Machine-Learning-Based Read Reference Voltage Estimation for NAND Flash Memory Systems Without Knowledge of Retention Time

HYEMIN CHOE¹, JEONGJU JEE², SEUNG-CHAN LIM³, (Member, IEEE), SUNG MIN JOE⁴, IL HAN PARK⁴, AND HYUNCHEOL PARK², (Senior Member, IEEE)

¹Samsung Research, Seoul 06765, South Korea
²School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea
³Agency for Defense Development, Daejeon 34186, South Korea
⁴Samsung Electronics, Hwasung 18448, South Korea

Corresponding author: Seung-Chan Lim (sclim@add.re.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2020R1A2C1101158, and in part by the KAIST-Samsung Electronics Device Solutions (DS) Division Strategic Industrial Project sponsored by Samsung Electronics.

ABSTRACT To achieve a low error rate of NAND flash memory, reliable reference voltages should be updated based on the accurate knowledge of program/erase (P/E) cycles and retention time, because those severely distort the threshold voltage distribution of memory cell. Due to the sensitivity to the temperature, however, a flash memory controller is unable to acquire the exact knowledge of retention time, meaning that it is challenging to estimate accurate read reference voltages in practice. In this article, we propose a novel machine-learning-based read reference voltage estimation framework for the NAND flash memory systems without the knowledge of retention time. To establish an unknown input-output relation of the estimation model, we derive input features by sensing and decoding memory cells in the minimum read unit. In order to define the relation between unlabeled input features and a pre-assigned class label, namely label read reference voltages, we propose three mapping functions: 1) $k$-nearest neighbors-based, 2) nearest-centroid-based, and 3) polynomial regression-based read reference voltage estimators. For the proposed estimation schemes, we analyze that the storage overhead and computational complexity are increasing function of the exploited feature dimension. Accordingly, we propose a feature selection (or dimension reduction) algorithm to select the minimum dimension and corresponding features to reduce the overhead and complexity while maintaining high estimation accuracy. Based on extensive numerical analysis, we validate that the derived features successfully replace unknown knowledge of retention time, and the proposed feature selection algorithm precisely adjusts the trade-off between overhead/complexity and estimation accuracy. Furthermore, the simulation and analysis results show that the proposed framework not only outperforms the conventional estimation schemes but also achieves the near-optimal frame error rate while sustaining low latency performance.

INDEX TERMS NAND flash memory, read reference voltage estimation, machine learning, dimension reduction.

I. INTRODUCTION

NAND flash memory is a non-volatile data storage medium with fast read/write speed and large storage capacity [1]. Due to the fascinating features, NAND flash memory has been widely used in many applications such as smart phones, USB flash drives, and solid state drives. Although the storage capacity has been greatly boosted owing to the advance of memory scaling and multi-leveling technologies [2], [3], the reliability of the NAND flash memory has been significantly degraded.

The logical value stored in a memory cell is determined by classifying where threshold voltages belong to. As a cell size decreases and the number of stored bits increases,
the range separating the logical values becomes smaller. Thus, the stored data can be vulnerable to circuit level noise such as random telegraph noise (RTN) [4], [5] and data retention [6], [7]. The increment in program/erase (P/E) cycles and retention time causes the shift or widen of threshold voltage distribution in a cell, resulting in increasing error rates due to the degradation of NAND flash memory channel [8].

Since the optimal read reference voltages are defined as the intersections of the neighboring distributions [9], those are directly affected by channel impairments such as the P/E cycles and retention time. Hence, the optimal read reference voltages can be obtained only if impairment parameters are exactly known at the NAND flash controller. In actual NAND flash operations, the flash memory controller can acquire the number of P/E cycles through wear-leveling [10]. However, the controller is generally unaware of the accurate knowledge of retention time [7], [11]. Therefore, the estimation of read reference voltages should be taken into account without the exact knowledge of retention time in practice.

There have been intensive researches investigated into the read reference voltage estimation problems. In [14], [15], a read reference voltage quantization method has been introduced to obtain reliable soft information, e.g., log-likelihood ratios (LLRs). This method obtains read reference voltages by using entropy value of each unreliable region or by maximizing the mutual information between the input and output of the discrete read channel. In [16], the read reference voltages were estimated by referring the pre-computed table which stores the relation between channel impairment parameters (e.g., the number of P/E cycles, retention time, page number, and page type) and pre-measured read reference voltages. However, the conventional methods in [14]–[16] have estimated read reference voltages under the assumption of perfectly known retention time, meaning that those may be improper for practical applications.

Meanwhile, there have been various parametric-search-based schemes for the read reference voltage estimation without the knowledge of retention time. A parameter estimation algorithm in [17] found best-fit mean and variance of a Gaussian mixture model to minimize Euclidean distance from measured threshold voltage distribution. A decision-directed estimation updated the parameters of threshold voltage distribution based on previously decoded bit error patterns [18]. The Levenberg-Marquardt algorithm with 10-bin equal-probability histograms showed good accuracy for modeling a dynamically varying NAND flash memory channel [19]. In [20], cumulative distribution was estimated by either employing multiple sensing and decoding or interpolation with a bounded function. A retention-aware belief-propagation assisted channel update algorithm adjusted the input LLR of the second round decoding by estimating the mean and variance of the threshold voltage distribution under the assumption that the voltage distribution follows a Gaussian distribution [21]. Although these parametric-search-based schemes have attempted to estimate threshold voltage distribution and corresponding read reference voltages without knowledge of retention time, they rely on information acquired from multiple sensing and decoding, which results in excessive read latency. In addition, these schemes may be impractical because they cannot accurately characterize the threshold voltage distribution of an actual NAND flash memory with a limited number of parameters.

Based on the acquired information from decoding output, dynamic threshold techniques have been proposed to adjust read reference voltages based on metadata implementation and balanced-codes approach [22], [23]. However, the dynamic threshold schemes require to store highly reliable metadata and impose restrictions on the application of arbitrary error-correcting codes. Reserving sentinel cells was another method to estimate read reference voltages by allocating a subset of cells to predict voltage state shift [24]. Nevertheless, this method requires high storage overhead and degrades storage efficiency because a large amount of pre-determined data patterns should be stored in a part of the memory. Read-retry method in [8], [25]–[27] have adjusted a candidate of the read reference voltages by fixed amount, and retry to read the memory using the adjusted voltages. By applying the read-retry method, the controller repeats the read and retry processes to find read reference voltages with higher accuracy if the data recovery fails by using the adjusted read reference voltages in the previous trial. Despite the advantage of simple implementation, the read-retry method may result in significant read latency due to the excessive read trials. An on-chip valley tracking (VT) algorithm estimated amount of voltage shift by measuring the altered number of cells in certain interval [28]. Although the VT method may alleviate the excessive read latency of the read-retry method, the multiple read trials still cause inevitable read latency.

