduce write overhead caused by unnecessary refresh operation and can improve the lifetime of NAND flash.

Zambelli *et al.* [12] propose to optimize ECC based on clustering algorithms, and adapt the code rate of LDPC to reduce the implementation cost based on the results of clustering, which can improve the lifetime of NAND flash.

Nakamura *et al.* [13] found that 25% of P.D. errors are concentrated in 3.5% of the memory cells, and these cells also have poor retention time. The authors take a machine learning approach to detect these cell in advance and screen these cell, which can reduce retention error and program disturb error.

The above discussed works take adaptive flash controller reconfigurability to improve flash reliability. In our paper, we aim to exploit the actual lifetime of NAND flash to expand the service time of NAND flash that has already been deployed and needn't take above reliability enhancement techniques. Our proposed method is also orthogonal with existing technologies or some other heuristic solutions [4], [14], [15].

Zous *et al.* [25] proposed a tolerance assessment method that initial P/E window and erase threshold voltage have a linear relation. This linear relation can be used to assess the performance of the tunnel oxide and optimize the erase waveform. According to the linear relation, it can be predicted when the cell can no longer store data.

Lee *et al.* [26] studied E_{aa} for sub 20*nm* NAND flash memory. They revealed the anomalous origin feature at E_{aa} (apparent activation energy) and derived the mathematical formula which is a function of E_a (the interface trap) in NAND flash, and used the proposed mathematical equation to estimate the lifetime of the NAND flash.

The above two works only consider the lifetime of data retention in NAND flash, our work will consider the lifetime of NAND flash which is consist of data retention time and the P/E cycle.

Fayrushin *et al.* [27] find that endurance degradation is determined by trapping properties of tunnel oxide and distribution of erase current. Therefore, the authors propose to predict endurance of flash by simulation of several P/E cycle steps with subsequent determination of midgap voltage, where each step of simulated P/E cycles corresponds to a specific distribution of trapped charge concentration in tunnel oxide.

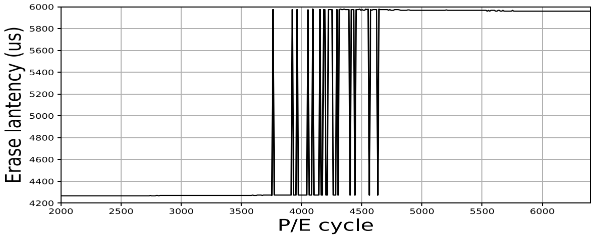


FIGURE 4. **Erase latency of block which is randomly chosen from chip.**

Peleato *et al.* [28] take RBER as a measure of block failure and calculate the average RBER of pages as the RBER in a block, then, they establish a relationship of P/E cycle, program time, RBER and RBER after 3 months, but it lacks the flexibility to retention time.

Fitzgerald *et al.* [29] propose to find flash metrics that could be measured while the device was P-E cycling, and use them to predict the true endurance of individual flash codewords. This paper didn't consider the impact of data retention time.

Hogan *et al.* [30]–[32] establish dataset between program latency, erase latency, and P/E cycles in 2D planar NAND flash. Then, they use genetic programming to perform symbol regression to achieve the prediction model for estimating blocks endurance. In 3D NAND flash, the erase latency exhibits a ladder-shaped growth and fluctuates near the joint of each two steps [2]. Figure 4 shows that the erase latency fluctuates between 3800 P/E cycles and 4700 P/E cycles, which could cause an over-fitting problem for establishing block lifetime prediction model. Furthermore, they also don't take account into the data retention problem. In our paper, we will regard the lifetime of NAND flash as the combination of P/E cycles and data retention time.

Failure prediction for HDD are based on SMART attributes and has achieved great progress in prediction performance. Nowadays, the flash-based device is gradually replacing HDD and become the most important storage device. However, lifetime prediction research of flash-based device is very rare, no large and available dataset for the flash-based device is an important reason, the most individual have no ability to collect flash-based device dataset such as Backblaze dataset [33] that is open source and widely used for HDD failure prediction. The lifetime of a flash-based storage device is also defined as the total amount of data that the device is guaranteed to able to write. Program and erase operation occur whenever existing data needs to overwrite in flash cell, thus, the lifetime of flash-based storage devices is mainly determined by NAND flash. In this paper, we target on NAND flash rather than the flash-based device and establish lifetime prediction models based on block dataset which are collected from real NAND flash chips.

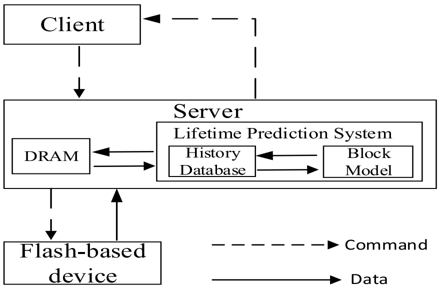


FIGURE 5. The overview of the proposed lifetime prediction architecture.

TABLE 1. The hyper-parameters for different machine learning models in our experiment.

Name	Hy-Parameter
SVM	Kernel type, Penalty parameter
RF	The number of trees, Split criterion, The maximum depth of the tree, The minimum number of samples required to split in a node
MLP	The number of neurons, Activation function, Learning rate, The number of layer, Optimizer
LSTM	The number of layer,the number of units in each layer, the size of sliding window, Activation function, The number of epoch

IV. OUR PROPOSED APPROACH

In this section, we will introduce our proposed lifetime prediction scheme in detail.

