Computers, Materials & Continua
DOI:10.32604/cmc.2022.019889
Article

Tech Science Press

A Novel Workload-Aware and Optimized Write Cycles in NVRAM

J. P. Shri Tharanyaa[1],[*], D. Sharmila[2] and R. Saravana Kumar[3]

[1]Department of ECE, Bannari Amman Institute of Technology, Tamil Nadu, India
[2]Department of CSE, Jai Shriram Engineering College, Tamil Nadu, India
[3]Department of ECE, Bannari Amman Institute of Technology, Tamil Nadu, India
*Corresponding Author: J. P. Shri Tharanyaa. Email: sshri2148@gmail.com

Received: 29 April 2021; Accepted: 17 September 2021

Abstract: With the emergence of the Internet of things (IoT), embedded systems have now changed its dimensionality and it is applied in various domains such as healthcare, home automation and mainly Industry 4.0. These Embedded IoT devices are mostly battery-driven. It has been analyzed that usage of Dynamic Random-Access Memory (DRAM) centered core memory is considered the most significant source of high energy utility in Embedded IoT devices. For achieving the low power consumption in these devices, Non-volatile memory (NVM) devices such as Parameter Random Access Memory (PRAM) and Spin-Transfer Torque Magnetic Random-Access Memory (STT-RAM) are becoming popular among main memory alternatives in embedded IoT devices because of their features such as high thickness, byte addressability, high scalability and low power intake. Additionally, Non-volatile Random-Access Memory (NVRAM) is widely adopted to save the data in the embedded IoT devices. NVM, flash memories have a limited lifetime, so it is mandatory to adopt intelligent optimization in managing the NVRAM-based embedded devices using an intelligent controller while considering the endurance issue. To address this challenge, the paper proposes a powerful, lightweight machine learning-based workload-adaptive write schemes of the NVRAM, which can increase the lifetime and reduce the energy consumption of the processors. The proposed system consists of three phases like Workload Characterization, Intelligent Compression and Memory Allocators. These phases are used for distributing the write-cycles to NVRAM, following the energy-time consumption and number of data bytes. The extensive experimentations are carried out using the IoMT (Internet of Medical things) benchmark in which the different endurance factors such as application delay, energy and write-time factors were evaluated and compared with the different existing algorithms.

Keywords: Internet of things; DRAM; PRAM; STT-RAM; machine learning; internet of medical things; endurance

1 Introduction

Internets of Things (IoT) technologies are growing exponentially, finding their application in various domains such as healthcare, automation and security. These devices are equipped with batteries, sensors, microcontrollers and transceivers. Since these devices are battery-driven, making the IoT node sensitive to battery lifetime is a tricky mechanism. Recent reports suggest that using DRAM-based memory systems is the main contributor to embedded devices' overall energy utility to solve this significant issue, Non-Volatile Random-Access Memory (NVRAM) finds its place in terms of DRAM memory systems. Non-volatile memory devices, including Phase Change Memory (PCM) as well as Resilient Random-Access Memory (ReRAM), are playing a vital role in building future memory systems due to their salient features like high scalability and low power consumption [1–5]. However, NVRAMs suffer from shortcomings like less endurance and high energy read/write cycles [6] when compared with the existing DRAM systems. Hence these constraints impose serious issues in using the NVRAM as the complete replacement for DRAM.

Several studies were investigated in improving the endurance of NVRAM in terms of reducing and effectively managing the memory write cycles [7,8]. Flip-N-Write (FNW) [9] becomes the method that combines old and new data to lessen the number of bits that help reducing redundancy between similar caches. Moreover, frequency compression techniques [10,11] was used to compress the data bits for an efficient write mechanism. Specifically, these methods retain a template table in the program memory and include several typical word patterns, including a highly responsible template. If all the word patterns in the table are matched, each word in the in-memory cache is represented by a compact format, which decreases the number of text bits. The term patterns in the template chart are predetermined and cannot be updated, defined as static patterns.

