Self-Learning Hot Data Prediction: Where Echo State Network Meets NAND Flash Memories

Qiwu Luo, *Member, IEEE*, Xiaoxin Fang, Yichuang Sun, *Senior Member, IEEE*, Jiaqiu Ai, *Member, IEEE*, and Chunhua Yang, *Member, IEEE*

***Abstract*— Well understanding the access behavior of hot data is significant for NAND flash memory due to its crucial impact on the efficiency of garbage collection (GC) and wear leveling (WL), which respectively dominate the performance and life span of SSD. Generally, both GC and WL rely greatly on the recognition accuracy of hot data identification (HDI). However, in this paper, the first time we propose a novel concept of hot data prediction (HDP), where the conventional HDI becomes unnecessary. First, we develop a hybrid optimized echo state network (HOESN), where sufficiently unbiased and continuously shrunk output weights are learnt by a sparse regression based on L_2 and $L_{1/2}$ regularization. Second, quantum-behaved particle swarm optimization (QPSO) is employed to compute reservoir parameters (i.e., global scaling factor, reservoir size, scaling coefficient and sparsity degree) for further improving prediction accuracy and reliability. Third, in the test on a chaotic benchmark (Rossler), the HOESN performs better than those of *six* recent state-of-the-art methods. Finally, simulation results about *six* typical metrics tested on *five* real disk workloads and on-chip experiment outcomes verified from an actual SSD prototype indicate that our HOESN-based HDP can reliably promote the access performance and endurance of NAND flash memories.**

***Index Terms*— NAND flash memory, solid state disk (SSD), echo state network (ESN), hot data prediction, regularization.**

I. Introduction

With the explosion of information driven by ubiquitous internet access, big-data storage industry is escalating demand for NAND flash-based solid state disks (SSDs) [1]. Compared with hard disk drives (HDDs), SSDs offer higher access speed and better reliability since no mechanical moving components are used, hence NAND flash memories are rapidly expanding into wide applications in consumer electronics and communications [2]. At the same time, today's escalating data traffic and information sharing jointly challenge both on-site distributed and back-end service data storages, affecting user experiences and QoS of enterprises [3]. To satisfy the ever-increasingly strict requirements of diverse applications as well as future storage system demands, more speedy, reliable and energy-efficient SSDs are desirable.

However, NAND flash memory has been facing at least two challenges, out-of-place update and limited endurance, which restrict its large-scale applications. Moreover, multi-level per cell (MLC) technique has been enjoying its popularity for significantly reducing cost by aggressively storing several bits in each transistor [3], [4]. In return, it has come at price in life-span and reliability. An outstanding flash translation layer (FTL) should well resolve the above issues, to enable users to utilize the flash memory like in-place update disks through conventional file systems. As shown in Fig. 1(a), a recycling policy on the FTL, namely garbage collection (GC), is set up for reclaiming the spaces occupied by the invalid data. And a decision policy, namely wear leveling (WL), is established to improve flash lifetime by evenly distributing erases over the entire flash memory. The design idea is to distribute the frequently written data (i.e., hot data) into the blocks that experience lower erase times and distribute the least recently used data (i.e., cold data) into the blocks that have larger erase times. On the basis of above, the identification of hot and cold data plays a vital role to prolong the flash endurance. In other words, both GC and WL are fundamentally determined by the hot data identification (HDI) [5]. Thus, well understanding and utilizing the statistical characterization of application access behavior are significant for NAND flash memory due to their crucial impact on its efficiency and endurance [1], [4].

The past two decades witness considerable efforts on studies of HDI from direct scheme [6], [7] to indirect scheme [5], [8]–[12]. The HDC [6] and two-level LRU [7] analyze the access behaviors by directly recording the statistical rule of logical block addresses (LBAs). In contrast, the scheme of multiple independent hash function (MIHF) [8] captures information of frequency and recency by using multi-hash functions and one bloom filter. Further, the scheme of multiple bloom filter (MBF) [5] achieves more fine-grained identification of hot data behavior by assigning each bloom filter with a discriminative hot degree weight. Based on sparse model for probability distribution of accesses, the scheme of kernel density function (KDF) [9] thus performs better than MBF.

Manuscript received July 24, 2019; revised November 19, 2019; accepted December 7, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 51704089, Grant 61973323, and Grant 61701157, and in part by the Anhui Provincial Natural Science Foundation of China under Grant 1808085QF190 and Grant 1808085QF206. This article was recommended by Associate Editor M. Martina. *(Corresponding author: Chunhua Yang.)*

Q. Luo and C. Yang are with the School of Automation, Central South University, Changsha 410083, China (e-mail: ychh@csu.edu.cn).

X. Fang is with the School of Electrical and Automation Engineering, Hefei University of Technology, Hefei 230009, China.

Y. Sun is with the School of Engineering and Computer Science, University of Hertfordshire, Hatfield AL10 9AB, U.K.

J. Ai is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2019.2960015

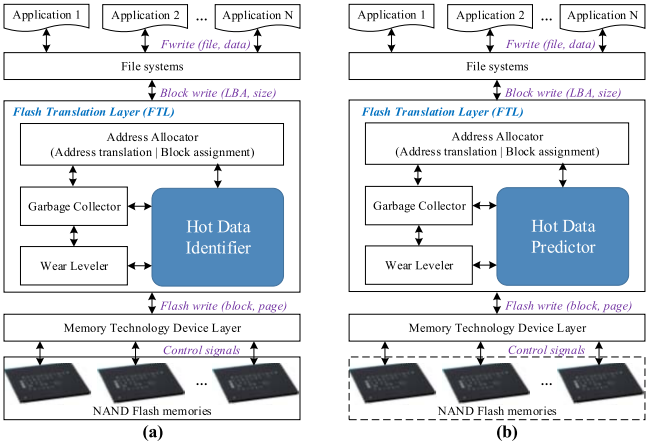


Fig. 1. Typical architecture of NAND flash memory-based systems with (a) traditional hot data identifier and (b) proposed hot data predictor.

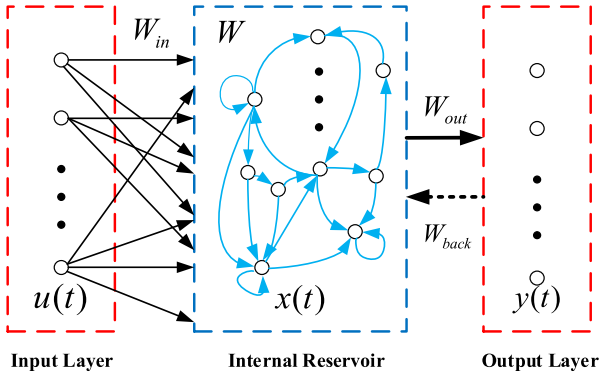


Fig. 2. Structure of echo state network.

Recent HotDataTrap (HDT) [10] and dual layer HDI (DL-HDI) [11] boost the performance to higher stage on both identification accuracy and memory overheads. In summary, all these HDI schemes focus on how to effectively capture the frequency and recency information of write accesses with less runtime consumptions and memory overheads. However, LBA accesses tend to be stochastic and time-variant [5], [10]–[12]. It is challenging to precisely identify the hot ratio of all kinds of workloads even for the recent state-of-the-art HDI schemes.

The essence of HDI is making attempt to well understand the access behavior of hot data so as to intelligently allocate different data to appropriate blocks. Compared with passive identification, is it possible to develop such a scheme that can actively learn or even forecast the rules of data behaviors?

Motivated by this proposal, this paper investigated that reservoir computing (RC) deals particularly well with temporal data classification and prediction task due to its low computational complexity and fast convergence. It is a practical machine learning tool that allows rapid computation on embedded hardware [13]. As one of the most powerful RC schemes, echo state network (ESN) has been innovatively applied in proactive deployment of unmanned aerial vehicles (UAVs) [14], symbol detection in communications [15], containment control of multivehicle systems [16], and adaptive fault tolerant control [17] in the most recent years. Notably, Shafin *et al.* proposed a green symbol detection method based on ESN for MIMO-OFDM systems [15], where the traditional estimation procedure of channel state information (CSI) is completely subverted.

Greatly encouraged by this irresistible trend, as shown in Fig. 1(b), we propose a novel concept of hot data prediction (HDP) where the conventional HDI becomes unnecessary. The main contributions are listed as follows

1) A hybrid optimized ESN (HOESN) is developed, where a sparse regression is adopted to calculate sufficiently unbiased and continuously shrunk output weights by maximizing the merits of $L_{1/2}$ and L_2 regularization.

2) Quantum-behaved particle swarm optimization (QPSO) [18] is inventively employed to compute the vital reservoir parameters (i.e., global scaling factor, reservoir size, scaling coefficient and sparsity degree). This intelligent method can avoid the uncertainty and inconvenience caused by manual parameter settings.

3) Extensive comparative experiments focusing on HDP have been simulated on *five* real disk workloads, and then verified on an actual SSD prototype, which provide a reference case of *different thinking* for improving the performance and endurance of NAND flash memories.

To our best knowledge, *this is the first work exploits the framework of ESN to form an innovative concept of hot data prediction to assist GC and WL*, so as to enhance the performance and endurance of NAND flash memories.