A. THE OVERVIEW OF LIFETIME PREDICTION ARCHITECTURE

Figure 5 shows an overview of the proposed block lifetime prediction architecture. In this framework, the client first sends operation commands to the flash-based device which is exposed to host system and can easily communicate with host system, controller of flash-based device records the number of P/E cycle and compute the retention time between two cycles, these data are then sent to the Server in a fixed period, collected data will be sent to Lifetime Prediction System which consists of History Database and Block Model, Lifetime Prediction System is used to pre-train lifetime prediction model and update model over time. Models can be saved as files in the persistent storage device such as SSD or HDD, each block has a corresponding model file. The detailed process can be described as follows: (1) The client sends a request to the Server (2) The server receives the request and sends operation command to the flash-based device for acquiring block dataset (3) The flash-based device receives these commands and implements them on NAND flash, controller of the flash-based device then sent tested data information to the Server and temporarily stored in DRAM (4) Data collected at a certain stage will be sent to History Database (5) Block Model will be trained based on data which is from the History Database and will be dynamically updated based on newly arrived data in a fixed period (6) The client receives the completion command from Server and block lifetime prediction model will be sent to DRAM and ultimately be serialized as a file in the persistent storage device of Server (7) Loading the corresponding model file into Block Model when we need to predict the lifetime of any flash blocks.

B. MACHINE LEARNING METHOD

In order to establish block lifetime prediction model, four well-known machine learning algorithms are represented in Table 1, which are respectively Support Vector Machine (SVM) [34], Random Forest (RF) [35], Multi-layer Perceptron (MLP) [36], Long Short Term Memory (LSTM) [37], they are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. SVM is a kind of machine learning algorithm based on VC dimension theory and risk minimization principle on statistical learning theory. SVM have excellent generalization capability with high predictive accuracy and it has been widely used in classification and regression problem. Random forests, also known as random decision forests, are a popular ensemble method that can be used to build predictive models for both classification and regression problems. The random forest algorithm is based on ensemble learning which is a type of learning where you join different types of algorithms or the same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple decision trees to obtain a more stable and accurate prediction. MLP is an artificial neural network, it consists of an input layer and an output layer that makes a prediction about the input, there is an arbitrary number of hidden layers, MLP is widely used for solving various classification and regression problems. LSTM is a type of recurrent neural networks (RNN) used in deep learning, its aim is to address the issue related to gradient-based learning methods when back-propagating over long sequences. It does so by enhancing previous RNN to include a memory cell and a gating mechanism, which allows for controlling what is remembered in memory and how the new input information contributes to what is already in this memory cell.

C. MACHINE LEARNING BASED BLOCK LIFETIME PREDICTION MECHANISM

As mentioned above, we can apply machine learning methods to exploit the actual endurance of flash blocks. We aim to establish lifetime prediction models based on collected block dataset and then apply trained models to predict the lifetime of flash blocks at a certain condition. In the paper, we take the P/E cycle and retention time as input values and RBER as the output value.

The variation of block-to-block lead to a problem that no individual model can fit all blocks. If we train lifetime prediction models for each flash blocks, which can cause great overhead. However, we respectively train block lifetime prediction models using the same type of machine learning on collected block datasets and find their hyper-parameters are the same or distributed in a specified range, which can be used to scale the range of parameter optimization. For reducing the overhead of searching hyper-parameter optimization, we first choose some blocks that are evenly distributed in the chip and determine the initial hyper-parameter range based these blocks dataset. We then traverse to find the optimal hyper-parameter based on achieved hyper-parameters range for remaining blocks.

Table 1 shows the main hyper-parameters for four different machine learning algorithms. For SVM, we will optimize kernel function, penalty parameter of the error term, kernel function includes the polynomial kernel, radial basis kernel, and sigmoid tanh kernel [34]. For RF, the number of trees, split criterion, the maximum depth of the tree and the minimum number of samples required to split in a node are several important hyper-parameter which will be optimized. For MLP, learning rate, optimizer, activation function will be optimized during fitting, where optimizer include lbfgs [39], sgd [40] and adam [41], etc, activation function include logistic sigmoid function [42], hyperbolic tan function [43] and rectified linear unit function [44]. For LSTM, hyper-parameter are respectively the number of layers, the number of units in each layer, and the size of a sliding window, et al., In the training process, root mean square error is calculated [45], and we set the tolerated error threshold value as 0.001s.

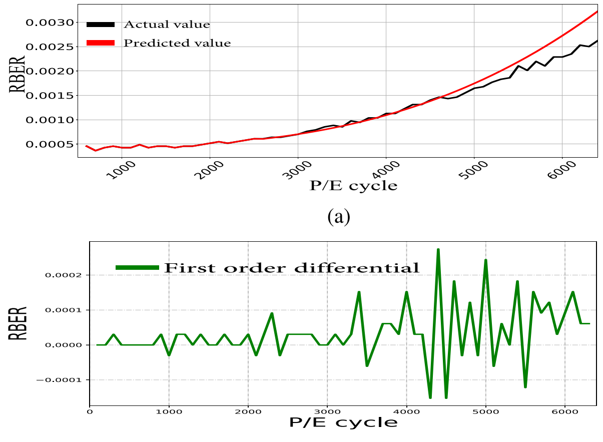
For evaluating models, R-squared measure, R^2, will be used to evaluate the predictive accuracy [38]. In our present paper, R^2 describes how well a lifetime prediction model can fit P/E cycle, retention time and RBER, it is calculated in Equation (3), where y is the actual vector value which consists of multiple samples, $\hat{y}$ is the predicted vector value which consists of multiple samples, y_i is the ith expected response sample, $\hat{y}_i$ is the corresponding predicted value, there are n_{sample} samples. The higher the value is, the better the result is. The best possible score is 1.0, the worst score is 0. Once the block lifetime prediction model is established, it can be used to predict a block's endurance at the required retention time or maximum data retention time in a certain P/E cycle.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{sample-1}} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{sample-1}} (y_i - \bar{y})^2},$$

$$\bar{y} = \frac{1}{n_{samples}} \sum_{i=0}^{n_{sample-1}} y_i \tag{3}$$

D. MODEL AGING PROBLEM IN BLOCK LIFETIME PREDICTION MECHANISM

In the practical scenario which flash-based device has been deployed, the information of blocks are continuously generated over time and future information are unknown at the moment, thus, block lifetime prediction models can only be trained based on historical information and future information are not contained in the training dataset. When the model training stage has finished, trained models will be directly applied to predict the lifetime of flash blocks and models remain unchanged over time, which is called "**Static Lifetime Prediction Scheme**" in the context. The fundamental assumptions of the static lifetime prediction scheme which machine learning methods perform well are that training and testing data follow the same distribution.