To achieve the above-mentioned challenge, methodologies such as Dynamic Frequency Compression techniques [12], Flash Translational Layer (FTL) mapping techniques [13–18] were proposed to reduce the high energy write levels, thus increasing the endurance levels of NVRAM. However, these methods find it difficult to handle many applications when ported in Embedded IoT devices.

To improve the endurance of the memory systems for large applications, intelligent analysis of larger applications is required, which can impart flexible and adaptive write cycles. Addressing the observations from the literary works, we put forward the new adaptive NVRAM write schemes called WHEAL (Workload Hybrid Energy Adaptive Learning) mechanism. The proposed technique works on dynamic compression techniques based on the workload characterizations with an effective NAND flash memory management. Moreover, WHEAL also expanded the complex pattern granularity from 32 bits to 64 bytes. In specific, our interventions are summed up as follows.

a. **Intelligent Workload Characterization**: In contrast to the previous schemes, which require more statistical and extraction methods for obtaining the data characteristics, we proposed a new powerful machine-learning algorithm to classify the data types based on the workload characterization. The proposed technique increases the versatility of data compression mechanisms in a cost-effective way.

b. **Workload-Energy-Aware NVRAM Compression Schemes**: The proposed write schemes introduce the new methodology Dynamic Workload Compression (DWC), which writes data to fit into dynamic compressive schemes. The DWC expands the data pattern for compacting an all-zero cache while leveraging the value localization. In addition, this methodology preserves the energy with the reduced mechanism.

c. **Endurance Aware Allocation Methods**: The proposed schemes introduce the compression and assign the adaptive write cycles for the intelligent allocation of the compressed workload bits in the caches following its characteristics to increase the memory's endurance.

The rest of the paper is organized as follows: Section 2 presents the related works of various literature. Section-2 also converses about the research gaps found in the existing systems. The preliminary overviews of NVRAM-Based Cache Memories and the motivation behind our research are detailed in Section-3. The complete working mechanism of the proposed framework is presented in Section-4. The implementation results with comparative analysis, performance metrics are exhibited in Section 5. Lastly the paper is concluded in Section-6.

2 Related Works

Power and performance optimization on embedded architectures is still a critical challenge. Many existing approaches are focused on designing an efficient software kernel for targeted architecture. On the other hand, hardware optimization in terms of design space area and memory optimization is still an open challenge problem to be focused on nowadays. In this literature survey, we initially discussed the workload characterization on embedded architectures followed by non-volatile memory optimization.

Workload Characterization can be utilized to foresee future asset pre-requisites that enhance the scope organization, task planning and resource utilization effectively. The remaining burden portrayal is commonly performed by utilizing two unique methodologies i.e., execution-based and model-based methods. In the model-based framework, the workloads are characterized using the performance tool [19] in terms of memory occupation, registers and other peripheral usages of execution of applications on the hardware platforms. Wang et al. [20] introduced multi level cell based phase change memory architecture. The significant value of this framework is relies upon energy variations in programming. To further enhance the performance, this framework incorporates data comparison write and results shows that this framework outperformed other existing methodologies in terms of energy consumption utilized for write cycles.

Shishira SR et al. studied performance improvement achieved through workload characterization [21]. The author initially defined the workload characterization with its various classes such as CPU, GPU, cloud workloads, physical environment, resource utilization and how the execution characteristics differ among each workload. The various tasks have distinct traits and the choice of the proper forum for a specific workflow processing is still an NP-hard task. The major limitation is a review paper, in which hypothetically detailed workload characterization and its classification are detailed. Maria Calzarossa et al. [22] developed and used a series of static tracks to establish task models based on the network usage and processing times.