To overcome the aforementioned drawbacks of conventional schemes, we propose a novel machine-learning-based read reference voltage estimation framework for NAND flash memory without the knowledge of retention time. It is difficult to characterize the relation between the channel impairments and the optimal read reference voltages in general. Accordingly, we introduce machine learning techniques to learn unknown input-output relations for the read reference voltage estimation. By establishing the machine learning model in off-line, the proposed framework estimates reliable read reference voltages while sustaining low latency via on-line process. The major contributions of this study are summarized as following.
We propose a machine-learning-based framework for the read reference voltage estimation without the knowledge of retention time. To define the input features of the proposed framework, we derive alternative information of unknown retention time, which are obtained by sensing and decoding the data in one wordline. The proposed framework consists of two phases: 1) off-line training phase and 2) on-line estimation phase. In the off-line training phase, we obtain a training set for learning the estimation model, where training samples are generated from the combination of various P/E cycles and retention time. By assigning the training samples to the corresponding pre-measured label read reference voltage, we can map a new input feature into the read reference voltages in the on-line estimation phase.

For the on-line estimation phase, we propose three estimation schemes: 1) \( k \)-nearest neighbors (\( k \)-NN)-based, 2) nearest-centroid (NC)-based, and 3) polynomial regression (PR)-based estimations. By applying these estimation schemes, an unlabeled input feature is simply mapped into a pre-assigned class label, namely label read reference voltages, via the on-line estimation phase. In this respect, we can interpret the proposed framework as the estimation problem in the supervised learning. It is worth noting that an input feature vector is obtained via sensing and decoding only one wordline, which is the minimum read unit in the operation of NAND flash memory, meaning that the proposed framework does not require excessive read trials to maintain low latency.

We analyze the storage overhead and computational complexity of the proposed estimation schemes in terms of the number of training samples in each class, the number of classes, and the number of exploited features, namely a dimension. Based on the analysis, we note that the overhead and complexity increase with respect to the exploited feature dimension, and there exists the trade-off between the overhead/complexity and estimation accuracy. Therefore, we further propose a feature selection algorithm to minimize the overhead and complexity while maintaining high estimation accuracy. Through the off-line process, the proposed feature selection algorithm determines the minimum dimension and corresponding features exploited in the on-line estimation phase.

Based on the simulation and analysis, we verify the derived features can successfully replace unknown knowledge of retention time, and validate that the proposed feature selection algorithm precisely adjusts the trade-off between the overhead/complexity and estimation accuracy of the established machine learning models. Moreover, the results show that the proposed framework outperforms the conventional estimation methods and approaches to the optimal error rate performance while sustaining low read latency performance.

The rest of the paper is organized as follows. After describing the NAND flash memory basics and system model in Section II, we introduce the machine-learning-based read reference voltage estimation framework in Section III. In Section IV, we propose three read reference voltage estimation schemes, and their storage overhead and computational complexity are analyzed in Section V. Based on the analyzed results, we attempt to minimize the overhead and complexity by developing the feature selection algorithm in Section VI. The performance of the proposed framework is verified by simulation and analysis in Section VII. Finally, some concluding remarks are provided in Section VIII.

II. BACKGROUNDS

A. NAND FLASH MEMORY BASICS

A NAND flash memory chip is composed of thousands of flash blocks, each of which is a two-dimensional array of flash cells. Each block typically contains 64–128 rows, namely wordlines, each of which contains 8KB–16KB cells [29]–[32]. Let \( n \) be the number of bits stored in a cell. The threshold voltage of each cell belongs to one of \( 2^n \) non-overlapping threshold voltage windows, each of which corresponds to one of \( 2^n \) states. The \( n \) bits of each state are divided into \( n \) separate pages. In triple-level cell (TLC) NAND flash memory (i.e., \( n = 3 \)), for example, three bits are allocated to most significant bit (MSB), central significant bit (CSB), and least significant bit (LSB) pages depending on their logical locations.

In the NAND flash memory, the amount of injected charge changes the threshold voltage to store data into a cell. There are three basic operations to store and recover the data as follows [33]–[38].

- Program: By the programming process, charges are injected into a floating gate in a cell. High voltages are applied to cells composing one page to be programmed simultaneously. The program operation is performed in a page unit.
- Erase: To erase written data, injected charges are removed from the cells composing one block. After the erase operation, all cells are configured to the erased state, and the cells can be reprogrammed with new data. The erase operation is performed in a block unit.
- Read: To detect stored data, read reference voltages are applied and the threshold voltage are identified to which of the \( 2^n \) voltage windows. Note that \( (2^n - 1) \) read reference voltages are required to read the stored \( n \)-bit information. The read operation is performed in a wordline unit.

The major impairments of the NAND flash memory system can be modeled as additive noise, which distorts the threshold voltage distribution with increment in the P/E cycle and retention time. The random telegraph noise (RTN) and data retention noise are major sources of additive noise. The RTN causes random fluctuation of threshold voltage and accelerated with an increase of P/E cycles. The data retention
noise is caused by voltage reduction due to the interface trap recovery and electron detrapping processes [39].

**B. SYSTEM DESCRIPTION**

Let us consider the NAND flash memory system depicted in Fig. 1. To overcome the physical noise sources and improve the reliability, the NAND flash memory system is applied with an error-correcting code, e.g., low-density parity-check (LDPC) code. After data encoding, the codewords are programmed into memory cells, and the programmed data is distorted by additive noise, which depends on the P/E cycles and retention time. To read the noisy data, the NAND flash memory controller applies read reference voltages, and noisy data is decoded by the decoder. We assume that an iterative decoding is applied to recover the noisy data, and the decoding process can be terminated early if decoded result is decided to be correct.

**III. MACHINE-LEARNING-BASED READ REFERENCE VOLTAGE ESTIMATION FRAMEWORK**

In this section, we propose the machine-learning-based read reference voltage estimation framework for NAND flash memory systems without the knowledge of retention time. We firstly investigate alternative information on the retention time via data analysis. The alternative information represents individual and independent input variables in machine learning, which are known as features. Based on the derived features, we propose the machine-learning-based read reference voltage estimation framework which consists of two phases: 1) off-line training phase and 2) on-line estimation phase.

**A. ALTERNATIVE INFORMATION ON RETENTION TIME**

To derive the input features of the proposed framework, we present alternative information to replace unknown knowledge of retention time. The features should be clearly distinguishable according to the variation of the retention time, meaning that those can successfully replace the unknown knowledge. Because the pages within the same block tend to have similar retention time, we can assume that the corresponding optimal read reference voltages are identical for the cells composing the same block [7]. In addition, since the wordline is the minimum read unit, we are motivated to observe the memory cells in one wordline to obtain the input features for \( n \) logical pages. Following the NAND flash memory architecture [29]–[32], we assume that a single wordline contains multiple codeword frames each of which are encoded from a smaller size of data frame composing the wordline.

Let \( \mathbf{V}_{\text{def}} \) be the default read reference voltage vector which is defined as

\[
\mathbf{V}_{\text{def}} = [V_{\text{def}}^{(1)}, \ldots, V_{\text{def}}^{(2^n-1)}],
\]

where \( V_{\text{def}}^{(i)} \) represents the \( i \)-th default read reference voltage for \( i \in \{1, \ldots, 2^n-1\} \). After sensing the memory with the default read reference voltages \( \mathbf{V}_{\text{def}} \), we decode multiple data frames composing a single wordline. Based on the successfully decoded results, which are decided by an (outer) error-detecting code such as cyclic redundancy check code, we can derive the following features for different page types \( j \in \{1, \ldots, n\} \), where an index \( 1 \) indicates features of MSB page (\( j = 1 \)), and \( n \) represents features of LSB page (\( j = n \)).