(a)

(b)

FIGURE 6. (a) Actual curve and predicted curve of a block based on static lifetime prediction scheme, where retention time is 0. (b) The first order differential of actual curve function.

However, we apply the trained lifetime prediction model to predict the lifetime of flash blocks, there can emerge a problem that these trained lifetime prediction models loss predictive accuracy over time. For illustration, we train lifetime prediction model for a block which is randomly chosen from tested blocks, where training dataset is from 0 to 2500 P/E cycle. As shown in Figure 6 (a), we establish its lifetime prediction model and apply the pre-trained model to predict block status which isn't included in the training set, we find that the predicted curve is above the actual curve when the number of P/E cycle exceed 4600 P/E cycle. Figure 6 (b) shows the first differential of actual curve function, it means that RBER changes with a different rate throughout the whole life of the block, thus, there aren't same distribution between the training set and testing set. Due to space limit, we only present the result of a block, other blocks exhibit the same characteristics with the block. Since lifetime prediction model of the block only relies on the dataset which ranges from 0 to 2500 P/E cycle, it doesn't contain these change situation in the later stage, which can lead to the failure for block lifetime prediction. Thus, it is very important to update model which will contain the situation for flash lifetime prediction.

Figure 7 shows the typical model updating flowchart for flash blocks, the client first sends a model detection request to check if the lifetime prediction model of each flash blocks needs to be updated, the Server receives the request and implement model checking operation. There is a Data Queue

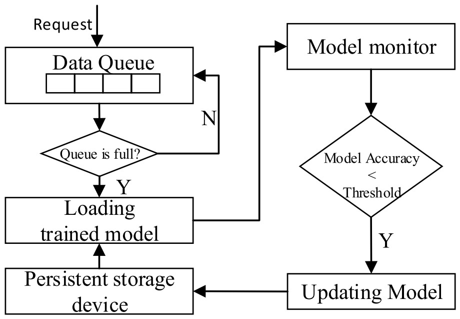


FIGURE 7. The typical model updating flow char of Block lifetime model.

in the host of the Server, the incoming data from the flash-based device will be first sent to Data Queue. When Data Queue is full, the model file of the block will be deserialized from a persistent storage device and loaded into the host system, Model monitor then evaluated the lifetime prediction scheme with newly arrived data from Data Queue and determine whether the model should be updated. The updated model will be serialized and stored in the persistent storage device, the original model will be marked as invalid.

Algorithm 1 Dynamic Lifetime Prediction Algorithm With Model Updating

```
Input: X= (P/E cycle, Retention time)
Output: RBER
/* Searching optimal machine learning algorithms and hyper-parameter range */
Choose some blocks which are evenly distributed in the chip
for K in SVM,...MLP do
    Search the hyper-parameter range for blocks in the chip
end for
/* Pre-training block lifetime prediction model */
Input collected dataset of blocks in the earlier stage
Get pre-trained model of each block based on the achieved hyper-parameter range
/* Updating trained models at run-time */
Acquire new arrived data at run-time, take interval of 500 P/E as a lifetime stage
$X_{stage(i)}$ are inputs and $y_{stage(i)}$ is true output in the i-th lifetime stage.
for i=1,2...n do
    if $R^2(model(X_{stage(i)}), y_{stage(i)}) < 0.9$ then
        Update model with arrived data in this stage
    end if
end for
```

E. OUR PROPOSED LIFETIME SCHEME

In order to solve the model aging problem mentioned in Section IV. D, we should adaptively adjust the lifetime prediction model at run-time. However, there are plenty of blocks in a chip, it is impractical to retrain lifetime prediction models once we find their R^2 are below a given threshold at run-time, which can cause huge overhead. To mitigate the problem, model monitor procedure will be performed infrequently, we define the interval of 500 P/E cycles as a lifetime stage, where each P/E cycle could correspond to different retention time at each lifetime stage, which is different from a fixed length queue mentioned in Figure 7. We then compute R^2 and monitor the results at runtime, the model updating process will be triggered once R^2 is less than 0.9 at any certain lifetime stage. We update the model using newly arrived data in this stage and then apply the updated model to the later prediction. There can also exist a problem that frequently model updating can cause great overhead and impact on system performance. Figure 8 shows probability density for per-block prediction accuracy within 600 blocks, their models are established based on datasets which range from 0 to 2500 P/E cycle and then apply the trained model to predict currently collected data for each block, we find that predictive accuracies of 48% blocks exceed 90%. Therefore, model updating is rare for these blocks. We also have statistics for the number of model updating in 600 blocks, the number of their model updating is less than 4 in their whole lifespan. Therefore, the model updating strategy will not cause frequent updates.

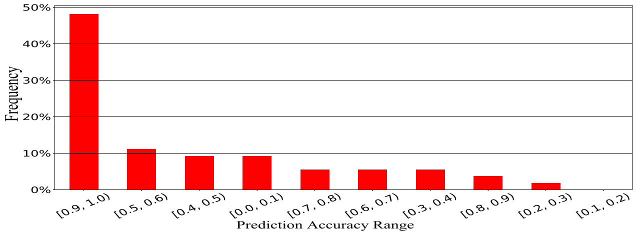


FIGURE 8. Prediction accuracy distribution of 600 blocks based on static lifetime scheme.

V. EXPERIMENT METHODOLOGY

In this section, we describe the methodology used in the experiment for acquiring block dataset, the dataset will be used to train block lifetime prediction model.

A. EXPERIMENT PROCEDURE

In our experiments, a 512G NAND flash is configured based on the specifications of a typical 3D-TLC flash memory structure BICS2 from TOSHIBA. Each chip has 2 LUNs, each LUN has 2 Planes, each Plane has 3944 blocks, each block has 576 pages, the size of one page is 16KB.

Figure 9 shows a NAND flash-based testing platform that allows us to issue commands to raw flash chips, there is not an ECC engine in the testing platform. Due to limitations of experimental condition, we only test a chip of 3D-TLC NAND flash which is described above, we leave a large-scale study of different chips for future work.