Alexander et al. developed a workload characterization model that focuses on multimedia application workloads. Multiprocessor system-on-chip is targeted for processing multimedia workloads in real-time. The workloads are categorized and differentiated based on the variability characterization curves and workload transformations [23]. The limitation of this work, the proposed framework is more suitable for multimedia workloads on the MPSoC platform. Memory, I/O workloads are not considered in this work. Zhang et al. [24] utilize regulation methods to describe the vector task of roles with dependent implementation sequence. This classification framework is orthogonal to our design in the context that we are not just attempting to model timeline relationships inside tasks triggered by dependent implementation. Baruah et al. studied a real time task model in the perspective of feasibility examination. This framework incorporates 2 models called generalized multi framework model and sporadic task model. It is a powerful framework and easily handle both static and dynamic tasks in real time [25].

Writam Banerjee [26] considered real-time challenges on non-volatile memory gadgets with different structures. The later advance faced the "memory wall", i.e., the speed crevice within rationale and memory. Overcoming these issues, primary framework execution bottleneck and essential confinements related to contracting gadget measure and expanded processing complexity, developing NVM (eNVM) with energizing models have been introduced in this article. As per the recent survey, eNVM devices called FeFET, PCM, STTRAM and RRAM are the most assured memory devices that can be used for high-efficient performance with low cost and power. Likewise, Jishen Zhao et al. reviewed the low-cost architecture of non-volatile memory, byte-addressed non-volatile memory (NVM) architectures, including spin-transfer torque memory (STTMRAM), phase-change memory (PCM) and resistive memory (ReRAM), as just a substitute for conventional memory systems used throughout the memory models [27].

Yuncheng Guo et al. developed an NVM writing scheme called Dynamic Frequent Pattern Compression (DFPC) to dramatically decrease the number of write units and increase their lifespan. The DFPC model comprises various stages such as sampling and dynamic patterns to improve the compression of data. Then, an enhanced DFCP algorithm is developed to optimize the latency and energy efficiency of NVM devices [12].

3 Methodology

3.1 NVRAM Background

NVM, Characteristics and Drawbacks unlike conventional charge-based memory such as DRAM and SRAM, evolving NVMs store information using series resistance memories with increased concentration and scalability. Therefore, NVMs can be commonly used in the primary memory [27].

Because all NVMs store data by modifying physical characteristics, the write operation acquires more time and resources than that of the read operation, contributing to the imbalance of reading and writing operations. Also, the write operation uses the NVM cell, particularly at high frequency, which results in reduced NVM durability. NVM-based structures also have to lessen the number of write-bits in the writing process. As one of the successful NVM innovations, PCM technology uses tolerance of chalcogenide glass for data storage.

The element used for phase shift is germanium, antimony and tellurium alloys, including Ge2Sb2Te5. The substance has two possible states, the state of crystalline (SET) and the state of amorphous (RESET). The sensitivity of the substance to various states is fundamentally different. In particular, the tolerance rate of the amorphous form is much greater than those of the crystal-state material. PCM stores binary information in states by using material tolerance differences.

In need of executing a RESET (SET) procedure to write "0" ("1") to the PCM cell, the PCM cell is warmed just above the melting point that melts the chalcogenide substance, followed by immediate chilling to adjust the structure. The ReRAM system utilizes a permeable dielectric in the Metal-Insulator Metal structure. This can be adjusted among low-resistance state (SET) or high-resistance state (RESET) while using the necessary voltages.

3.2 Motivation

(i) Achieving high-endurance with low power is still an Np-hard issue. Workload characterization scheduling improves resource utilization and performance through static model-based optimization, which misleads specific architecture and specific applications. Existing approaches are not suitable for mixed-critical workloads and are not designed for common architectures.

(ii) Memory (NVRAM) optimization is another emerging problem in recent research. However, few researchers have addressed the various non-volatile memory optimization structures which are not implemented on experimental boards.

(iii) There are many inherent problems with eNVM technologies, like cell-level and system-level durability, variability, device performance, extremely smooth framework architecture, etc. Nevertheless, the eNVM paves the way to overwrite primary memories, play SCM's function, explore new technologies, explore brain-inspired computational systems and model hardware safety systems.

(iv) The low-power eNVM could also be beneficial for sensors, including Smart devices. However, there are also several obstacles in developing large eNVM systems, such as the production process, components and tailored operation for various products.