The rest of this paper is organized as follows. Section II presents the theories of HOESN. And its detailed operation procedure, prediction merits, and computational complexity are explained and analyzed in Section III. Section IV demonstrates the evaluation results of HOESN on an open chaotic benchmark. Section V illustrates extensive simulation results carried out on real disk workloads and on-chip experiment outcomes on actual SSD device for HDP and its accommodating DVPFTL. Finally, conclusion is drawn in Section VI.

II. Theories of Hybrid Optimized Echo State Network

A. Echo State Network

A typical ESN includes an input layer, a hidden layer which is a dynamic reservoir with abundant recurrent connected neurons, and an output layer [13]. As shown in Fig. 2, the input layer is linked to the dynamic reservoir via input weights $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{N\times K}$. The dynamic reservoir has internal weights $\boldsymbol{W} \in \mathbb{R}^{N\times N}$. The dynamic reservoir is linked to the output layer through output weights $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{(N+K)\times 1}$. And the output is fed back to the dynamic reservoir through feedback weights $\boldsymbol{W}_{\text{back}} \in \mathbb{R}1\times N$. Intuitively, the distinct connectivity of neurons in the dynamic reservoir makes up of the major difference between ESN and RNN (i.e., recurrent neural network). Traditionally, RNN needs to learn the input and output weights through minimizing the overall mean-square error (MSE). In contrast, ESN only requires to calculate its output weights $\boldsymbol{W}_{\text{out}}$, while the values of $\boldsymbol{W}_{\text{in}}$, $\boldsymbol{W}$ and $\boldsymbol{W}_{\text{back}}$ are randomly chosen with priori. The in-depth idea is to stimulate a random, large but fixed RNN with external

stimuli, which excites reservoir neurons to generate nonlinear responses, and to combine the desired output signals after training through a linear combination among the trained response signals. Thus, ESN is a typical externally linear internally nonlinear network.

Assume input layer has K neurons, dynamic reservoir has N neurons and output layer has L neurons, the reservoir state at time step t is formulated as

$$\begin{cases} \boldsymbol{U}(t) = [u_1(t), u_2(t), \ldots, u_K(t)]^T \\ \boldsymbol{S}(t) = [s_1(t), s_2(t), \ldots, s_N(t)]^T \\ \boldsymbol{Y}(t) = [y_1(t), y_2(t), \ldots, y_L(t)]^T \end{cases} \tag{1}$$

where $\boldsymbol{U} \in \mathbb{R}^{K\times L}, \boldsymbol{S} \in \mathbb{R}^{N\times L}, \boldsymbol{Y} \in \mathbb{R}^{L\times 1}$. At the time step $t+1$, the internal reservoir is updated as

$$\boldsymbol{S}(t+1) = \text{logsig}(\boldsymbol{W}_{in}\boldsymbol{U}(t+1) + \boldsymbol{W}\boldsymbol{S}(t) + \boldsymbol{W}_{back}^T\boldsymbol{Y}^T(t)) \tag{2}$$

where logsig $(\cdot)$ stands for the log-sigmoid transfer function that will be applied to each element. It is worth mentioning that, in practice, after create a random matrix $\boldsymbol{W}_N$ with sparsity SD, $\boldsymbol{W}$ is estimated as $\gamma\,\boldsymbol{W}_N$, where γ is the global scaling factor to be used in iterative learning process. Then the predicted output signal $\boldsymbol{Y}^{\&} \in \mathbb{R}^{L\times 1}$ is obtained by

$$\boldsymbol{Y}^{\&} = [\boldsymbol{U} : \boldsymbol{S}]^T \boldsymbol{W}_{out} = \boldsymbol{X}\boldsymbol{W}_{out} \tag{3}$$

where $[\cdot:\cdot]$ denotes the operation of matrix connection, and $\boldsymbol{X} \in \mathbb{R}^{L\times(K+N)}$ is an L-by-$(K+N)$ intermediate matrix. The output weights $\boldsymbol{W}_{\text{out}}$ are calculated by minimizing the following square error

$$E_{org} = \min \left\| \boldsymbol{Y} - \boldsymbol{Y}^{\&} \right\|_2^2 = \min \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}_{out}\|_2^2 \tag{4}$$

The output weights $\boldsymbol{W}_{\text{out}}$ can be well obtained by Moore-Penrose pseudo-inversion method, it is estimated as $\boldsymbol{W}_{\text{out}}^{\&}$

$$\boldsymbol{W}_{\boldsymbol{out}}^{\&} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \tag{5}$$

There are four key parameters in the dynamic reservoir, jointly determining the final performance of ESN, which are spectral radius (SR), reservoir size ($N \in Z^+$), input layer scaling coefficient ($IS \in (0, 1]$), and reservoir sparsity degree ($SD \in (0, 1)$). The necessary condition (but not strict condition) for ensuring the system stability is that the SR less than one. Finally, the outputs can be estimated as $\boldsymbol{X}\boldsymbol{W}_{\text{out}}^{\&}$.

B. Hybrid Regularization

It is recognized that ill-posed solutions will visit when the high-dimensional reservoir states turn to be fairly correlated in the original ESN. A general method to solve the problem is to import a τ-norm penalty to regularize $\boldsymbol{W}_{\text{out}}$ with certain coefficient λ in the cost function (4). To clarify layout, we denote $\boldsymbol{W}_{\text{out}}$ as ρ, then (4) is modified as

$$\min \left\{ \|\boldsymbol{Y} - \boldsymbol{X}\rho\|_2^2 + \lambda \|\rho\|_\tau^\tau \right\} \tag{6}$$

This paper investigates that the L_2 regularization ($\tau = 2$) works well on continuously shrinking weights but fails to generate sufficiently sparse weights. In contrast, the $L_{1/2}$ regularization ($\tau = 1/2$) can generate extreme sparse solutions, but does not perform well when there is high correlation between predictors. A powerful combination with $L_{1/2}$ and L_2 regularization is inspired as

$$E_{regu} = \min_{\rho_k} \left\{ \|\boldsymbol{Y_k} - \boldsymbol{X}\rho_k\|_2^2 + \lambda_2 \|\rho_k\|_2^2 + \lambda_1 \|\rho_k\|_{1/2}^{1/2} \right\} \tag{7}$$

where λ_1 and λ_2 are non-negative coefficients, $k = 1, 2, \ldots, K$, based on extending formulation, (7) can be derived as

$$\min_{\rho_k^*} \left\{ \left\| \boldsymbol{Y}_{\boldsymbol{k}}^* - \boldsymbol{X}^*\rho_k^* \right\|_2^2 + \mu \left\| \rho_k^* \right\|_{1/2}^{1/2} \right\} \tag{8}$$

where $\boldsymbol{Y}_k^* = \begin{bmatrix} \boldsymbol{Y}k \\ \boldsymbol{0} \end{bmatrix}$, $\boldsymbol{X}^* = \frac{1}{\sqrt{1+\lambda 2}} \begin{pmatrix} \boldsymbol{X} \\ \sqrt{\lambda 2}\boldsymbol{I} \end{pmatrix}$, $\rho_k^* = \sqrt{1+\lambda 2}\rho_k$ and $\mu = \frac{\lambda 1}{\sqrt[4]{1+\lambda 2}}$.

The above (8) can be well solved by coordinate descent algorithm presented in [19]. Given

$$C_{p,k} = \frac{\sum_{t=1}^T (y_p(t) - \sum_{j\neq k}^{N+K} \overline{\beta}_{p,j}^* x_j(t)) x_k(t)}{\sum_{t=1}^T x_k^2(t)} \tag{9}$$

$$\lambda_{p,k} = \frac{\mu}{\sum_{t=1}^T x_k^2(t)} \tag{10}$$

where $p = 1, 2, \ldots, P$ are the index of coordinates. After several derivations, the ρ_k^* in (8) is solved as

$$\rho_{p,k}^* = \begin{cases} \frac{2}{3}C_{p,k}(1 + \cos(\frac{2}{3}\pi - \frac{2}{3}\varphi_{p,k})), & |C_{p,k}| \geq \frac{3}{4}\lambda_{p,k}^{\frac{2}{3}} \\ 0 & |C_{p,k}| < \frac{3}{4}\lambda_{p,k}^{\frac{2}{3}} \end{cases} \tag{11}$$

where

$$\varphi_{p,k} = \arccos(\frac{\lambda_{p,k}}{8}\left|\frac{C_{p,k}}{3}\right|^{\frac{-3}{2}}) \tag{12}$$

Herein, we get a hybrid regularization L_2-$L_{1/2}$-ESN version.

C. Quantum-Behaved Particle Swarm Optimization

The original PSO uses the concept of classical mechanics in which a particle is depicted by its position and velocity, which is very specialized in searching fitting solutions for complex temporal issues. However, it has been reported with drawbacks of substantial parameters tuning and premature convergence. Considering these problems, we adopt QPSO, which has fewer settings and better convergence capability of global optimization [18], to learn the key parameters of our HOESN. In the QPSO, besides evolutionary equation rewriting, the search strategy is also improved by introducing a so-called mean best position (*mbest*), which is the average of the self-best positions (*sbest*) of all particles, to well solve the premature convergence problem of the original PSO.