Though access patterns are dramatically different, however, there is a data randomization module (scrambler) in a modern flash-based device controller, the data 0 and 1 finally

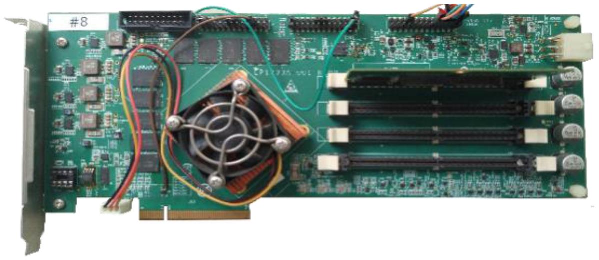


FIGURE 9. NAND flash-based testing platform in the experiment.

written to flash memory are basically balanced. To emulate this, we consider using a pseudo-random number to operate on the flash device. For accelerating experiment process, a data retention test will be carried out in a temperature chamber which can provide precise control over the internal temperature. For ensuring that the temperature can be maintained accurately, our NAND flash testing platform has a temperature sensor, which is used to monitor the ambient temperature. We consider the temperature is accurate if the temperature collected from the sensor is consistent with that set in the temperature chamber.

$$AF = exp^{\frac{-E_{aa}}{K} \times (\frac{1}{T_1} - \frac{1}{T_2})} \tag{4}$$

$$RetentionTime_{T2} = RetentionTime_{T1} \times AF \tag{5}$$

According to the Arrhenius equation [16], which is shown in Equation (4), where E_{aa} is the activation energy, K is the Boltzmann constant: $8.62 \times 10^{-5} V/K$, T_1 is the temperature of high-temperature baking, T_2 is the standard retention temperature which is usually 40°C [16], AF is the acceleration factor. In our experiment, E_{aa} is set to 1.0 eV, T_1 is 85°C, T_2 is 25°C, we convert them into Kelvin. AF is calculated, it is about 105. AF is used to determine the required time placed in the temperature chamber. The baking times need to be normalized to equivalent time at 85°C to simulate the retention time in 40°C. Retention time will be calculated based on Equation (5).

Block's reliability is related to its physic locations [2], thus, in this paper, we select blocks which are evenly distributed in the chip. In the experiment, we first scan the bad block table and remove the bad block from the selected chip. Due to the limitation of the experiment condition, we only select 600 blocks from remaining valid blocks, these blocks are evenly distributed in the different physic locations of the chip. During the experiment, there is a certain recovery time (dwelling time) between the program and erase operation at the room temperature (25°C), which is set to 5s. Then, we repeatedly program and erase target blocks with pseudo-random data to different P/E levels to cover the range from 0 to the P/E cycle which the erase failure or program failure happen. After each 100 P/E cycle interval, a pseudo-random data is written to flash blocks. We then immediately read the data from each block and compare the written pseudo-random data with the value read from the block, we compare it with raw true value, record error counts and calculate RBER. Finally, after completing the required P/E cycle, we perform write operation using pseudo-random numbers. In this paper, we only heat it in 85°C range 0 from 7 hours which correspond from 0 to 4 weeks in 40°C. For every 1.75 hours, we need to cool down to room temperature, we ten read data, compare, and count the number of bit errors at the end of each interval, the detailed flowchart is shown in Figure 10.

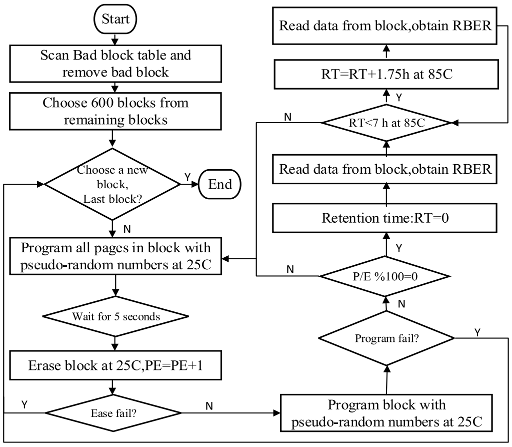

Start
Scan Bad block table and remove bad block
Choose 600 blocks from remaining blocks
Choose a new block, Last block?
Y
End
N
Program all pages in block with pseudo-random numbers at 25C
Wait for 5 seconds
Erase block at 25C,PE=PE+1
Y
Ease fail?
N
Program block with pseudo-random numbers at 25C
Program fail?
Y
N
P/E %100=0
N
Y
Retention time:RT=0
Read data from block,obtain RBER
RT<7 h at 85C
N
Y
RT=RT+1.75h at 85C
Read data from block,obtain RBER

FIGURE 10. Test procedure in the experiment.

TABLE 2. Dataset format of flash blocks in the experiment.

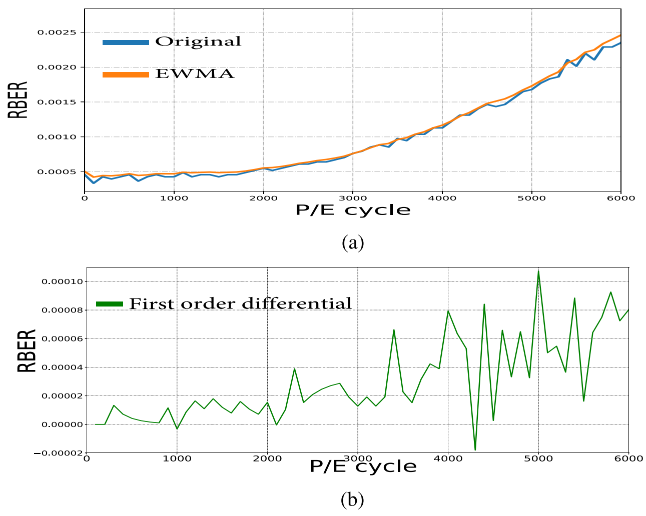
P/E cycle	Retention time	RBER
100 P/E	0 week	$RBER(100, 0)$
100 P/E	1 week	$RBER(100, 1)$
100 P/E	2 week	$RBER(100, 2)$
100 P/E	3 week	$RBER(100, 3)$
100 P/E	4 week	$RBER(100, 4)$
...	...	...
i P/E	0 week	$RBER(i, 0)$
i P/E	1 week	$RBER(i, 1)$
i P/E	2 week	$RBER(i, 2)$
i P/E	3 week	$RBER(i, 3)$
i P/E	4 week	$RBER(i, 4)$