- Frame error rate (FER, \( \phi_j \)): The ratio of decoded frames with errors to the total number of decoded frames
- Average number of iterations (\( \omega_j \)): The average number of iterations of the channel decoder after termination
- Raw bit error rate (RBER, \( \eta_j \)): The ratio of the number of bit errors to the total number of uncoded bits
- Bit flip ratio (BFR, \( \gamma_j \)): The ratio of the number of bit errors from 0 to 1 to the number of bit errors from 1 to 0

Based on the sensing and decoding process of memory cells in one wordline, we collect the observations \( \{\phi_j, \omega_j, \eta_j, \gamma_j\}_{j=1}^n \) and the known number of P/E cycles \( N \) to compose a feature vector. Among the collected \( 4n+1 \) features, some of those are selected to exploit as inputs to the read reference voltage estimation system. It is worth noting that the sensing and decoding one wordline may not be big overhead compared to decoding the entire block (e.g., 128 wordlines) to estimate the read reference voltages without the knowledge of retention time.

Note that the features are derived with a small number of decoded results, so those may be not enough to evaluate the actual statistical values. If we collect results from a large number of wordlines, we may track more accurate values. Nevertheless, we only perform the sensing and decoding process for the minimum read unit since one of our main goals is to develop low-latency machine-learning-based estimation framework for NAND flash memory.

**Experimental Examples:** We consider TLC NAND flash memory when the number of P/E cycle is 1 and the retention time varies from 25 to 50 hours. Fig. 2 shows the derived features with 100 realizations which are generated by Monte Carlo simulation. The features are normalized to have zero mean and unit variance. The simulation results show that \( \phi_2, \omega_3 \) and \( \eta_2 \) tend to increase as retention time increases while \( \gamma_2 \) tends to decrease. Since we can observe distinguishable variation of the derived features according to the change of retention time, these features are suitable to be exploited as the inputs of the machine-learning-based read reference voltage estimation framework when the knowledge of retention time is not available.
ment parameters (reference voltage vector of the lth class label) let those be workable from poor storage efficiency or high read latency to be obtained with sufficient reliability via off-line process. Since we derive (4n + 1) alternative information for n different page types, the dimension can be adjusted between 1 and (4n + 1). We will provide the guideline to optimize the dimension d in Section VI.

Let T and P be sets of retention time and P/E cycles, respectively. In the off-line training phase, M feature vectors are generated for the combination of impairment parameters (t, p) ∈ T × P. Let K be the cardinality of the set T × P (i.e., K = |T||P|), and L be an index set of all class labels (i.e., L = \{1, ..., K\}). The cardinality of the set S equals to MK. In a training set, all vectors are normalized so that each feature contributes uniformly to the estimation process.

For each (t, p), M feature vectors are assigned to the lth class label for l ∈ L, so those vectors are classified in the same class. Let \( \mathbf{V}_{\text{label}} \) be the pre-measured label read reference voltage vector of the lth combination of impairment parameters (t, p). For the lth pair (t, p), we assume that the label read reference voltage vector \( \mathbf{V}_{\text{label}} \) has been obtained with sufficient reliability via off-line process. Since we cannot directly utilize the optimal read reference voltages, we may apply the conventional estimation methods (e.g., [8], [24]–[28]) to obtain sufficiently reliable label read reference voltages \( \mathbf{V}_{\text{label}} \). Although the conventional methods may suffer from poor storage efficiency or high read latency to be applied for the on-line estimation process, those are workable to obtain near-optimal label read reference voltages in the off-line pre-process.

Training Overhead: To generate the M feature vectors via the off-line training phase, M read operations are required since the read operation is performed at the wordline granularity [36]. Assuming that we encode W codewords contained in a single wordline [29]–[32], we also need WM decoding trials. However, these operations are performed via the off-line process, meaning that those do not affect the read latency in the on-line estimation phase.

C. ON-LINE ESTIMATION PHASE

Let \( f \) be a mapping function which is defined as

\[
\mathcal{Y}^d \rightarrow \mathbb{R}^{2^m-1}
\]

\[
y \mapsto \mathbf{V}
\]

where \( \mathcal{Y}^d \) is a set of observable feature vectors and \( \mathbf{V} \) is an estimated read reference voltage vector, i.e., \( \mathbf{V} = [V^{(1)}, \ldots, V^{(2^m-1)}] \). In the on-line estimation phase, the controller maps an input feature vector \( y \) into the read reference voltage vector \( \mathbf{V} \) with the aid of the predefined mapping function \( f \). It is worth noting that the estimation phase of the proposed framework can be interpreted as the supervised learning since an unlabeled input feature vector (i.e., outputs of sensing and decoding in one wordline) is assigned to the class label (i.e., label reference voltage vector).

IV. PROPOSED READ REFERENCE VOLTAGE ESTIMATION

In this section, we propose three supervised-learning-based estimations: k-NN, NC, and PR-based estimations. Contrary to the conventional methods, the proposed framework does not require the knowledge of retention time and excessive read trials since the read reference voltages are estimated by utilizing the alternative information which is generated from the sensing and decoding processes in one wordline.

A. k-NN-BASED READ REFERENCE VOLTAGE ESTIMATION

Fig. 3 depicts the off-line training and on-line estimation phases of the k-NN-based read reference voltage estimation system. Following the description in Sec. III-B, we assume that the training set \( \mathcal{S} \) and the label read reference voltage vectors \( \{\mathbf{V}_{\text{label}}\}_l \) are already obtained via off-line process.

For an input feature vector \( y \) in the on-line estimation phase, the k-NN classifier finds the \( k \) nearest training samples, where all training samples have their own class labels. Then, the k-NN classifier decides the class of \( y \) based on the
majority voting of the class labels of the \( k \) nearest samples. To measure the distance metric, we consider the Euclidean distance between \( y \) and \( x \in \mathcal{S} \). By applying the \( k \)-NN algorithm for the read reference voltage estimation, an index of a class label \( l^* \) is found by

\[
I^* = \arg \max_{l \in \mathcal{L}} \sum_{z \in \mathcal{N}_k(y)} \| \lambda(z), l \|
\]

(3)

where \( \mathcal{L} \) is an index set of all class labels; \( \mathcal{N}_k(y) \) is a set of the \( k \) nearest training samples from \( y \); \( \lambda(z) \) is a function that outputs an index of a class label of the input vector \( z \); and \( \|a, b\| = 1 \) if \( a = b \) and 0 otherwise. Note that the input vector \( y \) is scaled by the normalization factor of the training set.

By using the set \( \{\mathcal{N}_k(y)\}_{l=1}^L \) instead of the training set \( \mathcal{S} \), we can not only reduce the overhead to store the training set but also simplify the following estimation phase.