The particle position ϑ_{i} for dimension j at the $(t+1)^{th}$ iteration is updated according to

$$\vartheta_{ij}^{t+1} = \begin{cases} p_{ij}^t + \beta \times \left| mbest_j^t - \vartheta_{ij}^t \right| \times \ln(1/u_{ij}^t), & if\ u_{ij}^t \leq 0.5 \\ p_{ij}^t - \beta \times \left| mbest_j^t - \vartheta_{ij}^t \right| \times \ln(1/u_{ij}^t), & if\ u_{ij}^t > 0.5 \end{cases} \tag{13}$$

$$\beta = \omega_{\max} - \frac{iter}{iter_{\max}} \times \omega_{\min} \tag{14}$$

$$p_{ij}^t = \varphi_{ij}^t \times sbest_{ij}^t + (1 - \varphi_{ij}^t) \times gbest_j^t \tag{15}$$

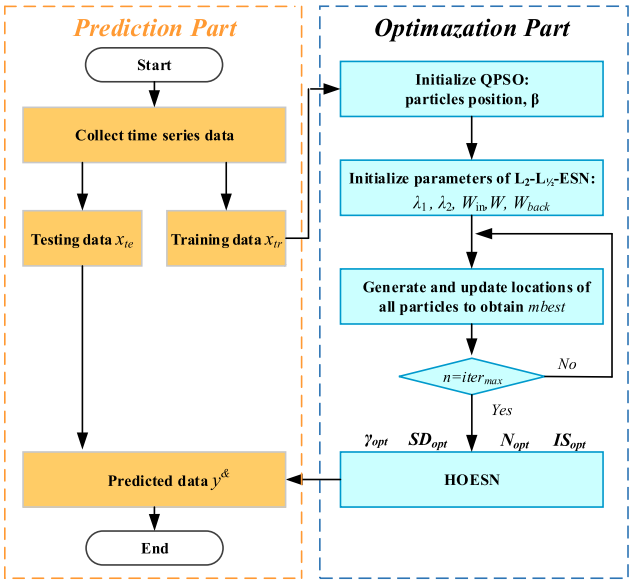


Fig. 3. The flow chart of HOESN.

where $j = 1, \ldots, D$ is the index varying in D problem dimension, and i delegates the current particle. u_{ij}^t is a random number within the range of [0,1]. β is the contraction expansion coefficient (CE), which is the only tunable parameter of QPSO and has crucial impact on controlling its convergence speed, *iter* and $iter_{max}$ denote the current and the maximum number of iterations, respectively, ω_{min} and ω_{max} are the minimum and maximum inertia weight. Obviously, in the early stages of iteration we can get a larger β, then in the late iteration β become minor, which will help us to converge quickly at an early stage and to converge precisely at a later stage. p_{ij}^t is the local attractor of particle i, where φ_{ij}^t is a random number within the range (0,1).

Finally, *mbest* is called mean best position, it is the mean of *sbest* positions of all particles which can be evaluated by

$$mbest_j^t = \frac{1}{N_{ps}} \sum_{i=1}^{N_{ps}} sbest_{ij}, \quad j = 1, \ldots, D \tag{16}$$

where N_{ps} is the population size of particles.

III. Methodology and Analysis

A. Operation Procedure

The QPSO is adopted to optimize the four key reservoir parameters (i.e., global scaling factor γ, reservoir sparse degree *SD*, reservoir size N, and input layer spectral radius *IS*). In each iteration, the L_2 and $L_{1/2}$ regularization will be utilized to constraint the output weights, then determine which set of parameters are the optimal parameters by the fitness value. As shown in Fig. 3, the main procedures are listed as follows.

1) Initialize parameter β by setting $iter_{max}$, ω_{min} and ω_{max} in QPSO, initialize its population size N_{ps}, give each particle a random position within a certain range.

2) Initialize parameters and collect data series of L_2-$L_{1/2}$-ESN, including penalty coefficients λ_1, λ_2, $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}$, training data $x_{tr} \in \mathbb{R}^{1\times dtr}$ and testing data $x_{te} \in \mathbb{R}^{1\times dte}$, where d_{tr} and d_{te} are the length of training data and testing data.

3) Train L_2-$L_{1/2}$-ESN according to (1)-(3), (5) and (7), then obtain the regularized output weights in (11), where (7) help to calculate the fitness value of current particle application in the L_2-$L_{1/2}$-ESN model.

4) Iterate for parameter optimization according to (13)-(16), utilize step *3)* recalculate the fitness value. Compare the current fitness value to the optimal fitness value in history, if smaller, update optimal fitness value with current value of (7), if not, then keep optimal fitness value. The four optimized parameters are updated under the same manner during the same loop.

5) Judge whether iteration ends. If end, jump to step *6)*, if not, jump back to step *4)*.

6) Output the four optimal dynamic reservoir parameters for the prediction model.

Herein, we get the final HOESN with optimal parameters, being ready for the upcoming prediction task.

B. Hot Data Prediction for GC and WL

Both the GC and WL have critical impacts on the access latency and lifetime of flash memory. By collecting hot data to the same block, the garbage collection efficiency can be improved considerably. And the WL is dedicated to prolong the flash endurance by allocating hot data to the flash blocks with a low erase count [10]. Early in 2012, Park [20] reported in his Ph.D. dissertation that predicting the future hot data based on past behaviors of a workload might be beneficial for GC and WL. And an ideal expectation was drawn that if storing only an item (i.e., requested LBA) that will become hot in near future would generate revolutionary HDI scheme.

This paper makes attempt to try the above enlightening conjecture. In specific, an independent block RAM space in FPGA is open up for supporting the HOESN program to learn (or to say predict) the changing trends of access requests from host, rather than to obtain the probability statistics rule from dynamic LBA sequences in HDI. This embedded prediction mechanism is called HDP. It works like a detective running on a compact software RISC processor PicoBlaze (occupies 96 slices in Virtex6-240T, only 0.25% of total) embedded on FPGA to detect the migration trends of accessing LBAs in real time, so as to mine the intrinsic priors of access behaviors (e.g., regularity, periodicity, or homomorphism), these dynamically refreshed information offers firm supports for establishing intelligent GC scheme and cognitive WL strategy. That is, compared with the traditional HDI, the proposed concept of HDP extend the range of descriptive information for hot data from the previous *frequency* (i.e., the number of appearance) and *recency* (i.e., closeness to the present) to *trend* (i.e., in near future). In addition, the mined intrinsic priors of LBAs for a certain workload benefit to dynamically tracking hot LBAs by setting a self-adaptive hot threshold.

The concept of HDP is basically driven by a *concealed yet objective fact* that intrinsic priors exist in access behaviors of users, especially to certain applications in specific

operation scenarios. Notably, many access patterns in workloads exhibit high spatial localities as well as temporal localities [10]. And the predictive hot/cold data clustering can be implicitly achieved through mining intrinsic priors from access LBA sequences. It is worth mentioning that the HDP is with no conflict to any modules in current FTL due to its exclusive block RAM and PicoBlaze. Even if developers plan to inherit the already equipped HDI, it can be compatible with our HDP, while the *trend* information mined by HDP will help the original HDI decrease false identification rates of hot data.

C. Computational Complexity

Computation complexity analysis is a common measure to evaluate the performance of algorithms. As mentioned above, the HOESN is optimized by two significant aspects, QPSO and hybrid regularization. QPSO optimizes reservoir parameters by using historical LBA records during off-line training phase, its computational overheads can be excluded during the prediction process. The remaining task is to analyze the total complexity of the hybrid regularization L_2-$L_{1/2}$-ESN. According to [21], the computational complexity analysis involves two parts of matrix product and function operation in (2) and (3). For $\boldsymbol{W}_{in}\boldsymbol{U}$, $\boldsymbol{WS}$ and $\boldsymbol{W}_{back}^{T}\boldsymbol{U}^{T}$ in (2), the computational complexities of matrix products are $O(NKL)$, $O(NNL)$ and $O(NL)$, respectively. Although there has $N > K$ in general, the hybrid regularization limits most elements of the matrix $\boldsymbol{W}$ to zero so as to maintain the sparsity of reservoir. Thus, the $O(NNL)$ can be deduced to $O(N_{nz}L)$, where N_{nz} stands for the number of non-zero elements in the sparse $\boldsymbol{W}$. When it comes to the function operation in (2), the active function needs to be repeated $N \times L$ times due to $(\boldsymbol{W}_{in}\boldsymbol{U} + \boldsymbol{WS} + \boldsymbol{W}_{back}^{T}\boldsymbol{U}^{T}) \in \mathbb{R}^{N\times L}$, so its complexity is O (NL). As for (3), the matrix products occupy a computational complexity of $O(L(N + K))$. Therefore, the total complexity can be evaluated as O (max(NKL, $N_{nz}L$, NL, $L(N + K)$)). As K, N, N_{nz}, and L are positive integers, they conform to $NK > N + K > N$, $NK > N_{nz}$, so the total computational complexity of HOESN equals $O(NKL)$. Considering that the $\boldsymbol{W}_{out}^{\&}$ in (5) is learnt in advance and required only once, while the prediction process in (2) and (3) is produced repeatedly in HDP application. The total complexity of HOESN can be approximated to the same level of the fundamental ESN.