B. BLOCK DATASET

Our dataset collected are based on the above experiment, which includes RBER, P/E cycle and retention time. In this experiment, RBER is collected in 4KB units, therefore, there is totally 2304 4KB's RBER value in a block. We choose the highest RBER from 2304 4K's RBER corresponding to the ith P/E level and retention time of j week in a block and take the value as $RBER(i, j)$ of the block. For the condition's limitation, retention time ranges from 0 to 4 weeks, the detail representation is shown in Table 2.

C. RESULTS ANALYSIS

In this section, we present the evaluation results of our proposed RBER-aware lifetime prediction scheme based on machine learning techniques. We compare four machine learning algorithms in terms of predictive accuracy and time overhead.



(a)

(b)

FIGURE 11. (a) Original data and data after EWMA. (b) First order difference data after EWMA.

1) EXPERIMENT ENVIRONMENT

The experiment is implemented on Windows 64 bits using a desktop equipped with CPU of Intel(R) Core(TM) i5-4460 @3.2GHz, DRAM of 16GB, SSD of 128G, and HDD of 1TB. Tests are implemented in Python 3.5.

2) DATA PREPROCESSING

We analysis dataset and find there is a lot of jitter throughout their life. Because the block data set we have collected is relatively small, these jitters can bring noise pollution for block lifetime prediction model, which can cause an over-fitting problem and seriously affect the predictive stability of models. In order to solve the problem, we take the P/E cycle and RBER as time series data under the same retention time, we then apply a non-uniform weighting to training set, but recent data is weighted more heavily, which is called Exponentially Weighted Moving Average (EWMA) [46], EWMA is calculated in Equation (6). Figure 11 (a) shows the results of applying $N = 5$. Figure 11 (b) shows RBER changes with different rates over time, thus, the model still needs to be dynamically adjusted to adapt change over time.

$$\begin{aligned} EWMA(RBER(i,j)) &= a * RBER(i,j) \\ &\quad +(1-a) * EWMA(RBER((i-1),j)) \end{aligned} \tag{6}$$

$$x_{Norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{7}$$

where x is P/E cycle or retention time, x_{min} is the minimum value and x_{max} is the maximum value, N specify decay in terms of span, $a = \frac{2}{N+1}$, $0 < a <= 1$, i indicates P/E cycle, j indicates retention time.

In the true scenario, we apply the EWMA method to process streaming data, which could cause extra memory overhead. However, Equation (6) shows we need to save the last data that was processed. Experiment results show that the time overhead of data transformation is about 0.0003s for our training set.

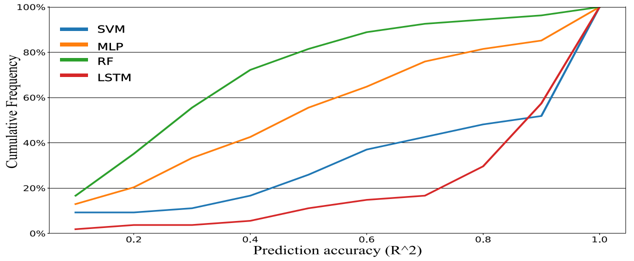


FIGURE 12. Cumulative predictive accuracy distribution of four different machine learning models.

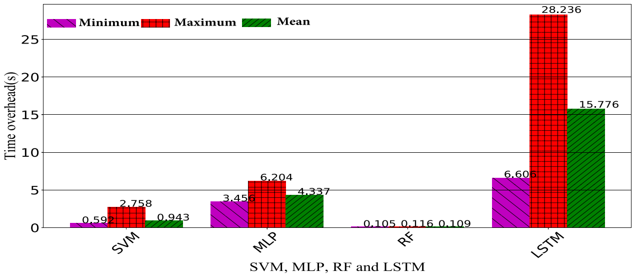


FIGURE 13. Time overhead statistics of block models under four different machine learning algorithms.

Prior to any model training and testing, it is critical to do data normalization so that block data points to be within uniform scale range, which can equal contribution for all features. In the experiment, the P/E cycle and retention time are first normalized using Equation (7).

3) WHICH MACHINE LEARNING ALGORITHM?

We pre-train block lifetime prediction model based on SVM, MLP, RF, LSTM and then evaluate their models in terms of time and predictive accuracy overhead, where training set consists of P/E cycle which ranges from 100 to 2500 P/E cycle and corresponding RBER, testing set consists of P/E cycle which ranges from 2500 to 6000 P/E cycle and corresponding RBER, retention time ranges from 0 to 4 weeks. Experimental results are shown in Figure 12, 13, 14.

According to our pre-training results, their initial parameters of pre-trained models are set as follow : For SVM, Kernel type is polynomial, Penalty parameter range from 1 to 100 ; For MLP, Activation function is tanh, Learning rate is 10^{-3}, the number of neurons is 100, total epoch is 50, Solver is Stochastic Gradient Descent, the number of hidden layers is 1; For RF, the number of trees is 200, split criterion is Gini impurity, the maximum depth of the tree is 6, the minimum number of samples is 60. For LSTM, the size of the sliding window is set 5, the number of neurons for the first hidden layer and second hidden layer respectively are 100 and 50, the activations of the two-layer respectively are both RELU, a fully connected standard neural network, without activation function in its neurons, is later applied to the output of the last LSTM layer, the number of epoch is 10.