4 Proposed Work

4.1 Proposed WHEAL-NVRAM Architecture

The proposed WHEAL-NVRAM-based architecture has been shown in Fig. 1. The new workload awareness-compression and new wear-levelling approaches have been integrated to wear out the low energy cycles. In addition, the new lightweight learning algorithms are incorporated in memory to categorize the nature of the workloads. Based on the nature of workloads, novel compression and new adaptive-levelling and allocation technique were adopted to reduce the write cycle in the NVRAM for enhancing its endurance.

4.2 Workload Characterization

This section discusses the workload parameters and intelligent workload characterization based on the lightweight machine learning algorithms.

4.3 Workload Parameters

Tab. 1 illustrates the workload parameters which are extracted from the different input application programming threads. These workloads are extracted by using the PSutils software, which runs on the kernel's architecture.

These workloads are considered the micro architecture-independent workload parameters used as input to the proposed learning model for efficient workload characterization. The major advantage of using the micro architecture-independent workloads is to capture the actual inherent program behaviours and prove its usefulness in characterizing suites for emerging workloads.

4.4 Machine Learning Based Workload Extraction Mechanism

Several algorithms such as Clustering, principal component analysis, histogram analysis, correlation methods were used for typical workload characterization. However, an accurate and intelligent workload characterization technique is still mandatorily needed for a better compression and levelling technique. This purpose can be served by using the machine learning algorithms and the proposed system incorporates a single-layer feedforward learning model for effective workload characterization.

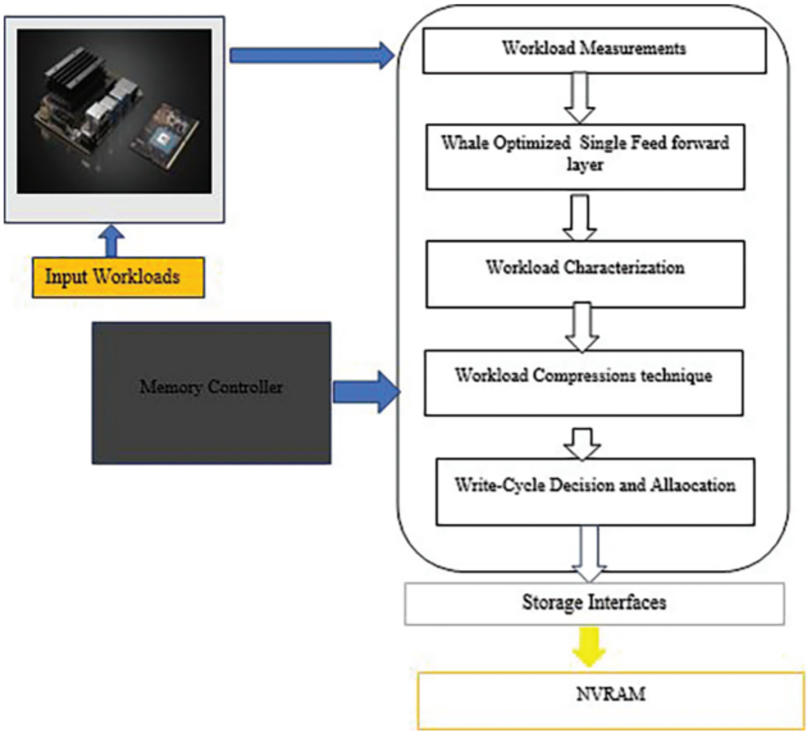
Input Workloads
Memory Controller
Workload Measurements
Whale Optimized Single Feed forward layer
Workload Characterization
Workload Compressions technique
Write-Cycle Decision and Allaocation
Storage Interfaces
NVRAM

Figure 1: Proposed framework for the WHEAL-architectures

Table 1: Workload parameters used for the characterization

S no	Workload parameters	Specifications
01	Register usage	Defines the number of registers in the workloads
02	Instruction mix	Defines load, store, multiplication, addition operations
03	Branch predictability	Defines the branch statements in workloads
04	Memory size	Defines the number of memory size used for storing the workload
05	Instruction pipelining	Number of pipelining stages used in the input workloads

The single-layer feedforward networks are based on the Extreme Learning Machines (ELM). Fig. 2 shows the architectural diagram for the proposed ELM.