IV. Performance Evaluation of HOESN

This section carries out a set of performance evaluation tests on an open chaotic benchmark (3-D Rossler chaos), to confirm that our HOESN can be well qualified for the hot data prediction task in advance.

A. Selected Competitors and Evaluation Criteria

The fundamental ESN (FESN) [13] is taken as the baseline, three state-of-the-art enhanced ESN models, i.e., support vector echo state machine (SVESM) [22], ridge regression-ESN (RESN) [23] and adaptive elastic ESN (AEESN) [24], are implemented for longitudinal contrastive analysis. Furthermore, two neural-network-based regression algorithms (i.e., Elman network [25] and extreme learning machine (ELM) [26]) are selected for crosswise contrastive analysis. Three landmark criteria, root mean square error (RMSE), normalized RMSE (NRMSE) and symmetric mean absolute percentage error (SMAPE), are considered for performance verification. They are formulated as follows

$$RMSE = \sqrt{\sum_{t}^{M}[y(t) - \hat{y}(t)]^2/(M-1)} \tag{17}$$

$$NRMSE = \sqrt{\sum_{t}^{M}[\hat{y}(t) - y(t)]^2/(\sum_{t}^{M}[y(t) - \bar{y}(t)]^2)} \tag{18}$$

$$SMAPE = \frac{2}{M}\sum_{t=1}^{M}\left|(y(t) - \hat{y}(t))/(y(t) + \hat{y}(t))\right| \tag{19}$$

where y and $\hat{y}$ are the actual target data and the predicted data, $\bar{y}$ denotes the mean value of actual target data [$y(1)$, $y(2)$, …, $y(M)$], and M is the target data length. Smaller criteria represent better prediction performance.

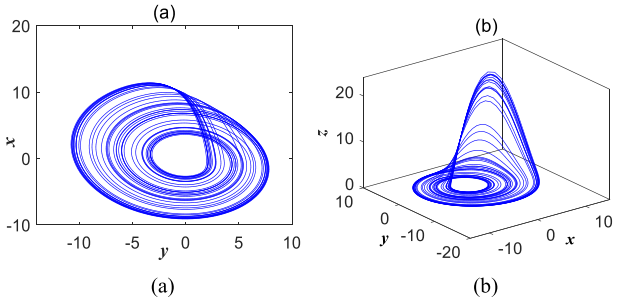


Fig. 4. Portraits of system variables in Rossler system: (a) x-y; (b) x-y-z.

B. Rossler Chaos System and Parameter Setup

As a classical representative, Rossler system can produce sufficiently chaotic phenomenon. It can be expressed as

$$\begin{cases} dx/dy = -y - z \\ dy/dz = x + ay \\ dz/dt = b + z(x - c) \end{cases} \tag{20}$$

where x, y, z are system variables, a, b, c are adjustable coefficients, and t denotes time dimension. When $a = 0.2$, $b = 0.2$, $c = 5.7$, formula (20) produces chaos. Fig. 4 exhibits two typical variable portraits in Rossler chaos system. For fair comparison, we adopt the precisely same conditions in [24] to generate Rossler time series: initial condition $(-1, 0, 3)$, sample step 0.01, and four-order Runge-Kutta method. We also select the same configurations to reconstruct Rossler system with embedding dimensions of (3, 3, 3) and time delays of (13, 13, 13). And QPSO is initialized as follows: the population size of particles $N_{ps} = 20$, the maximum number of iterations $iter_{max} = 100$, the minimum inertia weight $\omega_{min} = 0.3$, and the maximum inertia weight $\omega_{max} = 0.9$.

Referring to [24], 4000 samples are used to train HOESN, where the first 100 of 4000 training samples are washed out for insuring better reservoir status of HOESN. In the

TABLE I
PREDICTION PERFORMANCE OF ESTIMATED MODELS FOR ROSSLER-x

Model	Data	RMSE	NRMSE	SMAPE
Elman [25]		0.1147	0.0239	0.0550
ELM [26]		0.0828	0.0172	0.0211
FESN [13]		0.2293	0.0456	0.1166
SVESM [22]	Data1	0.0321	0.0067	0.0162
RESN [23]		0.0256	0.0053	0.0376
AEESN [24]		0.0165	0.0034	0.0113
HOESN		**0.0023**	**0.0005**	**0.0011**
Elman [25]		0.4496	0.0935	0.3968
ELM [26]		2.3521	0.5607	0.7874
FESN [13]		0.7763	0.1544	0.2647
SVESM [22]	Data2	1.6061	0.3227	1.0265
RESN [23]		0.5412	0.1121	0.6166
AEESN [24]		0.2815	0.0588	0.1934
HOESN		**0.1303**	**0.0186**	**0.0329**

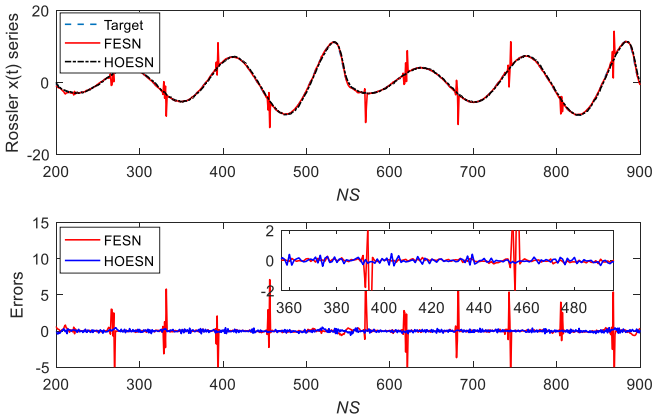


Fig. 5. Prediction performance comparisons of FESN and HOESN for Rossler chaos system with 20 dB Gaussian white noise.

prediction stage, 1000 new Rossler chaotic samples are used to evaluate network performance. Besides, two testing datasets with or without noise interference are involved for estimating anti-noise abilities of competitive models. To be specific, Data1 stands for the noise-free time series, and Data2 denotes the noisy time series suffered with 20 dB Gaussian white noise.

C. Results and Discussions

After the QPSO training, the optimal parameters of HOESN are obtained as follows: $\gamma = 0.14$, $SD = 0.17$, $N = 100$ and $IS = 0.99$ for Data1; $\gamma = 0.27$, $SD = 0.10$, $N = 96$ and $IS = 0.82$ for Data2. The SR can be then deduced as 0.4482 and 0.5624, respectively, which prove that our HOESN has echo state property (ESP) for both Data1 and Data2 with a great probability, because that the values of SR are less than 1.

TABLE I lists the three evaluation criteria of Elman, ELM, FESN, SVESM, RESN, AEESN and our HOESN. Evidently, whether for Data1 or Data2, our HOESN gains the best prediction accuracy. For Data1, FESN performs slightly better than the two NN-based schemes, Elman and ELM, the score is further increased by its two enhanced version, SVESM and RESN. The recent AEESN begins to show significant effect as it has combined the strengths of both lasso and ridge regression. Finally, our HOESN is leading the competition with nearly perfect scores. When it comes to Data2 involving 20 dB Gaussian white noise, all the seven competitors experience performance decrease in varying degrees, but FESN and SVESM suffer more significantly, the main reason is that the estimation method of output weights $\boldsymbol{W}^{\&}_{\text{out}}$ in (5) is quite noise-sensitive to training samples or models. With the regression schemes, the last three predictors have better reliability. Notably, our HOESN ranks first again, with 0.1303 of RMSE, 0.0186 of NRMSE and 0.0329 of SMAPE. For in-depth understanding, the target data, predicted data and errors between them by using FESN and HOESN under noisy condition are illustrated in Fig. 5. Intuitively, FESN suffers with unacceptable burrs and fluctuations. For an example, its error peak is high up to 7.694 when series index NS is 462. In contrast, for our HOESN, the predicted series are even overlapping to the target series, and the error curve experiences much less fluctuations.

TABLE II
WORKLOAD CHARACTERISTICS

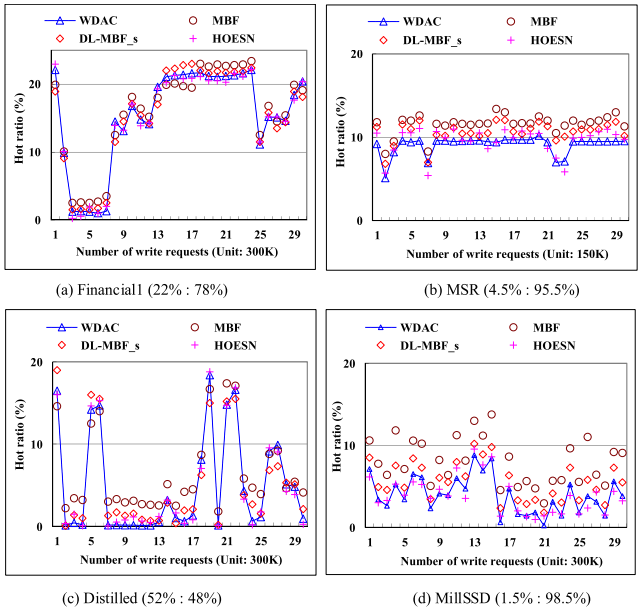
Workloads	Total Requests	Request Ratio (Read: Write)	Inter-arrival Time (Avg.)
Financial1	5,334,987	R: 1,235,633(22%) W: 4,099,354(78%)	8.194ms
Financial2	3,699,194	R: 3,046,112(82%) W: 653,082(18%)	11.081ms
MSR	1,048,577	R: 47,380(4.5%) W: 1,001,197(95.5%)	N/A
Distilled	3,142,935	R: 1,633,429(52%) W: 1,509,506(48%)	32ms
MillSSD	3,581,731	R: 53,726(1.5%) W: 3,528,005(98.5%)	809.27ms

V. EXPERIMENT RESULTS AND DISCUSSIONS

This section presents extensive experimental results and comparative analyses. First, the performance improvement of SSD brought by the proposed HOESN-based HDP was evaluated quantitatively in various aspects. Second, the overall access speed of HDP-based DVPFTL is demonstrated on our previously implemented on-chip prototype.