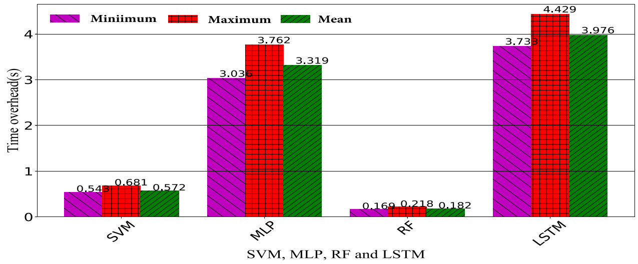


FIGURE 14. Time overhead statistics of one update under different machine learning algorithms.

To choose the optimal machine learning algorithm for flash block, we compare four different machine learning algorithms based on four machine learning models in terms of prediction accuracy and time overhead. We train machine learning models based on the training set and apply the trained models to testing set which has not any overlap with the training set. Figure 12 presents the cumulative probability of prediction accuracy under different machine learning models. The horizontal axis is the predictive accuracy range, vertical axis represent the cumulative probability distribution of 600 blocks under different machine learning algorithms. RF performance worst in terms of predictive accuracy, LSTM have the best predictive accuracy and stability for most blocks, SVM also shows better prediction accuracy and stability. However, as shown in Fig 13, LSTM and MLP can cause great training time overhead compared with SVM and RF, they are unpractical for applying in a real-world system, which can seriously impact the overall system performance. Figure 14 show time overhead of one model updating for the four machine learning models, MLP and LSTM models also cause great time overhead at the model updating process.

In summary, there is a trade-off between predictive accuracy and time overhead for machine learning models, which is crucial for applying it to a practical environment. RF model has the fastest convergence rate, however, its prediction performance is the worst. Therefore, the SVM model is the optimal choice with consideration of predictive accuracy and time overhead, it can be applied in our proposed lifetime prediction scheme.

4) WHICH LIFETIME PREDICTION SCHEME?

In this section, we will compare the static lifetime prediction scheme without model updating and dynamic lifetime prediction scheme with model updating in terms of predictive accuracy and time overhead.

In the dynamic lifetime prediction algorithm, models will be monitored and adjusted over time, which could cause extra time overhead compared to the static lifetime prediction scheme. Figure 14 shows time overhead caused by one model updating under four different machine learning algorithms, which consist of the maximum value, minimum value and mean value for 600 block models. The model updating consists of original model loading and model updating process.

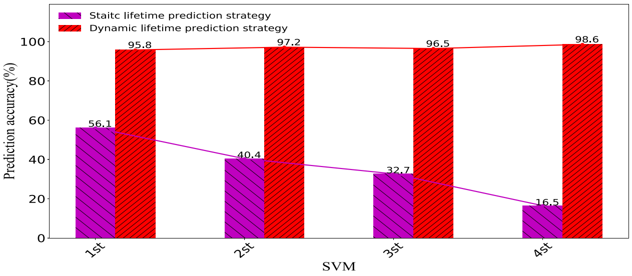


FIGURE 15. Average predictive accuracy of 600 SVM models under static lifetime prediction scheme and dynamic lifetime prediction scheme.

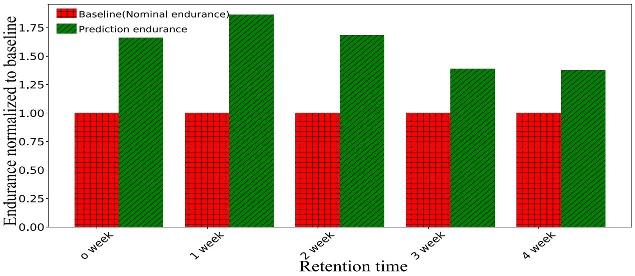


FIGURE 16. Endurance comparison of baseline and predicted endurance based on our proposed lifetime prediction scheme.

Figure 15 shows the predictive results of SVM models based on static lifetime prediction scheme and dynamic lifetime prediction scheme at 4 different lifetime stage. The results show SVM models based on dynamic lifetime prediction scheme has a significant improvement compared with SVM models based on static lifetime prediction scheme in terms of predictive accuracy. Therefore, although the proposed scheme causes a little extra time overhead, it can significantly improve the predictive accuracy of blocks model.

D. ENDURANCE PREDICTION RESULTS

In this paper, the allowable RBER of ECC is assumed as 5×10^{-3} and we define the maximum P/E cycle of the worst block as the nominal endurance within the 600 blocks. We then predict block's endurance based on proposed lifetime prediction scheme, the results represent an average result of 600 blocks at five different retention time. We take the nominal endurance as a baseline, Figure 16 shows the proposed lifetime prediction scheme can significantly improve the endurance of flash blocks from 37.5% to 86.3%.

E. USE CASES FOR BLOCK LIFETIME PREDICTION

In this sector, we will discuss some applications of the lifetime prediction model in a storage system.

1) BLOCK REPLACEMENT

In SSD, flash blocks will gradually wear out over time, with the help of lifetime prediction model, we can predict when flash blocks will fail in advance. We then allocate a new block from over-provisioning space and data in the old block can be migrated to the block before arriving the failure point.

2) WEAR LEVELING

Traditional wear leveling scheme take P/E cycle as the target, which causes great storage waste. Using the lifetime prediction model, we can predict RBER of flash blocks at real-time and take RBER as the target, which can greatly improve the block's lifetime.

3) BLOCK ALLOCATION

With the help of block lifetime prediction model, we can know block's maximum P/E cycles in advance, which can motivate us we can allocate hot data in a strong endurance block and put cold data in a weak endurance block, which can greatly improve the overall lifetime of NAND flash memory.

4) ERROR CORRECTING CODES

Traditional ECC for flash memory is designed to tolerate the worst case to ensure data integrity, it is redundancy for most flash blocks. We can take adaptive ECC for flash blocks based on block's endurance. And the reliability of flash cells changes over time, we also can adaptively adjust ECC at a different stage. We can use a less powerful code at the beginning at the low RBER and stronger ECC when RBER is higher based on the predictive results, which can greatly reduce decoding latency.