In the on-line estimation phase, the NC classifier assigns an index which corresponds to the centroid nearest to a new input feature vector \( y \), i.e.,

\[
l^* = \arg \min_{l \in \mathcal{L}} \| y - \bar{x}_l \|_2.
\]

(6)

After the index \( l^* \) is determined by the NC classifier, the estimated read reference voltages are set to the corresponding label read reference voltages, i.e.,

\[
V^{NC} = V^{l^*}_{\text{label}}.
\]

(7)

Fig. 5(a) and Fig. 5(b) depict the operations of the \( k \)-NN-based and NC-based estimations, respectively. In the illustration, we assume that there are two classes \( (K = 2) \), each of which contains 20 feature vectors \( (M = 20) \) obtained via the off-line training phase. The \( k \)-NN-based estimator needs to store 40 training samples \( (MK = 40) \) in the NAND flash controller, and calculates 40 Euclidean distances to find \( k \) nearest samples \( (k = 3) \) for an input feature vector \( y \) in the on-line estimation phase. In contrast, the NC-based estimator requires to store two centroids \( (K = 2) \), and computes two Euclidean distances between the centroids and an input feature vector \( y \) in the on-line estimation phase. For a large number of \( M \), in this respect, the required storage overhead and computational complexity of the NC-based estimation can be much less than those of the \( k \)-NN-based estimation.

C. PR-BASED READ REFERENCE VOLTAGE ESTIMATION

The PR-based read reference voltage estimation system is depicted in Fig. 6. In the PR-based estimation system, we assume that the relation between an input feature vector \( y \) and a read reference voltage vector follows a polynomial function of an order \( m \). Based on the training set \( \mathcal{S} \) and the pre-measured label read reference voltage vectors \( \{V_{\text{label}}\}_{i=1}^K \), we further determine the polynomial function via the off-line training phase. In the on-line estimation phase, then, we simply evaluate the pre-determined function with respect to an input \( y \) to estimate read reference voltage vector.

Let \( V^{PR} = \{V^{PR,(1)}, \ldots, V^{PR,(2^n-1)}\} \) be the estimated result of the PR-based read reference voltage estimation. For \( i \in \{1, \ldots, 2^n - 1\} \), the \( i \)-th PR estimator, \( f^{PR}_i : \mathcal{D} \to \mathbb{R} \), finds the \( i \)-th read reference voltage \( V^{PR,(i)} \) based on the \( m \)-th order polynomial with \( d \) variables which is expressed as

\[
V^{PR,(i)} = f^{PR}_i(y | w^{(i)}) = \sum_{m_1 + \cdots + m_d = m} w^{(i)}_{m_1 \ldots m_d} y_1^{m_1} \cdots y_d^{m_d}.
\]

(8)
where \( w_{m_1 \ldots m_d}^{(i)} \) represents the coefficients of \( y_{m_1}^{(i)} \ldots y_{m_d}^{(i)}. \) If \( m = 1 \) and \( d = 2, \) for example, the polynomial can be expressed as

\[
 f_{PR}^{(i)}(y)w^{(i)} = w_{00}^{(i)} + w_{10}^{(i)}y_1 + w_{01}^{(i)}y_2. \tag{9}
\]

where \( w_{00}, w_{10}, \) and \( w_{01} \) are the coefficients of the polynomial. Likewise, if \( m = 2 \) and \( d = 2, \) the PR estimator becomes

\[
 f_{PR}^{(i)}(y)w^{(i)} = w_{00}^{(i)} + w_{10}^{(i)}y_1 + w_{01}^{(i)}y_2 + w_{02}^{(i)}y_2^2. \tag{10}
\]

where \( w_{00}, w_{10}, w_{01}, w_{20}, w_{11}, \) and \( w_{02} \) are the coefficients of the polynomial.

To optimize the PR-based estimator via the off-line training phase, we need to find the coefficients \( \{w^{(i)}\} \) that minimize a certain cost function. As the cost function, we consider the average mean squared error (MSE) between estimated read reference voltages of the PR estimator and the label read reference voltages. In the training phase of the PR-based estimation, the average MSE is computed by

\[
 MSE^{(i)} = \frac{1}{|S|} \sum_{x \in S} \left( V_{\text{label}}^{(i)}(x) - f_{PR}^{(i)}(x)w^{(i)} \right)^2, \tag{11}
\]

where \( V_{\text{label}}^{(i)}(x) \) is the \( i \)-th label read reference voltage of the feature vector \( x \in S. \) The solution \( w^{(i)} \) which minimizes the average MSE should satisfy

\[
 \frac{d(MSE^{(i)})}{dw^{(i)}} = 0, \tag{12}
\]

and this condition can be expressed in a matrix form as

\[
 X^T X w^{(i)} = X^T V_{\text{label}}^{(i)}, \tag{13}
\]

where the \( j \)-th row of \( X \) is identical to the \( j \)-th feature vector \( x^j \) for \( j \in \{1, \ldots, MK\}, \) and a vector \( V_{\text{label}}^{(i)} \) is defined as

\[
 V_{\text{label}}^{(i)} = [V_{\text{label}}^{(i)}(x_1), \ldots , V_{\text{label}}^{(i)}(x_{MK})]^T. \tag{14}
\]

Thus, the optimal coefficient \( w^{(i)} \) can be calculated by

\[
 w^{(i)} = (X^T X)^{-1} X^T V_{\text{label}}^{(i)}. \tag{15}
\]

In the on-line estimation phase, we directly estimate the read reference voltages \( V_{PR}^{(i)} \) in (8) by plugging the optimized coefficients \( w^{(i)} \) into the \( m \)-th order polynomial \( f_{PR}^{(i)}(y)w^{(i)} \) for \( i \in \{1, \ldots, 2^m - 1\}. \)

\section{V. ANALYSIS ON THE STORAGE OVERHEAD AND COMPUTATIONAL COMPLEXITY}

In order to analyze the performance of the proposed estimation schemes, we consider the storage overhead and computational complexity. The storage overhead is defined as the amount of data which needs to be stored in the NAND flash controller to operate the proposed on-line estimation phase. The computational complexity is measured by the required number of multiplications in the on-line estimation phase since it dominates the complexity of arithmetic operations.

Note that we do not consider the training overhead, which is analyzed in Section III-B, since the training phase is performed via the off-line process. After performing the off-line training phase, the established machine learning model is not updated with increment in data size, meaning that the storage overhead and computational complexity of the on-line estimation phase remain without variations.

\subsection{A. k-NN-BASED READ REFERENCE VOLTAGE ESTIMATION}

Let \( S_{kNN} \) and \( C_{kNN} \) be the storage overhead and computational complexity of \( k \)-NN-based estimation, respectively. For the \( k \)-NN-based estimation, we need to store all \( d \)-dimensional feature vectors in a training set \( S \) with cardinality of \( MK, \) so the storage overhead \( S_{kNN} \) is given by

\[
 S_{kNN} = MKd. \tag{16}
\]

Since the calculation of the Euclidean distance \( ||y - x||_2 \) requires \( d \) multiplications for each \( x \in S, \) the required number of multiplications is readily derived as

\[
 C_{kNN} = MKd. \tag{17}
\]

From the analysis, we observe that the overhead and complexity linearly increase with respect to the size of the training set \( (|S| = MK) \) and the dimension of the input feature vector \( (d). \) Therefore, the \( k \)-NN-based estimation may suffer from large storage overhead and computational complexity if we increase the size of the training set and the number of exploited features to improve the estimation accuracy.