A. Performance Simulation Results

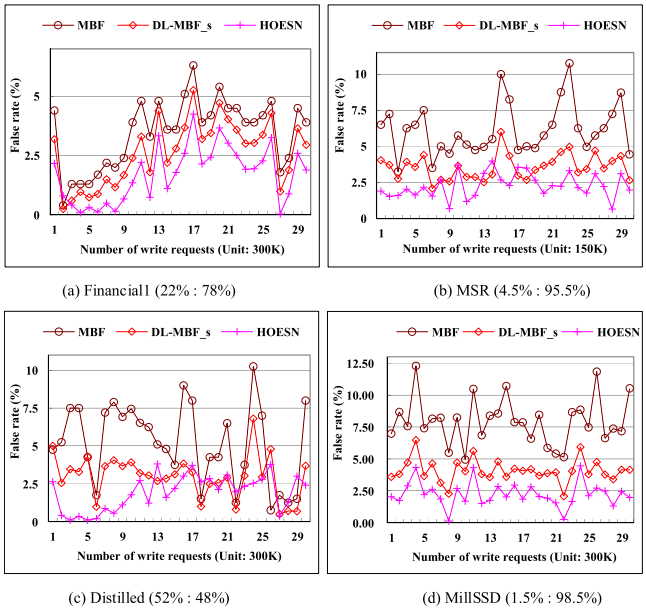
1) Simulation Setup: We compare the HDP with one state-of-the-art HDI scheme, MBF [5], and our recently proposed HDI scheme, DL-MBF_s [11]. To give a more explicit picture, the window-based direct address counting (WDAC) [5] is used as the testing baseline. All the parameters (i.e., size, number and width of bloom filter, hash function number and type, *et al.*) are inherited from [11]. Similar to the testing procedure in [5], [11], we adopted *five* real workloads for objective evaluation. As shown in TABLE II, *Financial1* is a write intensive trace file [27], *Financial2* is a read intensive trace file [27], *MSR* is a common workload of large enterprise servers [28], *Distilled* represents typical usage patterns in a personal computer, and lastly [7], *MillSSD* is gathered from an industrial surface defect inspection device [29], with hardware configurations of Runcore RCS-V-T25 SSD, Intel X2 7400 and 2G DDR3. *MillSSD* is also a write intensive trace file due to its role for substantial image backup.



(a) Financial1 (22% : 78%) (b) MSR (4.5% : 95.5%)

(c) Distilled (52% : 48%) (d) MillSSD (1.5% : 98.5%)

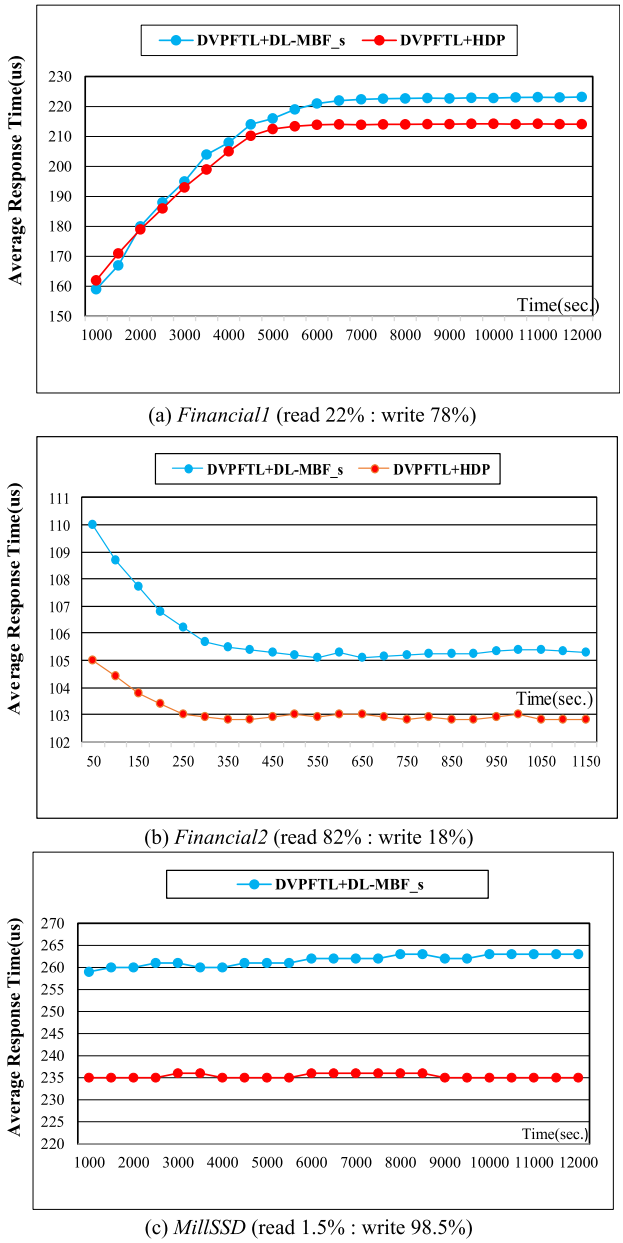
Fig. 6. Hot ratios under various workloads (read %: write %).



(a) Financial1 (22% : 78%) (b) MSR (4.5% : 95.5%)

(c) Distilled (52% : 48%) (d) MillSSD (1.5% : 98.5%)

Fig. 7. False identification rates under various workloads (read %: write %).



(a) *Financial1* (read 22% : write 78%)

(b) *Financial2* (read 82% : write 18%)

(c) *MillSSD* (read 1.5% : write 98.5%)

Fig. 8. Performance of average response time.

2) Simulation Results and Analysis: Two typical performance metrics of our proposed HDP, *hot ratio* and *false identification rate*, are evaluated in this section in Fig. 6 and Fig. 7. Furthermore, *four* key metrics from FTL performance aspect, *average response time*, *number of block erase operations*, *memory cost* and *energy consumption*, are tested in Fig. 8, Fig. 9, TABLE III and Fig. 10. To be fair, this part of tests is evaluated under our DVPFTL framework.

a) Hot ratio: A hot ratio is a ratio of hot data to all data [5]. Fig. 6 illustrates the hot ratios of MBF, DL-MBF_s and HOESN compared to those of WDAC. Taking WDAC as the benchmark, the identified hot ratio curve of MBF fluctuates near its upside, which indicates the bloom-filter-based HDI scheme is effective to classify hot and cold data. The improved DL-MBF_s provides a similar trajectory, the better identification effect mainly benefits from its dual-layer structure with a pre-classifier to effectively drop cold data in advance. As for our HOESN, the hot ratio curves nearly overlap with those of WDAC most of the time. This main trend can be clearly found under all four workloads, especially to more write intensive *MSR* and *MillSSD*. The results show that our forecasting method can well learn the access behavior of disk workloads, which is the basic precondition to provide reliable service for GC and WL.

b) False identification rate: Even though both hot ratios of two algorithms are identical, hot data classification results of both schemes may be able to be different since an identical hot ratio means the same number of hot data to all data and does

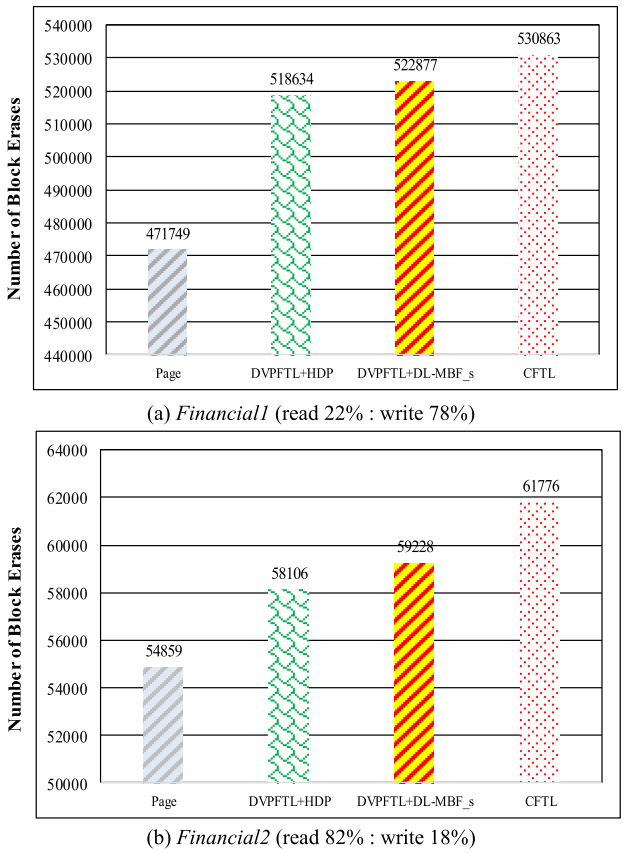


Fig. 9. Performance of number of block erases.