This infrastructure uses a single hidden layer, high speed, precision, speed planning and highly speculative and standard feature estimation skills. Hence, these algorithms are portable to embedded memory storage systems. In the same kind of model, the 'K' neurons in the hidden layer are expected to deal via an activation function that is very distinguishable while the output layer component is linear. In ELM, hidden layers are need not be tuned. Loads of the hidden layer are randomly allocated (counting loads of the bits). It is not the situation that hidden nodes are meaningless, but they do not need to be tuned and hidden neuron parameters can be haphazardly generated even in advance. Those are, until having to take care of a learning data collection. For a single layer ELM, the performance of the method is specified by Eq. (1)

$$f_K(a) = \sum_{i=1}^{K} \beta_i h_i(a) = h(a)\beta \tag{1}$$

where a is the input

β be the output weight vector

$$\beta = [\beta_1,\ \beta_2,\ \ldots \beta_K]^T \tag{2}$$

H(a) is the hidden layer output

$$h(a) = [h_1(x), h_2(x), \ldots h_K(a) \tag{3}$$

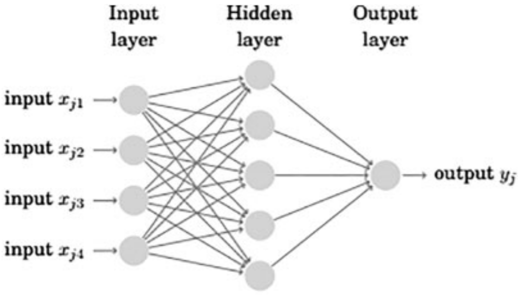

Figure 2: Structure of Extreme Learning Machines

To formulate the output vector S that is considered as the target, the eqn can further be derived as follows

$$H = [h(a_1)h(a_2) \vdots h(a_N)] \tag{4}$$

The general implementation of the ELM is based on the minimum non-linear least square approaches expressed in Eq. (5)

$$\beta' = H^* S = H^T (HH^T)^{-1} S \tag{5}$$

where H* is the inverse of H known as Moore–Penrose generalized inverse

$$\beta' = H^T (\frac{1}{C} HH^T)^{-1} S \tag{6}$$

The output can be derived as

$$f_K(a) = h(a)\beta = h(a)\ H^T(\frac{1}{C}HH^T)^{-1}S \tag{7}$$

4.5 *Disadvantages of Existing ELM*

Although Extreme Learning Machines seem to be effective both in preparation and practice, the key downside was its non-optimal calibration of input weights. ELM often uses several hidden units to change the weight values relative to many other traditional learning strategies, which can influence the accuracy of the identification.

To address the above downside, a novel Whale algorithm is used to optimize input connection weight variables to reach the optimal accuracy of the task classification system. There seems to be a significant trend in Whale optimization in recent decades. This evolutionary algorithm model is a computational approximation of the action and activity of humpback whales in their quest for food and supplies. Whale Optimization Algorithm (WOA) has been influenced by the Bubble-net assault technique, whereby whales begin catching fish by forming spiral-shaped bubbles surrounding their fish to 12 meters depth from the sea and then dive back and caught their intended prey. Centred on the shortest distance of the whales, the discovery phase in this method is a randomized quest for food that can be numerically converted by modifying old strategies rather than picking the right ones by randomly assigning other alternatives. In addition to this curious action, WOA separates it from other evolutionary algorithms because it only expands two values. These attributes determine a rapid integration among extraction and exploration processes. Fig. 3 shows Encircling attack prey searching methodology for humpback whales.