5) DATA REFRESH

In order to guarantee data integrity, data refresh operation need to be implemented. A refresh operation consists of reading, correcting, and rewriting the stored data at a fixed frequency that is at least as fast as the current internal retention time, which could cause extra program and erase operation. Data refresh at a fast rate will detect errors more quickly, while a slow data refresh speed imposes less load on the system. Using the block lifetime prediction model, we can predict the maximum retention time in advance and adaptively adjust refresh frequency based on the predictive results, which can reduce the impact caused by data refresh operation on system performance.

VI. CONCLUSION

In this paper, we exploit large variation in flash blocks and traditional block retirement policy introduces a great waste of storage space. To address the problem, this paper introduces a dynamic lifetime prediction scheme which is based on the machine learning technique for 3D-TLC NAND flash memory. We take RBER as the measurement of block reliability and take a machine learning approach to build lifetime prediction models which consist of RBER, P/E cycle, and retention time for each flash blocks. The lifetime prediction models can adapt to the changes at run-time. We also compared four well-known machine learning algorithms in terms of predictive accuracy and time overhead under our proposed lifetime prediction scheme, the results show SVM models perform better in predictive accuracy as well as stability and interpretability. We apply SVM models based on proposed lifetime prediction scheme to predict endurance of flash blocks, the experimental results show it can greatly improve the block's lifetime.

REFERENCES

[1] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, ''Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis,'' in *Proc. DATE*, Mar. 2012, pp. 521–526.

[2] F. Wu *et al.*, ''Characterizing 3D charge trap NAND flash: Observations, analyses and applications,'' in *Proc. ICCD*, Oct. 2018, pp. 381–388.

[3] L. Shi *et al.*, ''Exploiting process variation for write performance improvement on NAND flash memory storage systems,'' *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 1, pp. 334–337, Jan. 2016.

[4] X. Jimenez, D. Novo, and P. Ienne, ''Wear unleveling: Improving NAND flash lifetime by balancing page endurance,'' in *Proc. FAST*, 2014, pp. 47–59.

[5] S. Boboila and P. Desnoyers, ''Write endurance in flash drives: Measurements and analysis,'' in *Proc. FAST*, 2010, pp. 115–128.

[6] L. M. Grupp *et al.*, ''Characterizing flash memory: Anomalies, observations, and applications,'' in *Proc. MICRO*, Dec. 2009, pp. 24–33.

[7] M.-C. Yang, Y.-H. Chang, C.-W. Tsao, and P.-C. Huang, ''New ERA: New efficient reliability-aware wear leveling for endurance enhancement of flash storage devices,'' in *Proc. DAC*, May/Jun. 2013, pp. 1–6.

[8] C. Zambelli *et al.*, ''A cross-layer approach for new reliability-performance trade-offs in MLC NAND flash memories,'' in *Proc. DATE*, Mar. 2012, pp. 881–886.

[9] S. Di Carlo *et al.*, ''FLARES: An aging aware algorithm to autonomously adapt the error correction capability in NAND flash memories,'' *Trans. Archit. Code Optim.*, vol. 11, no. 3, 2014, Art. no. 26.

[10] D. Bertozzi *et al.*, ''Performance and reliability analysis of cross-layer optimizations of NAND flash controllers,'' *Trans. Embedded Comput. Syst.*, vol. 14, no. 1, 2015, Art. no. 7.

[11] V. Gherman, E. Farjallah, J.-M. Armani, M. Seif, and L. Dilillo, ''Improvement of the tolerated raw bit error rate in NAND flash-based SSDs with the help of embedded statistics,'' in *Proc. ITC*, Nov./Oct. 2017, pp. 1–9.

[12] C. Zambelli *et al.*, ''Characterization of TLC 3D-NAND flash endurance through machine learning for LDPC code rate optimization,'' in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2017, pp. 1–4.

[13] Y. Nakamura, T. Iwasaki, and K. Takeuchi, ''Machine learning-based proactive data retention error screening in 1Xnm TLC NAND flash,'' in *Proc. IRPS*, Apr. 2016, pp. 3-1–3-4.

[14] Y. Luo, Y. Cai, S. Ghose, J. Choi, and O. Mutlu, ''WARM: Improving NAND flash memory lifetime with write-hotness aware retention management,'' in *Proc. MSST*, May/Jun. 2015, pp. 1–14.

[15] J. Jeong and S. S. Hahn, ''Lifetime improvement of NAND flash-based storage systems using dynamic program and erase scaling,'' in *Proc. FAST*, 2014, pp. 61–74.

[16] *Stress-Test-Driven Qualification of Integrated Circuits*, Standard JESD47H-01, JEDEC Solid State Technol. Assoc., Vienna, Austria, 2011.

[17] Y. Luo, S. Ghose, Y. Cai, E. F. Haratsch, and O. Mutlu, ''HeatWatch: Improving 3D NAND flash memory device reliability by exploiting self-recovery and temperature awareness,'' in *Proc. HPCA*, Feb. 2018, pp. 504–517.

[18] N. Mielke *et al.*, ''Bit error rate in NAND flash memories,'' in *Proc. IRPS*, Apr./May 2008, pp. 9–19.

[19] Y. Cai, S. Ghose, E. F. Haratsch, Y. Luo, and O. Mutlu, ''Error characterization, mitigation, and recovery in flash-memory-based solid-state drives,'' *Proc. IEEE*, vol. 105, no. 9, pp. 1666–1704, Sep. 2017.

[20] Y. Cai, Y. Luo, E. F. Haratsch, K. Mai, and O. Mutlu, ''Data retention in MLC NAND flash memory: Characterization, optimization, and recovery,'' in *Proc. HPCA*, Feb. 2015, pp. 551–563.

[21] L. P. Queiroz *et al.*, ''A fault detection method for hard disk drives based on mixture of Gaussians and nonparametric statistics,'' *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 542–550, Apr. 2017.

[22] J. Li *et al.*, ''Hard drive failure prediction using classification and regression trees,'' in *Proc. DSN*, Jun. 2016, pp. 383–394.

[23] F. Mahdisoltani, I. Stefanovici, and B. Schroeder, ''Improving storage system reliability with proactive error prediction,'' in *Proc. ATC*, 2017, pp. 391–402.