\subsection{B. NC-BASED READ REFERENCE VOLTAGE ESTIMATION}

We denote the storage overhead and computational complexity of the NC-based estimation as \( S_{NC} \) and \( C_{NC}, \) respectively. Since the NC classifier needs to store a set of \( d \)-dimensional centroids \( \{x_i\}_{i=1}^K \) for \( K \) classes, the storage overhead can be represented as

\[
 S_{NC} = Kd. \tag{18}
\]

For a \( d \)-dimensional input feature vector \( y, \) the NC classifier finds the class label of the nearest centroid in (6), so the computational complexity to calculate \( K \) Euclidean distances is derived as

\[
 C_{NC} = Kd. \tag{19}
\]

The analysis shows that both \( S_{NC} \) and \( C_{NC} \) are proportional to the number of centroids \( (K), \) and the dimension of the exploited features \( (d). \) We also note that the storage overhead and computational complexity are \( M \) times less than those of the \( k \)-NN-based estimation since those are independent on the number of feature vectors in a class \( (M). \)

\subsection{C. PR-BASED READ REFERENCE VOLTAGE ESTIMATION}

Let \( S_{PR} \) and \( C_{PR} \) be the storage overhead and computational complexity required for the PR-based read reference voltage estimation. For the \( m \)-th order polynomial with \( d \) variables, the number of coefficients with degree \( k \) equals to \( \binom{d+k-1}{k}. \)}
Therefore, the storage overhead can be expressed as the total number of coefficients to store the polynomials in (8) for all \( i \in \{1, \ldots, 2^n - 1\} \), i.e.,

\[
S_{PR} = (2^n - 1) \sum_{k=0}^{m} \binom{d + k - 1}{k}.
\]

(20)

Also, since the calculation of the monomial with degree \( k \) requires \( k \) multiplications, the computation complexity can be similarly derived as

\[
C_{PR} = (2^n - 1) \sum_{k=1}^{m} \binom{d + k - 1}{k}.
\]

(21)

D. NUMERICAL EVALUATION

In Figs. 7 and 8, we evaluate the analyzed storage overhead and computational complexity for \( M = 100 \) and \( K = 18 \). The evaluation results show that the overhead and complexity of the k-NN-based estimation is the highest since (16) and (17) depend on \( M \). In contrast, the overhead and complexity of the NC-based and PR-based estimations are much less than those of the k-NN-based estimation because those are not dependent on \( M \). If the first-order PR-based estimation is applied, we can achieve the lowest overhead and complexity. On the other hand, we observe that the overhead and complexity of the second-order PR-based estimation are higher than those of the NC-based estimation. From the numerical analysis, we observe that the storage overhead and computational complexity increase with respect to the feature dimension \( (d) \) for all proposed estimation methods. We therefore should carefully determine the number of exploited features in order to reduce both the overhead and complexity while maintaining high estimation accuracy.

VI. DIMENSION REDUCTION VIA THE FEATURE SELECTION

To adjust the trade-off between the overhead/complexity and estimation accuracy in the dimensionality reduction problem, we propose a feature selection algorithm, which is performed in the off-line phase. Since the analytical and numerical results in Section V show that the storage overhead and computational complexity of the on-line estimation methods increase with respect to the feature dimension, the proposed algorithm aims to minimize the number of exploited features while ensuring high estimation accuracy. Based on the wrapper method in [40], the proposed algorithm determines not only how many features to select but also which features to exploit in the on-line estimation phase.

For a given dimension, we evaluate the estimation accuracy of all possible combinations of features, and compare them to choose the combination with the highest accuracy. This approach is computationally expensive since the accuracy should be compared for all possible feature combinations. However, it is performed via the off-line process to convert the feature selection problem into the simple search problem efficiently.

The summary of the proposed feature selection method is provided in Algorithm 1. In the proposed algorithm, we regard \( d \) as a current dimension which is initially set to one without loss of generality. For the current dimension \( d \), we define a set of combination of \( d \) features as \( \mathcal{F}_d \), where the

Algorithm 1 Iterative Feature Selection Algorithm

Require: \( S \) and \( V_{\text{label}}(x) \) for \( x \in S \)

1: Initialize \( d = 0 \) and \( \varepsilon_0 = \infty \)
2: do
3: \( d \leftarrow d + 1 \)
4: for \( f_k \in \mathcal{F}_d \) do
5: \hspace{1em} Calculate \( \text{MSE}(f_k) \) in (22)
6: end for
7: \( f^*_{d} = \arg \min_{f_k \in \mathcal{F}_d} \text{MSE}(f_k) \)
8: \( \varepsilon_{d} = \min_{f_k \in \mathcal{F}_d} \text{MSE}(f_k) \)
9: while \( \varepsilon_{d} \geq \delta \)
10: \( d^* = d - 1 \)
11: return \( d^* \) and \( f^*_{d^*} \)

\( \text{FIGURE 7.} \) The storage overhead required for the proposed machine-learning-based read reference voltage estimation.

\( \text{FIGURE 8.} \) The number of multiplications required for the proposed machine-learning-based read reference voltage estimation.
cardinality equals to $|F_d|$. For a given feature combination $f_d \in F_d$, the average MSE is calculated by

$$\text{MSE}(f_d) = \frac{1}{|S|} \sum_{x \in S} \|V_{\text{label}}(x) - V(x)\|^2,$$  
(22)

where $V_{\text{label}}(x)$ is the label read reference voltage vector of the training sample $x \in S$, and $V(x)$ is a read reference voltage vector estimated by one of the proposed methods, i.e., $V(x) \in \{V^{\text{ANN}}(x), V^{\text{NC}}(x), V^{\text{PR}}(x)\}$.

For the current dimension $d$, we find the best combination of $d$ features, denoted by $f^*_d$, that minimizes the average MSE between the estimated read reference voltages and the label read reference voltages. Therefore, the best combination of $d$ features is selected as

$$f^*_d = \arg \min_{f_d \in F_d} \text{MSE}(f_d),$$  
(23)

and the corresponding minimum MSE is defined as

$$\varepsilon_d = \min_{f_d \in F_d} \text{MSE}(f_d),$$  
(24)

where $\varepsilon_0$ is set to infinity (i.e., $\varepsilon_0 = \infty$) without loss of generality.

The above procedure is iteratively performed with increasing the current dimension by one. The iteration is terminated if the current minimum MSE in (24) is not fairly changed compared to the previous minimum MSE, i.e., $\varepsilon^{|d|-1}_d < \delta$ where $\delta$ is a fidelity parameter. After termination, the proposed algorithm determines the optimal dimension to be the previous dimension $d^* = d - 1$, and we select the corresponding $d^*$ features $f^*_d$. Then, the selected features are exploited in the on-line estimation phase of the proposed machine-learning-based read reference voltage estimation framework.

VII. PERFORMANCE VERIFICATION

Based on the NAND flash memory system model described in Section II, we discuss the performance of the proposed machine-learning-based read reference voltage estimation framework in terms of frame error rate (FER) and the read latency. For modeling NAND flash memory channel, we extract threshold voltage distribution by emulating a 3D flash memory, specifically 64-stack TLC vertical NAND (V-NAND), following the similar process in [41]. In the NAND flash memory model, we assume that 128 wordlines are contained in a flash block and each wordline can store codewords of 16KB data frames. Each of 1KB data frame is encoded by the regular LDPC code with code rate 0.9 and column degree 5. For the LDPC decoding, we apply the min-sum algorithm [42], where the maximum number of iterations is set to 50, and the iterative decoding is terminated early if the decoded result is decided to be successful.