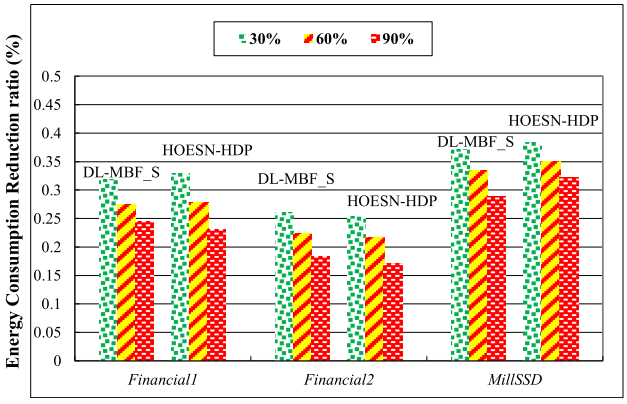


Fig. 10. Performance of energy consumption, both are based on DVPFTL.

not necessarily mean all classification results are identical [5]. Hence, we analyze the *false identification rate* (*FIR*), which is the ratio of mis-identified (or mis-predicted) hot data to actual hot data, to learn more details of the prediction scheme. Whenever write requests are issued, we try to compare each predicted result of each scheme. Fig. 7 exhibits the FIRs of MBF, DL-MBF_s and HOESN. It is fairly clear that, under four workloads, the FIRs of our HOESN is lowest, followed by DL-MBF_s. Although MBF experiences relatively higher FIRs, it is still a good HDI scheme for SSDs, where the WDAC was proposed, which becomes a classical benchmark for the following research. It is noticed that, among the four workloads, the improvement degree of HOESN is the most impressive for *MillSSD* (from 4.08% to 2.23%), the main reason may be that the behaviors of data access tend to be regular and stable.

c) Average response time: Tests in Fig. 8 mainly focus the metric of *average response time* which are sensitively affected by the overhead of a garbage collection and address translation time as well as system service time. For write intensive trace file *Financial1*, as shown in Fig. 8(a), benefited from our active prediction scheme, the DVPFT+HDP enters the convergence time region earlier than DVPFTL+DL-MBF_s, not only that, the average response time of DVPFT+HDP is about 12 us less than that of DVPFTL+DL-MBF_s when both fall into stable time region. When it comes to the *Financial2*, evidently in Fig. 8(b), compared with the DL-MBF_s, our HDP scheme decreases the average response time firmly by some 2.2 us almost all the time. Next, a more regular and write intensive workload (*MillSSD*) is tested. As shown in Fig. 8(c), the curves of the two schemes show smooth features, and DVPFTL+HDP reduces average response time by about 27us compared to that of DVPFTL+DL-MBF_s. This encouraging result indicates that more regular and write-intensive workloads can placidly converge in the early stage and be precisely predicted by our HDP. The main reason is that the current HOESN version is trained in offline, imagine the access behavior fluctuates violently, the adaptability achieved by the HOESN with pre-trained weights would degrade to some extent.

d) Number of block erase operations: It is worth noting that erase operations are almost 10 times slower than write operations and over 100 times slower than read operations. This part chooses the number of block erase operations as a representative measure of the SSD performance driven by diverse FTL frameworks and hot data management schemes. Taking the page-based FTL as the baseline and the CFTL equipped with MBF-based HDI [5] as a state-of-the-art case, we evaluate the performance of the aforementioned two DVPFTL schemes, DVPFTL+HDP and DVPFTL+DL-MBF_s on *Financial1* and *Financial2*. As shown in Fig. 9(a), both DVPFTL-based schemes have smaller number of block erases than CFTL, while the page-based FTL ranks first, but the corresponding cost is that it needs the highest memory consumption. Notably, the DVPFTL+HDP performs better than the DVPFTL+DL-MBF_s, which proves that the active prediction mechanism of HDP possesses evident advantage to the traditional passive HDI schemes regarding analysis performance of access behavior. Further, Fig. 9(b) targets to the read intensive workload *Financial2*, the improvement degree of DVPFTL-series is slightly more remarkable than those observed in Fig. 9(a), one explanation could be that HDP tracks read intensive workload better since this kind of request tends to access more adjacent LBAs.

e) Memory cost: In the TABLE III, we have calculated the basic memory costs of our HOESN scheme and the contrastive schemes. It is clear learned that the proposed HOESN-based HDP consumes 1.547 KB RAM when the reservoir size N, input layer size K, prediction length L, time delay, and embedding dimension are set to 16, 5, 5,

TABLE III
MEMORY SPACE OVERHEADS AND RECENCY STEPPING

*Para.1	HDP	The state-of-the-art HDI			#Para.2
	HOESN	MIHF [8]	MBF [5]	DL-MBF [11]	
u of $\boldsymbol{U}$	15 B	2^{12} *bit*	2^{11} *bit*	2^{11} *bit*	M
$\boldsymbol{W}_{\text{in}}$	320 B				
$\boldsymbol{W}$	1024 B	1	4	5	V
x of $\boldsymbol{X}$	54 B				
$\boldsymbol{W}_{\text{out}}$	84 B	4	1	1	T
window	40 B				
MC	1.547 KB	2 KB	1 KB	1.25 KB	MC
RWS	0.1	0.5	0.5	0.5	RWS

Notes: LBA length = 3B, variable length = 4B, MC denotes the sum of memory costs, and RWS is the recency weight stepping, 'B' means Byte.

* N=16, K=5, time delay = 2, embedded dimension = 5, $u \in \mathbb{R}^{K\times 1}$, $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{N\times K}$, $\boldsymbol{W} \in \mathbb{R}^{N\times N}$, $x \in \mathbb{R}^{N\times 1}$, $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{(N+K)\times 1}$.

M is the bloom filter (BF) size, V and T are BF number and width.

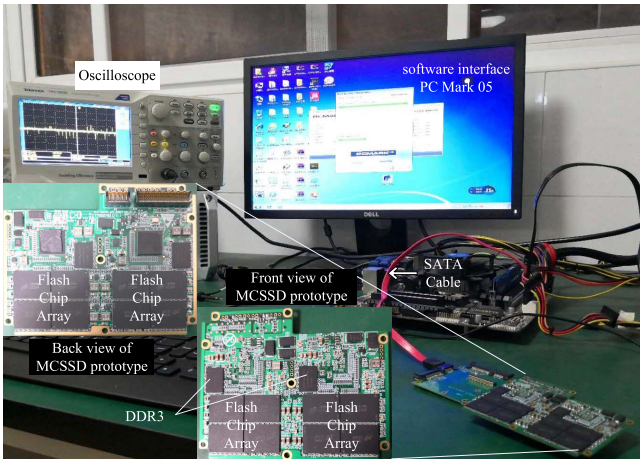


Fig. 11. Prototype photo and testing scene.

2, and 5, respectively. Under this compact configuration, the memory cost of the HOESN-HDP is comparable to those of MIHF [8], MBF [5] and DL-MBF_s [11]. While we can get more fine-grained recency weight stepping of 0.1, compared with 0.5 of others, which mainly benefits from the added 5 more predicted LBAs among the 10 total LBAs (with *recency* and *trend*) in the hot/cold clustering window. It is worth mentioning that, we can increase the time delays in HOESN to down-sample the LBA sequences like that did in the HDT in [10], this operation can not only discard infrequently accessed LBAs (i.e., cold data) in advance, but also save memory costs and CPU clocks.

f) Energy consumption: The total energy consumption of SSD mainly consumes by flash memory chips, RAM and FPGA resources (i.e., slices, multipliers). For fair comparison, both the schemes of DL-MBF_s and HOESN-HDP are running on DVPFTL, we choose the page-based FTL as the identical baseline. For simplicity, we present only the energy consumption reduction ratio of the tested scheme to the page-based FTL. As analyzed above, the two tested schemes consume comparable RAMs. Thus, for DL-MBF_s, we only considered the energy consumption on flash memory itself, which is dominantly brought by the erase and write operations. While for HOESN-HDP, besides the flash memory itself, the extra resources occupied by PicoBlaze and multipliers will consume energy. In addition, the free space of flash memory will affect the energy consumptions as GC frequency is closely related with it, which has also been considered in this test.