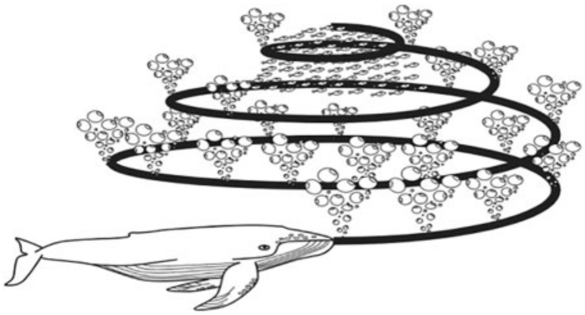

Figure 3: Encircling attack prey searching methodology for humpback whales

We will explain the computational formula for prey conquering, prey seeking, spiral bubble-net hunting and gathering in the later subsection. Surrounding Prey: By raising the number of loops from the beginning to the optimum number, humpback whales encircle the prey and refine their location in the path of the best solution. This action can be expressed mathematically as:

If (p < 0.5 and mod(U) < 1)

Then the position of the candidate X(t + 1) is updated and

$$\mathrm{D} = \mathrm{mod}\{(\mathrm{C.X}) - \mathrm{X(t)}\} \tag{8}$$

$$\mathrm{X(t+1)} = [\mathrm{X(t)} - \{\mathrm{U.D}\}] \tag{9}$$

where $p = 0.1$ (constant) $X(t + 1)$ is the best position in the current situation. U and D are calculated by the following Eqs. (9) and (10)

$$U = \mathrm{mod}\{2.a.r - a\} \tag{10}$$

$$C = 2.r \tag{11}$$

where a decrease linearly from 2 to 0 and r is the randomly selected vector

Prey Searching: In prey searching mechanism, X is replaced with the random variables X_{random} and mathematical equation are given by

$$D = \mathrm{mod}\{(C.X_{random}) - X(t)\} \tag{12}$$

$$X(t + 1) = [X_{random}(t) - \{U.D\}] \tag{13}$$

The encircling of prey and spiral updation of prey have been done during the exploration phase of the whale optimization algorithm. The mathematical expression for updation of new position during the spiral process is given by Eq. (14)

$$X(t + 1) = D^{l}.e^{bl}.\cos(2\pi l) + X^{*}(t) \tag{14}$$

Here, D is the distance among the new position and updated position in the new generation, b is the constant, which varies from the 0 to 1.

4.6 Optimized ELM for Workload Characterization

The main limitation of the ELM is that the non-optimum collection of hidden units can trigger the formation of increasingly prevalent that impair the prediction accuracy. The proposed ELM network implements a whale algorithm to maximize input weight and bias variables to address this issue. The benefit of the whale algorithm in ELM is that it improves the global minimum search path, which can be more effective than the current optimization techniques. Throughout this case, accuracy is used as a feature of fitness. If the classification accuracy is equivalent to the accuracy of the standard, then the output variables will be deemed correct else, they will be ignored and their iterations will begin. This optimized learning model saves energy consumption and area overhead when implemented as the API in the Embedded Systems kernel. Hence the Eq. (7) can be modified as

$$f_L(x) = Optimized\ \{h(x)\}\beta = h(x)\ H^{T}\left(\frac{1}{C}HH^{T}\right)^{-1} O \tag{15}$$

Eq. (15) depicts the final output from the proposed learning models, which is used to categorize the different types of the workloads such as Very Heavy (VH), Heavy (H), Medium (M) and Normal (N). Before categorizing the workloads by the proposed learning models, workload thresholds are calculated using the mathematical expression. Then, these values are used to label the workloads, which makes an effective categorization using the proposed learning models.

4.7 Threshold Decision

After calculating the workload parameters using the PSutils tools in the Embedded systems, energy and instruction per clock cycles are calculated using the decision rule for categorizing the workloads. The energy of each workload is calculated by using the mathematical expression, which is given as follows

$$E(W) = V(CPU) * I(CPU) * IPC(Memory) \tag{16}$$

where E(W) is the Energy of the workloads, V is the voltage of the CPU, I is the current consumption and IPC is Clock cycles for writing in the memory.