In the off-line training phase of the proposed framework, we consider various channel impairment parameters for $T = \{25, 30, 35, 40, 45, 50\}$ and $P = \{1, 100, 500\}$. For each $(t, p) \in T \times P$, we collect 100 feature vectors ($M = 100$) and corresponding label read reference voltages $V_{\text{label}}$ via the off-line process. After applying the feature selection algorithm, we obtain the sets $S$, $(\tilde{V}^{(k)}k)_{k=1}^{K}$, and $(\tilde{w}^{(l)}l)_{l=1}^{L}$ for the k-NN-based, NC-based, and PR-based read reference estimation systems, respectively. Based on the pre-obtained sets, we can directly apply the proposed read reference voltage estimations for an input feature vector $y$ in the on-line phase.

For the performance comparison with the proposed methods, we consider five benchmark schemes as following.

- **Optimal sensing:** The optimal sensing ideally sets the optimal read reference voltages which are the intersections between adjacent threshold voltage distributions with perfect knowledge of channel impairment parameters [9]. By applying the optimal sensing, we can achieve the optimal FER of the NAND flash memory systems.

- **Default sensing:** The default sensing applies the initially set read reference voltages regardless of the channel variation. We set the default read reference voltages as the average of the optimal read reference voltages when the $P/E$ cycle varies from 1 to 1000 and the retention time varies from 20 to 50 hours.

- **Sentinel-cells-enabled (SCE) method [24]:** Among cells within a wordline, the SCE method reserves a part of cells, namely sentinel cells, to program known data. Using the default read reference voltages, we read the data which is programmed on the sentinel cells. By comparing the known data with the readout data, the SCE method estimates the errors over an wordline, and adjust read reference voltages. To implement the SCE method, we reserve 1.39% and 30.56% of whole cells to the sentinel cells, which are uniformly allocated for each state in an wordline.

- **Read-retry (RR) method [25]:** The read-retry method finds the local minimum of threshold voltage distribution by iteratively shifting current read reference voltage to a direction, where the number of cells becomes smaller. In each iteration, the read-retry algorithm reads $N_{WL}$ wordlines and updates current read reference voltage $V$ to either $V - \Delta V$ or $V + \Delta V$, where $\Delta V$ represents a voltage step size. After comparing the number of cells whose threshold voltages belong to the interval [$V - \Delta V$, $V$] or [$V$, $V + \Delta V$], new read reference voltage is updated to $V + \Delta V$ if the number of cells in [$V$, $V + \Delta V$] is smaller, and updated to $V - \Delta V$ otherwise. We set the voltage step size to be 0.01 ($\Delta V = 0.01$), and

\[2\] To collect $M$ feature vectors in a real NAND flash memory for a given impairment parameter $(t, p) \in T \times P$, we can perform the following procedures: 1) For $M$ wordlines in a block, we repeat $p$ process of writing random data and erasing them. 2) To equivalently implement the retention time $t$, we increase the temperature and perform cycling after Arrhenius extrapolation [39], or apply the unified detrapping metric method [43].
TABLE 1. Selected combination of \( d \) features for the \( k \)-NN-based read reference voltage estimation.

| \( d \) | Selected combination \( f_j^* \) in (23) | MSE \( \epsilon_d \) in (24) |
|-------|---------------------------------|------------------|
| 1     | \( \{\eta_3\} \)                | \( 1.5 \times 10^{-8} \) |
| 2     | \( \{\eta_3, N\} \)             | \( 9.1 \times 10^{-5} \) |
| 3     | \( \{\eta_3, \gamma_3, N\} \)  | \( 3.7 \times 10^{-5} \) |

TABLE 2. Selected combination of \( d \) features for the NC-based read reference voltage estimation.

| \( d \) | Selected combination \( f_j^* \) in (23) | MSE \( \epsilon_d \) in (24) |
|-------|---------------------------------|------------------|
| 1     | \( \{\omega_1\} \)                | \( 1.6 \times 10^{-5} \) |
| 2     | \( \{\eta_3, N\} \)             | \( 1.1 \times 10^{-4} \) |
| 3     | \( \{\eta_3, \omega_3, N\} \)  | \( 9.9 \times 10^{-5} \) |

TABLE 3. Selected combination of \( d \) features for the first-order PR-based read reference voltage estimation \((m = 1)\).

| \( d \) | Selected combination \( f_j^* \) in (23) | MSE \( \epsilon_d \) in (24) |
|-------|---------------------------------|------------------|
| 1     | \( \{\eta_3\} \)                | \( 1.5 \times 10^{-8} \) |
| 2     | \( \{\omega_1, \eta_3\} \)       | \( 4.6 \times 10^{-4} \) |
| 3     | \( \{\omega_1, \omega_3, \eta_3\} \)  | \( 3.6 \times 10^{-4} \) |

TABLE 4. Selected combination of \( d \) features for the second-order PR-based read reference voltage estimation \((m = 2)\).

| \( d \) | Selected combination \( f_j^* \) in (23) | MSE \( \epsilon_d \) in (24) |
|-------|---------------------------------|------------------|
| 1     | \( \{\eta_3\} \)                | \( 1.5 \times 10^{-8} \) |
| 2     | \( \{\eta_3, N\} \)             | \( 9.5 \times 10^{-5} \) |
| 3     | \( \{\eta_3, \gamma_3, N\} \)  | \( 6.2 \times 10^{-5} \) |

A. RESULTS OF THE PROPOSED FEATURE SELECTION ALGORITHM

Tables 1–4 show the results of the proposed feature selection algorithm for the \( k \)-NN-based, NC-based, and PR-based read reference voltage estimation schemes. For each proposed read reference voltage estimation scheme, the best combination of \( d \) features is listed in the corresponding Table. Following the definitions in Sec. III-A, \( \omega_j \), \( \eta_j \), and \( \gamma_j \) denote the estimated average number of iterations, RBER, and BFR of page type \( j \in \{1, 3\} \), respectively, where the subscripts 1 and 3 indicate that features are obtained from MSB and LSB pages, respectively. Also, \( N \) represents for the known number of wordlines to be either 1 (\( N_{WL} = 1 \)) or 128 (\( N_{WL} = 128 \)).

- Valley tracking (VT) method [28]: For a step size \( \Delta V \) and current read reference voltage \( V \), we apply three read trials to obtain the number of cells which belong to \( [V - \Delta V, V] \), denoted by \( N_A \), and the number of cells which belong to \( [V, V + \Delta V] \), denoted by \( N_B \). Assuming that voltage shift is linearly proportional to \( N_A - N_B \), the VT method updates new read reference voltage to \( V + \alpha (N_A - N_B) \), where a parameter \( \alpha \) is found by the least-square method. We set \( \Delta V = 0.1 \) empirically, which results in the lowest FER performance of the VT method.

B. FER PERFORMANCE

In Fig. 9, we compare the FER performances of the proposed estimation schemes with respect to a feature dimension \( d \in \{1, 2, 3\} \). As we can expect from the results in Section VII-A, the proposed estimation schemes cannot achieve the FER of the optimal sensing when the dimension is set to one \( (d = 1) \). By exploiting more than two features \( (d \in \{2, 3\} \) listed in Tables 1, 2, and 4, however, we observe that the \( k \)-NN-based, NC-based, and the second-order PR-based estimation schemes can approach to the optimal FER performance, respectively. From these results, we note that those schemes need to exploit at least two features to achieve the near-optimal FER performance. From Fig. 9, moreover, we can interpret the trade-off between a dimension \( d \) and FER performance, which is discussed in Section VI. In Fig. 9, the results reveal that the FER performance can be degraded as a dimension \( d \) decreases although the storage overhead and computational complexity reduces. From this perspective, we note that the proposed feature selection algorithm precisely adjusts the trade-off to achieve near-optimal FER performance while minimizing an exploited feature dimension.