Intuitively in Fig. 10, the reduction ratios show a downward trend with the increase of SSD free space in all cases. It is not difficult to understand that, the more free space, the weaker GC and WL demand, and the smaller energy consumption. Thus, the case of with 90% free space is most challenging to obtain energy reduction. Notably, for the write intensive *MillSSD*, even in case of with 90% free space, the HOESN-HDP has 3.4% advantage of energy reduction to that of DL-MBF_s (32.2% *vs.* 28.8%), and the other records also show reliable leaderships. When it comes to the read intensive *Financial2*, the HOESN-HDP consumes higher energy than DL-MBF_s. It means that the energy saving obtained by HOESN cannot cover its own energy consumption if the workload is with weak needs for GC and WL (e.g., in this test, most requests are reads.). Interestingly, for write intensive *Financial1* (more irregular), our scheme of HOESN-HDP help the DVPFTL win the first two rounds of competitions on energy consumption reduction ratios. These preliminary test results on energy consumption indicate the HOESN-based HDP possesses advantages on hot/cold data clustering especially on regular and write intensive workloads due to its active prediction mechanism. However, as the HOESN requires some basic multiply-add operations, we need to pay attention to its natural energy consumption during the application process.

In summary, the above test results prove our initial thought that the problem of hot data identification can be converted to active hot data prediction for NAND flash memories. Follow this roadmap, the statistical analysis of access behavior can be considered as time series prediction problem.

B. Overall On-Chip Performance Results

1) Prototype Glance and Test-Bench Setup: This whole set of experiments are realized in our previously designed multi-chipped SSD (MCSSD) in [11], in particular, all the configurations in [11] are inherited in this paper except that the HDI scheme of DL-MBF_s in DVPFTL has been replaced with our HOESN-based HDP. Some key hardware details are summarized as follows: The model of FPGA is XC6VLX240T, which has been pre-configured with an embedded 32-bit software processor, MicroBlaze. The whole MCSSD prototype includes 16 NAND flash memory chips with model of MT29F64G08AFAAAWP. There are abundant

communication interfaces such as SATA 3.0, G-bit Ethernet, and UART. The realized MCSSD and its test scene photo are presented in Fig. 11.

For fair comparison, the classical version, PCMark05, continues to be applied especially for covering the previous proposed state-of-the-art work (i.e., Hydra in [30]). The storage benchmark which contains five kinds of workloads of *OS Startup*, *Application Loading*, *General Usage*, *Virus*

Scan, and *File Write* has been used for emulating typical PC environments. The workloads details are listed as follows:

OS Startup emulates the boot-up behavior of Windows XP startup operations, the ratio of request quantity of reading to writing operations is about 90% : 10%.

Application Loading includes about 83% read and 17% write operations generated by some frequently used applications, such as Chrome, Microsoft Office, etc.

General Usage stands for the daily usage of a PC, which consists of about 60% read and 40% write.

Virus Scan represents host requests operated when scanning 600 MB of files for viruses, and read operations dominate all the requests (about 99.5% read).

File Write contains host requests for writing 680 MB of files.

2) Overall Performance of HDP-Based DVPFTL SSD: In this part, we select two state-of-the-art FTL schemes (Hydra [30], CFTL+MBF [5]) and one our previously designed FTL scheme (DVPFTL+DL-MBF_s [11]) for comparative experiments. And the proposed method in this paper is denoted as DVPFTL+HDP. To be as fair as possible, the last three FTL schemes are verified on the MCSSD prototype with an identical 2-way-4-channel architecture, while the test result of Hydra is directly fetched from [30]. Fig. 12 demonstrates the access performance of four methods by using five test workloads. Overall, CFTL and DVPFTL-series yield better performance than Hydra, the main reason is that the block-level mapping scheme cannot completely overcome the inherent limitation of low write performance. In contrast, the core mapping table of CFTL is a page-level mapping so that CFTL inherits the outstanding write performance well. And the schemes of hybrid-level mapping and the dynamic virtual page prevent DVPFTL-series far away from low access performance. As expected in [11], DVPFTL+DL-MBF_s performs slightly better than CFTL in most cases, which mainly benefited from its cold data eviction ability provided by the double-layer structure of DL-MBF_s. This technical route of pre-discarding cold data has also played an important role in [3], which motivates this paper to find any breakthrough from hot data analysis. Consequently, the concept of HDP is proposed to this end. More positively, for all the five workloads, DVPFTL+HDP performs promisingly better than DVPFTL+DL-MBF_s. With the identical DVPFTL, we can draw a conclusion that the HDP scheme achieves better accuracy of hot data recognition than the DL-MBF_s, so that acquires higher throughput speed for MCSSD. It is worth emphasizing that the neural-network-based HDP tracks the rules of regular operations better, hence more performance improvements can be observed for the workloads of *Virus Scan* and *File Write* which involve significantly read or write intensive operations. Interestingly, for the *general usage*, the testing scores experience a sharp decline first and then a rise for the FTL schemes of CFTL+MBF, DVPFTL+DL-MBF_s, and DVPFTL+HDP, which indicates that the unsatisfactory performance of DVPFTL on handing workloads interlaced with massive random reads and writes pointed out in [11] has been improved by our proposed HDP to some extent. However, compared with other three methods, our scheme of

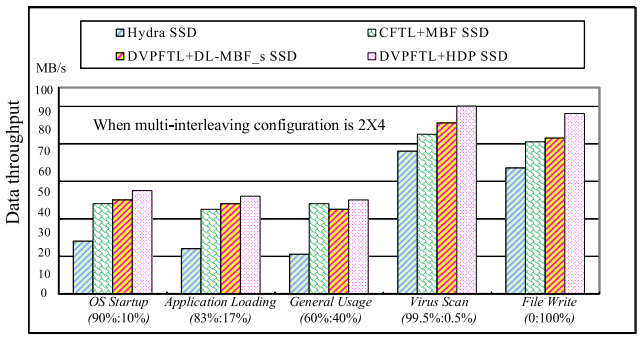


Fig. 12. On-chip scores with various workloads (read % : write %).

DVPFTL+HDP consumes maximum memory for holding the proposed HOESN-based HDP, although our MCSSD prototype is able to accommodate them.

VI. Conclusion

SSD storage plays an important role in today's fields of communications and electronics. The average download speed will be higher than the current storage access speed when 5G communication is widely applied. It is not wise to wait until that day to boost the performance of storage. Hot/Cold data clustering is a significant precondition to improve access performance and life-span of NAND flash memory. This paper focuses on how to precisely track the access behavior of hot data, so as to well serve the garbage collection and wear leveling. Inventively, based on an echo state network, a novel concept of neural-network-based hot data prediction is drawn in this paper. For improving the prediction accuracy and noise robustness, a hybrid optimized echo state network (HOESN) is built based on output weight regularization and initial parameter optimization. In this model, ill-posed solutions are avoided to a large extent, sufficiently sparse and continuously shrunk output weights are calculated by $L_{1/2}$ and L_2 regularization. In addition, reservoir parameters are learnt through a quantum-behaved particle swarm optimization to further improve prediction accuracy and flexibility. HOESN is first verified on a classical Rossler chaos system, then is tested with *five* real disk workloads. The extensive results indicate that our HOESN performs better on multivariate chaotic time series prediction than several classical neural networks (Elman [25], ELM [26], FESN [13], SVESM [22], RESN [23] and AEESN [24]), and also produces more satisfactory performance on hot data classification than some recent HDI schemes (i.e., MBF [5] and DL-MBF_s [11]. Both the simulation and on-chip verification results indicate our method performs better than Hydra [30], CFTL+MBF [5] and DVPFTL+DL-MBF_s [11].

However, the resource overheads greatly determine whether the HOESN-based hot data prediction scheme could be accepted by storage fields. Reducing hardware complexity and time overhead without compromising prediction accuracy will become our focus in the near future. As a simple and efficient network, HOESN is hopeful to be more lightweight. In the meanwhile, a new scheme of quasi-online training is under way, which is expected to improve the adaptability of HDP to

workload changes by periodically updating the weights $\boldsymbol{W}_{\text{out}}$ of HOESN.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to Dr. Q. Xie, a senior engineer from RunCore Co., Ltd., who assisted in verifying the performance of the prototype in this paper, and Mr. Y. Wang, who assisted in data analysis of this paper.

REFERENCES

[1] C. Sun, T. O. Iwasaki, T. Onagi, K. Johguchi, and K. Takeuchi, "Cost, capacity, and performance analyses for hybrid SCM/NAND flash SSD," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 8, pp. 2360–2369, Aug. 2014.

[2] S. Tanakamaru, M. Doi, and K. Takeuchi, "NAND flash memory /ReRAM hybrid unified solid-state-storage architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1119–1132, Apr. 2014.

[3] C. Sun, K. Miyaji, K. Johguchi, and K. Takeuchi, "A high performance and energy-efficient cold data eviction algorithm for 3D-TSV hybrid ReRAM/MLC NAND SSD," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 2, pp. 382–392, Feb. 2014.

[4] J. Moon, J. No, S. Lee, S. King, S. Choi, and Y. Song, "Statistical characterization of noise and interference in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 8, pp. 2153–2164, Aug. 2013.

[5] D. Park and D. H. C. Du, "Hot data identification for flash-based storage systems using multiple bloom filters," in *Proc. IEEE MSST*, Denver, CO, USA, May 2011, pp. 1–11.

[6] M. W. Lin, S. Y. Chen, G. P. Wang, and T. S. Wu, "HDC: An adaptive buffer replacement algorithm for NAND flash memory-based databases," *Optik*, vol. 125, no. 3, pp. 1167–1173, 2014.