[24] J. Xiao, Z. Xiong, S. Wu, Y. Yi, H. Jin, and K. Hu, ''Disk failure prediction in data centers via online learning,'' in *Proc. ICPP*, 2018, p. 35.

[25] N.-K. Zous, ''An endurance evaluation method for flash EEPROM,'' *IEEE Trans. Electron Devices*, vol. 51, no. 5, pp. 720–725, May 2004.

[26] K. Lee, M. Kang, Y. Hwang, and H. Shin, "Modeling of apparent activation energy and lifetime estimation in NAND flash memory," *Semicond. Sci. Technol.*, vol. 30, no. 12, 2015, Art. no. 125006.

[27] A. Fayrushin, C. Lee, Y. Park, J. Choi, J. Choi, and C. Chung, "Endurance prediction of scaled NAND flash memory based on spatial mapping of erase tunneling current," in *Proc. Int. Memory Workshop (IMW)*, May 2011, pp. 1–4.

[28] B. Peleato, H. Tabrizi, R. Agarwal, and J. Ferreira, "BER-based wear leveling and bad block management for NAND flash," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 295–300.

[29] B. Fitzgerald, D. Hogan, C. Ryan, and J. Sullivan, "Endurance prediction and error reduction in NAND flash using machine learning," in *Proc. NVMTS*, Aug./Sep. 2017, pp. 1–8.

[30] D. Hogan, T. Arbuckle, and C. Ryan, "Estimating MLC NAND flash endurance: A genetic programming based symbolic regression application," in *Proc. 15th Annu. Conf. Genetic Evol. Comput.*, 2013, pp. 1285–1292.

[31] D. Hogan, T. Arbuckle, and C. Ryan, "Evolving a storage block endurance classifier for Flash memory: A trial implementation," in *Proc. CIS*, Aug. 2012, pp. 12–17.

[32] D. Hogan, T. Arbuckle, and C. Ryan, "How early and with how little data? Using genetic programming to evolve endurance classifiers for MLC NAND flash memory," in *Proc. EuroGP*, 2013, pp. 253–264.

[33] (2018). *The Backblaze Hard Drive Data and Stats*. [Online]. Available: https://www.backblaze.com/b2/hard-drive-test-data.html

[34] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[35] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[36] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, nos. 5–6, pp. 183–197, 1991.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] J. L. DeVore, *Probability and Statistics for Engineering and the Sciences*. Boston, MA, USA: Cengage Learning, 2011, pp. 508–510.

[39] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, pp. 503–528, Aug. 1989.

[40] Paras, "Stochastic gradient descent," *Optimization*, Dec. 2014, pp. 113–132.

[41] Kingma, "Adam: A method for stochastic optimization," in *Proc. ICRL*, Dec. 2015, pp. 1–13.

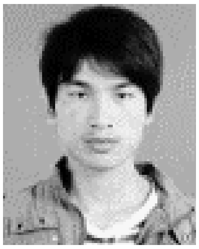
[42] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *Proc. Int. Workshop Artif. Neural Netw.*, 1995, pp. 195–201.

[43] M. Abramowitz, "Hyperbolic functions," in *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York, NY, USA: Dover, 1972, pp. 83–86.

[44] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *Proc. ICML*, 2013, p. 3.

[45] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

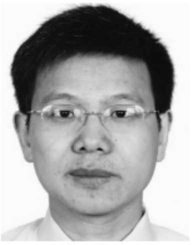
[46] Zangari, "Estimating volatilities and correlations," RiskMetrics-Tech Document, 4th ed. Dec. 1996, pp. 77–101.



RUIXIANG MA is currently pursuing the Ph.D. degree with the Wuhan National Laboratory for Optoelectronics (WNLO), Huazhong University of Science and Technology (HUST), Wuhan, China. His current research interests include intelligent storage, machine learning, and non-volatile memory.

FEI WU received the B.S. and M.S. degrees in electrical automation, control theory, and control engineering from Wuhan Industrial University, Wuhan, China, in 1997 and 2000, respectively, and the Ph.D. degree in computer science from the Huazhong University of Science and Technology (HUST), China, in 2005, where she is currently an Associate Professor with the Information Storage Laboratory, Wuhan National Laboratory for Optoelectronics. Her research interests include computer architecture, non-volatile storage, and green storage. She is a Senior Member of the China Computer Federation and a member of information storage of the China Computer Society.

MENG ZHANG is currently pursuing the Ph.D. degree with the Wuhan National Laboratory for Optoelectronics (WNLO), Huazhong University of Science and Technology (HUST), Wuhan, China. His current research interests include error correction codes (ECC), applications of ECC in non-volatile memory (NVM) technologies, flash memory reliability, and NVM storage systems.



ZHONGHAI LU received the B.S. degree in radio and electronics from Beijing Normal University, Beijing, China, in 1989, and the M.S. degree in system-on-chip design and the Ph.D. degree in electronic and computer system design from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2002 and 2007, respectively, where he has been an Associate Professor, since 2011. He was an Engineer of electronic and embedded systems in Beijing, from 1989 to 2000. He has authored over 130 peer-reviewed papers. His current research interests include interconnection networks, performance analysis, and real-time systems.

JIGUANG WAN received the bachelor's degree from Zhengzhou University, China, in 1996, and the M.S. and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), China, in 2003 and 2007, respectively, all in computer science. His research interests include computer architecture, networked storage systems, file systems, and parallel and distributed systems.

CHANGSHENG XIE received the B.S. and M.S. degrees in computer science and technology from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1982 and 1988, respectively, where he has served as the Deputy Director and is currently a Professor with the Wuhan National Laboratory for Optoelectronics. His research interests include new storage technology and architecture, multimedia computing and networks, computer storage, and network storage and security. He has served as the Deputy Director of the Computer Peripheral Equipment Committee of China Computer Federation, the Committee Member of information storage of the China Computer Society, and the Vice Chairman of the China Expert Committee of the International Network Storage Industry Association.