In Fig. 9(c), it is also noticeable that the FER of the first-order PR-based estimation does not tightly coincide with that of the optimal sensing. Comparing the results in...
Fig. 10 shows the FER performance of each proposed estimation scheme with respect to the combination of selected features. Since the proposed estimation schemes cannot achieve the optimal FER by using only one feature, we consider the combinations of two and three features \( d \in \{2, 3\} \) to verify the results of the proposed feature selection algorithm. To validate the FER of the selected features in Tables 1–4 (which indicate the best combination that minimizes the MSE \( \varepsilon d \) in (24)), we consider the worst combination that maximizes the MSE. The simulation results in Fig. 10 show that the FER of the worst combination is significantly degraded with respect to that of the best combination. From this observation, we confirm that the MSE is a proper metric to optimize \( d \) selected features with high accuracy, and the proposed feature selection algorithm can find valid combination of \( d^* \) features to achieve near-optimal FER performance.

In Fig. 11, we compare the FER of the proposed machine-learning-based read reference voltage estimation framework with those of the benchmark schemes. Since the results in Figs. 9 and 10 reveal that the \( k \)-NN-based, NC-based, and second-order PR-based schemes commonly attain the near-optimal performance by setting \( d = 2 \), we only present the FER of the second-order PR-based method (PR–2). The FER of the read-retry method with \( N_{WL} = 1 \) (RR–1WL) is significantly inferior compared to other schemes since the read-retry algorithm fails to find the local minimum with an insufficient number of cells to express an actual threshold voltage distribution. Though the read-retry method with \( N_{WL} = 128 \) (RR–1block) shows superior FER performance, the FER is slightly degraded with respect to either the proposed methods or optimal sensing. When 1.39% of cells are allocated to the sentinel cells (SCE–1.39%), the FER of the SCE method is also degraded with respect to that of the proposed methods. In contrast, the SCE method can approach to the FER of the proposed methods when we reserve 30.56% of cells (SCE–30.56%), meaning that the SCE method needs to sacrifice storage efficiency significantly to guarantee sufficient reliability. Moreover, we observe that the VT method shows comparable FER performance with the proposed methods by employing two additional read trials. Based on the results, we can verify the high estimation accuracy of the proposed machine-learning-based framework.

In Fig. 12, we verify the feasibility of the proposed framework against a degraded memory channel due to the process variation across different chips. To perform the feasibility verification, we emulate a threshold voltage distribution of a
TABLE 5. Analytical result of read latency performances.

| Methods                        | Read latency          |
|--------------------------------|-----------------------|
| Proposed machine-learning-based method | $T_S + T_D$           |
| Sentinel-cells-enabled (SCE) method [24] | $T_S$                |
| Read-retry (RR) method [25]    | $N_{WL} \times N_I \times T_S$ |
| Valley tracking (VT) [28]      | $3T_S$                |

degraded NAND flash memory chip, which belongs to three standard deviations below from a mean among 700 different chips. Following the off-line training phase, we apply the identical default read reference voltages to extract features and label read reference voltages for various channel impairment parameters. Based on the obtained inputs, we then perform the on-line estimation of the proposed framework to recover stored data. Fig. 12 shows that the FERs of the proposed estimation methods approach to that of the optimal sensing. By collecting the features and class labels from the chip via the off-line training phase and performing on-line estimation phase, we note that proposed framework is feasible to estimate sufficiently reliable read reference voltages even for a degraded NAND flash memory chip due to the process variation.

C. READ LATENCY PERFORMANCE

In Table 5, we summarize the analytical results of the latency performances for read reference voltage estimation methods. For the proposed framework, we do not take into account the training overhead because the training phase is performed via off-line. We instead focus on the required time of the on-line estimation phase to analyze the read latency performance of the proposed methods.

Let $T_S$ and $T_D$ be the required time for sensing and decoding memory cells in one wordline, respectively. The estimation phase of the proposed framework needs to obtain the input features by sensing and decoding one wordline. Hence, the proposed read reference voltage estimation requires read latency of $T_S + T_D$.

Since the SCE method applies only one read trial to estimate read reference voltages, the required read latency is fixed to $T_S$. For the read-retry-based frameworks with $N_{WL} = 1$ and $N_{WL} = 128$, we define the average number of read reference voltage shifting as $N_{I, WL}$ and $N_{I, BL}$, respectively. Since the read-retry algorithm iteratively shifts current read reference voltages for sensing the memory cells in $N_{WL}$ wordlines, the required read latency is derived as $N_{WL} \times N_{I, u} \times T_S$ for $u \in \{WL, BL\}$. After sensing with current read reference voltages, the VT method applies two additional trials by adjusting the current read reference voltages with a pre-defined step size. Accordingly, the read latency of the VT method is given as $3T_S$ since three sensing trials are applied for the read reference voltage estimation.

Based on the analysis summarized in Table 5, we evaluate the required decoding time of 16 codewords within one wordline. In Fig. 13, we compare the read latency of the proposed methods with those of the benchmark schemes. The results show that the read latency of read-retry methods are higher than those of the proposed methods. In particular, the read-retry with $N_{WL} = 1$ (RR–1WL) shows approximately 2.98 times higher read latency. Although the results in Fig. 11 shows that the read-retry with $N_{WL} = 128$ (RR–1block) may have comparable FER performance, the read latency is significantly higher due to excessive read trials. Compared to the VT method, it is noticeable that the proposed methods can achieve $1.5 \sim 2$ times lower read latency though their FER performances are comparable in Fig. 11. Since the read latency of the proposed methods are $100 \sim 150\mu s$ lower than that of the VT methods, the proposed methods can reduce the read latency which corresponds to $1 \sim 1.5$ sensing trials. Although the SCE method shows the lowest read latency among the estimation schemes, we note that this superior latency performance is obtained by sacrificing the storage efficiency significantly. As a result, we conclude that the proposed machine-learning-based framework can achieve both high reliability and low read latency for the NAND flash memory systems.

VIII. CONCLUSION

In this article, we have presented a novel machine-learning-based read reference voltage estimation framework for NAND flash memory systems when the knowledge on retention time is unavailable. To replace unknown knowledge on retention time, we have derived the alternative information by sensing and decoding data in the minimum read unit. Utilizing the derived information as input features of the proposed framework, we have proposed the k-NN-based, NC-based, and PR-based read reference voltage estimation methods. For the proposed estimation frameworks, we have analytically shown that the storage overhead and computational complexity are increasing function of the exploited feature dimension. Accordingly, we have further proposed the feature selection algorithm to find the minimum dimension.
and corresponding features, which precisely adjusts the trade-off between overhead/complexity and estimation accuracy. Based on the simulation and analysis, we have verified that the proposed framework can achieve high-reliable and low-latency performances in NAND flash memory systems without the knowledge of retention time.