[7] L.-P. Chang and T.-W. Kuo, "An adaptive striping architecture for flash memory storage systems of embedded systems," in *Proc. IEEE RTAS*, Sep. 2002, pp. 187–196.

[8] J.-W. Hsieh, T.-W. Kuo, and L.-P. Chang, "Efficient identification of hot data for flash memory storage systems," *ACM Trans. Storage*, vol. 2, no. 1, pp. 22–40, Feb. 2006.

[9] J. Liu, S. Chen, T. Wu, and H. Zhang, "A novel hot data identification mechanism for NAND flash memory," *IEEE Trans. Consum. Electron.*, vol. 61, no. 4, pp. 463–469, Nov. 2015.

[10] D. Park, B. Debnath, Y. Nam, D. H. C. Du, Y. Kim, and Y. Kim, "HotDataTrap: A sampling-based hot data identification scheme for flash memory," in *Proc. ACM SAC*, Trento, Italy, Mar. 2012, pp. 1610–1617.

[11] Q. Luo, R. C. C. Cheung, and Y. Sun, "Dynamic virtual page-based flash translation layer with novel hot data identification and adaptive parallelism management," *IEEE Access*, vol. 6, pp. 56200–56213, 2018.

[12] C. H. Wu, P. H. Wu, K. L. Chen, W. Y. Chang, and K. C. Lai, "A hotness filter of files for reliable non-volatile memory systems," *IEEE Trans. Depend. Sec. Comput.*, vol. 12, no. 4, pp. 375–386, Jul. 2015.

[13] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, Apr. 2004.

[14] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[15] U. Challita, W. Saad, and C. Bettstetter, "Deep reinforcement learning for interference-aware path planning of cellular-connected UAVs," in *Proc. IEEE ICC*, Kansas City, MO, USA, May 2018, pp. 1–7.

[16] Z. Peng, J. Wang, and D. Wang, "Distributed containment maneuvering of multiple marine vessels via neurodynamics-based output feedback," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 3831–3839, May 2017.

[17] L. Liu, Z. Wang, X. X. Yao, and H. Zhang, "Echo state networks-based data-driven adaptive fault tolerant control with its application to electromechanical system," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 3, pp. 1372–1382, Jun. 2018.

[18] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proc. IEEE CEC*, Portland, OR, USA, Jun. 2004, pp. 325–331.

[19] Y. Liang *et al.*, "Sparse logistic regression with a $L_{1/2}$ penalty for gene selection in cancer classification," *BMC Bioinf.*, vol. 14, no. 1, Jun. 2013, Art. no. 198.

[20] D. Park, "Hot and cold data identification: Applications to storage devices and systems," Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. Minnesota, Minneapolis, MN, USA, Jul. 2012, pp. 38–58.

[21] Y. Zhao, H. Gao, N. C. Beaulieu, Z. Chen, and H. Ji, "Echo state network for fast channel prediction in Ricean fading scenarios," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 672–675, Mar. 2017.

[22] Z. Shi and M. Han, "Support vector echo-state machine for chaotic time-series prediction," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 359–372, Mar. 2007.

[23] X. Dutoit, B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, and M. Nuttin, "Pruning and regularization in reservoir computing," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1534–1546, Mar. 2009.

[24] M. Xu and M. Han, "Adaptive elastic echo state network for multivariate time series prediction," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2173–2183, Jul. 2016.

[25] M. C. Ozturk, D. Xu, and J. C. Príncipe, "Analysis and design of echo state networks," *Neural Comput.*, vol. 19, no. 1, pp. 111–138, Jan. 2007.

[26] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, pp. 306–314, Oct. 2005.

[27] UMASS. (2002). *OLTP Trace From UMass Trace Repository*. [Online]. Available: http://traces.cs.umass.edu/index.php/Storage/Storage

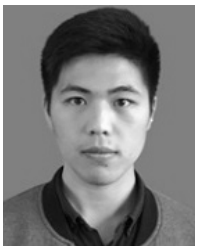
[28] Microsoft. (Dec. 2019). *SNIA IOTTA Repository: MSR Cambridge Block I/O Traces*. [Online]. Available: http://iotta.snia.org/traces/list/BlockIO

[29] Q. Luo and Y. He, "A cost-effective and automatic surface defect inspection system for hot-rolled flat steel," *Robot. Comput.-Integr. Manuf.*, vol. 38, pp. 16–30, Apr. 2016.

[30] S. L. Min *et al.*, "Hydra: A block-mapped parallel flash memory solidstate disk architecture," *IEEE Trans. Comput.*, vol. 59, no. 7, pp. 905–921, Jul. 2010.

Qiwu Luo (M'17) received the B.S. degree in communication engineering from the National University of Defense Technology, Changsha, China, in 2008; and the M.Sc. degree in electronic science and technology and the Ph.D. degree in electrical engineering from Hunan University, Changsha, in 2011 and 2016, respectively.

He was a Senior Engineer of instrumentation with WASION Group Ltd. Company, Changsha, and the Deputy Technical Director with Hunan RAMON Technology Co., Ltd., Changsha. In 2016, he joined the School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, China, where he also completed his postdoctoral research on automatic optic inspection (AOI). Since 2019, he has been an Associate Professor with the School of Automation, Central South University, Changsha. His current research interests include computer vision, industrial AOI, machine learning, parallel hardware architecture design, and reconfigurable computing.



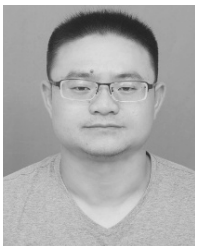
Xiaoxin Fang received the B.S. degree in electrical engineering and automation from the Jiangsu University of Science and Technology in 2018. He is currently pursuing the M.Sc. degree with the School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, China, under the guidance of Dr. Luo.

His current research interests include texture analysis, image classification, and machine learning.

Yichuang Sun (M'90–SM'99) received the B.Sc. and M.Sc. degrees from Dalian Maritime University, Dalian, China, in 1982 and 1985, respectively, and the Ph.D. degree from the University of York, York, U.K., in 1996, all in communications and electronics engineering.

He is currently a Professor of communications and electronics, the Head of the Communications and Intelligent Systems Research Group, and the Head of the Electronic, Communication and Electrical Engineering Division, School of Engineering and Computer Science, University of Hertfordshire, U.K. He has published over 330 articles and contributed ten chapters in edited books. He has also published four text and research books: *Continuous-Time Active Filter Design* (CRC Press, 1999), *Design of High Frequency Integrated Analogue Filters* (IEE Press, 2002), *Wireless Communication Circuits and Systems* (IET Press, 2004), and *Test and Diagnosis of Analogue, Mixed-signal and RF Integrated Circuits—The Systems on Chip Approach* (IET Press, 2008). His research interests are in the areas of wireless and mobile communications, RF and analogue circuits, microelectronic devices and systems, and machine learning and deep learning.

Dr. Sun was a Series Editor of *IEE Circuits, Devices and Systems* Book Series from 2003 to 2008. He was a Guest Editor of eight IEEE and IEE/IET journal special issues: High-frequency Integrated Analogue Filters in *IEE Proceedings Circuits, Devices and Systems* in 2000, RF Circuits and Systems for Wireless Communications in *IEE Proceedings Circuits, Devices and Systems* in 2002, Analogue and Mixed-Signal Test for Systems on Chip in *IEE Proceedings Circuits, Devices and Systems* in 2004, MIMO Wireless and Mobile Communications in *IEE Proceedings Communications* in 2006, Advanced Signal Processing for Wireless and Mobile Communications in *IET Signal Processing* in 2009, Cooperative Wireless and Mobile Communications in *IET Communications* in 2013, Software-Defined Radio Transceivers and Circuits for 5G Wireless Communications in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-II: EXPRESS BRIEFS in 2016, and the 2016 IEEE International Symposium on Circuits and Systems in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I I: REGULAR PAPERS in 2016. He has been an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: REGULAR PAPERS from 2010 to 2011, from 2016 to 2017, and from 2018 to 2019. He is also an Editor of *ETRI Journal*, *Journal of Semiconductors*, and *Journal of Sensor and Actuator Networks*. He has also been widely involved in various IEEE technical committee and international conference activities.



Jiaqiu Ai (M'17) received the B.S. degree in electronics and information from Beijing Information Science and Technology University in 2007 and the Ph.D. degree in information and communication from the University of Chinese Academy of Sciences in 2012.

He served as a Senior Engineer with the 38th Institute, China Electronic Technology Group Corporation (CETC), from 2012 to 2016. He is currently an Associate Professor with the Hefei University of Technology, Anhui, China. He has authored over 20 journal articles. His current research interests include SAR image processing, artificial intelligent, radar target detection, and radar system design.

Chunhua Yang (M'17) received the M.S. degree in automatic control engineering and the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 1988 and 2002, respectively.

From 1999 to 2001, she was a Visiting Professor with the University of Leuven, Leuven, Belgium. From 2009 to 2010, she was a Senior Visiting Scholar with the University of Western Ontario, London, ON, Canada. Since 1999, she has been a Full Professor with the School of Information Science and Engineering, Central South University. She is currently the HOD with the School of Automation, Central South University. Her current research interests include modeling and optimal control of complex industrial processes, and intelligent control systems.