An important future work is to apply the proposed machine learning-based read reference voltage estimation framework to a real NAND flash memory. For the emulated 64-stack TLC V-NAND model, in this article, we have validated the proposed kNN-based, NC-based, and second-order PR-based estimation schemes can achieve sufficient reliability with low read latency. By exploiting the revealed potential in this article, we expect that the application of the state-of-the-art machine learning techniques into the proposed framework can overcome complicated physical characteristics of the advanced NAND flash memory.

REFERENCES

[1] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, “Introduction to flash memory,” Proc. IEEE, vol. 91, no. 4, pp. 489–502, Apr. 2003.

[2] G. G. Marotta, A. Macerola, and A. D’Alessandro, “A 3bit/cell 32Gb NAND flash memory at 34nm with 6MB/s program throughput and with dynamic 2bcell blocks configuration mode for a program throughput increase up to 13MB/s,” in IEEE ISSCC Dig. Tech. Papers, Feb. 2010, pp. 444–445.

[3] Y. Li, S. Lee, Y. Fong, F. Pan, and T. C. Kuo, “A 16 gb 3-bit per cell (X3) NAND flash memory on 56 nm technology with 8 MB/s write rate,” IEEE J. Solid-State Circuits, vol. 44, no. 1, pp. 195–207, Jan. 2009.

[4] C. M. Compagnoni, M. Ghidotti, A. L. Lacaita, A. S. Spinelli, and A. Visconti, “Random telegraph noise effect on the programmed threshold-voltage distribution of flash memories,” IEEE Electron Device Lett., vol. 30, no. 9, pp. 984–986, Sep. 2009.

[5] C. M. Compagnoni, R. Gusmeroli, A. S. Spinelli, A. L. Lacaita, M. Bonanomi, and A. Visconti, “Statistical model for random telegraph noise in flash memories,” IEEE Trans. Electron Devices, vol. 55, no. 1, pp. 388–395, Jan. 2008.

[6] J.-D. Lee, J.-H. Choi, D. Park, and K. Kim, “Effects of interface trap generation and annihilation on the data retention characteristics of flash memory cells,” IEEE Trans. Device Mater. Rel., vol. 4, no. 1, pp. 110–117, Mar. 2004.

[7] Y. Cai, Y. Luo, S. Ghose, Y. Cai, E. F. Haratsch, and O. Mutlu, “Data retention in MLC NAND flash memory: Characterization, optimization, and recovery,” in Proc. IEEE 21st Int. Symp. High Perform. Comput. Archit. (HPCA), Feb. 2015, pp. 551–563.

[8] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, “Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling,” in Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE), Mar. 2013, pp. 1285–1290.

[9] B. Peleato, R. Agarwal, J. Cioffi, M. Qin, and P. H. Siegel, “Towards overhead/complexity and estimation accuracy. Based on the simulation and analysis, we have verified that the proposed framework can achieve high-reliable and low-latency performances in NAND flash memory systems without the knowledge of retention time. An important future work is to apply the proposed machine learning-based read reference voltage estimation framework to a real NAND flash memory. For the emulated 64-stack TLC V-NAND model, in this article, we have validated the proposed kNN-based, NC-based, and second-order PR-based estimation schemes can achieve sufficient reliability with low read latency. By exploiting the revealed potential in this article, we expect that the application of the state-of-the-art machine learning techniques into the proposed framework can overcome complicated physical characteristics of the advanced NAND flash memory.
[37] K. Takeuchi, T. Tanaka, and H. Nakamura, “A double-level-Vth select gate array architecture for multilevel NAND flash memories,” *IEEE J. Solid-State Circuits*, vol. 31, no. 4, pp. 602–609, Apr. 1996.

[38] G. Dong, Y. Pan, N. Xie, C. Varanasi, and T. Zhang, “Estimating information-theoretical NAND flash memory storage capacity and its implication to memory system design space exploration,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 9, pp. 1705–1714, Sep. 2012.

[39] N. Mielke, H. Belgal, and I. Kalastirsky, “Flash EEPROM threshold instabilities due to charge trapping during program/erase cycling,” in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1995, pp. 192–197.

[40] R. Kohavi and D. Sommerfield, “Feature subset selection using the wrapper method: Overfitting and dynamic search space topology,” in *Proc. Int. Conf. Artificial Intelligence*, San Francisco, CA, 1995.

[41] C. Hsieh, H. Hsieh, Y. C. Li, T. Chen, H. Li, and C. Lu, “A novel dichotomic programming algorithm applied to 3D NAND flash,” in *Proc. Symp. VLSI Technol. (VLSI Technol.)*, Jun. 2015, pp. T180–T181.

[42] M. P. Fossorier, M. Mihaljevic, and H. Imai, “Reduced complexity iterative decoding of low-density parity check codes based on belief propagation,” *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.

[43] N. Mielke, H. P. Belgal, A. Fazio, Q. Meng, and N. Righos, “Recovery effects in the distributed cycling of flash memories,” in *Proc. IEEE Int. Rel. Phys. Symp.*, Mar. 2006, pp. 29–35.

[44] G. Wu and X. He, “Reducing SSD read latency via NAND flash program and erase suspension,” in *Proc. FAST*, vol. 2, 2012, p. 3.

[45] R.-S. Liu, M.-Y. Chuang, C.-L. Yang, C.-H. Li, K.-C. Ho, and H.-P. Li, “Improving read performance of NAND flash SSDs by exploiting error locality,” *IEEE Trans. Comput.*, vol. 65, no. 4, pp. 1090–1102, Apr. 2016.

**HYEMIN CHOE** received the B.S. degree in electrical engineering from the University of Seoul, Seoul, South Korea, in 2018, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2020. She has been with Samsung Electronics Company Ltd., Seoul, since 2020. Her research interests include the read voltage estimation for NAND flash memory and wireless communications.

**JEONGJU JEE** received the B.S. and M.S. degrees from the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering.

His research interests include massive MIMO systems, hardware-constrained communications, and NAND flash memory systems.

**SEUNG-CHAN LIM** (Member, IEEE) received the B.S. degree in electronic and electrical engineering from Hongik University, Seoul, South Korea, in 2011, the M.S. degree in electronic and electrical engineering from the Pohang University of Science and Technology, Pohang, South Korea, in 2013, and the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2019.

From September 2019 to December 2019, he was a Postdoctoral Researcher with the Information and Electronics Research Institute, KAIST. He has been a Senior Researcher with Agency for Defence Development (ADD), Daejeon, since December 2019. His current research interests include wireless communications, signal processing, and machine learning.

**SUNG MIN JOE** received the M.S. degree from Kyungpook National University, in 2010, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Seoul National University, in 2015. He is currently working at the Flash Design Team, Samsung Electronics Company Ltd. His current research interest includes the next generation VNAND flash memory for high density, high performance, and low power.

**IL HAN PARK** received the B.S. and Ph.D. degrees in electrical engineering from Seoul National University, in 2003 and 2009, respectively. He is currently working at the Flash Design Team, Samsung Electronics Company Ltd. His current research interest includes wireless communications, signal processing, and machine learning.

**HYUNCHEOIL PARK** (Senior Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Yonsei University, Seoul, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 1997.

He was a Senior Engineer, from 1985 to 1991, and a Principal Engineer with Samsung Electronics Company Ltd., South Korea, from 1997 to 2002. Since 2002, he has been with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, where he is currently a Professor. His research interests include 5G/B5G mobile communications, communication theory, and machine learning.