The Eq. (16) is used for labelling the workloads that better categorize workloads by the optimized machines. Based on the energy calculated, proposed learning models categorizes the different workloads as very heavy (VH), heavy (H), medium (M) and normal (N), which act as the inputs to the compressor and allocators.

4.8 Intelligent Workload Compression and Allocation

After the extractions of the workloads, these workloads are compressed by the dynamic compression technique and new threshold-based workload levelling is adopted to allocate the application threads in the cache memories. The architecture which is used for implementing the compression and allocator technique is shown in Fig. 4.

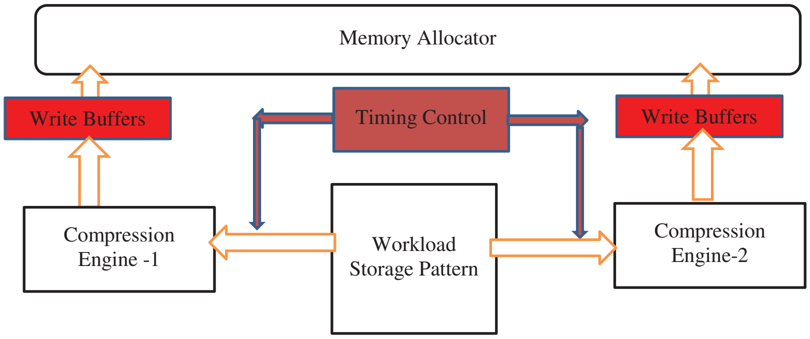

Figure 4: Memory controller architecture for compression and allocator

The proposed memory controller architecture consists of two compression engines, a workload pattern table and an allocator. All these are controlled by the timing circuits, which control the workload patterns to write in the buffers. The working mechanism of the WHEAL memory controller is given as follows.

4.8.1 Workload Storage Pattern

These storage systems store the workload categorized by the proposed learning models and the categorization bits. These categorization bits are used to differentiate the workloads and following that, compression will take place concurrently for the different loads.

4.8.2 Compression Engine

The approach splits the entire 64-byte cache line as 16 32-bit words to perform data compression for different workloads. Further, the categorization bits are used to check the type of workloads, which avoids the repeated type of workloads. Also, these compression techniques work based on sampling time which is provided by the timing control circuits.

5 Experimentation

The proposed framework is simulated in a GEM-5 simulator emulated on the Raspberry Pi 3 Model B+ hardware. To analyze the workload characteristics, IoMT benchmarks were considered and implemented on hardware Raspberry Pi Model B+. Then the proposed NV-RAM architecture is simulated on GEM-5 software. The complete specification used for the experimentation is presented in Tab. 2.

Table 2: Specifications used for the experimentation

Sl. no	Specifications	Descriptions
01	Hardware emulation	Raspberry Pi model B+
02	Simulator	GEM05
03	Clock frequency used	600 MHZ to 1.4 GHZ
04	Cache memoires	I and D caches with 32 KB/32 KB
05	Programming for hardware emulation	Micropython
06	Embedded CPU used	Cortex A-53

For analysis, IoMT benchmarks are taken as input workload datasets in which 70% of workloads are used for training and 30% are used for testing. The different analysis of the proposed learning model in categorizing the workloads are given as follows.

Figs. 5–8 show the performance analysis of the different machine learning models in categorizing the different workloads. Fig. 5 shows the performance metrics of different learning models in categorizing the very heavy workloads in which the proposed algorithm has exhibited 98.5% accuracy, 98% sensitivity and 97.5% specificity, which has an edge of performances of 2% than SVM (Support Vector machines), 3.5% than KNN (K-nearest neighbourhood), 4% than DT (Decision Tree), 5% than ANN-1 (Artificial Neural Networks-Backpropagation layer) and ANN-2 (Artificial Neural Networks-Feed forward layer) and finally 6% than Naïve Bayes algorithms. A similar fashion of performance is found in Figs. 6–8 in categorizing the different types of workloads. The integration of the optimized hidden layers in ELM has proved its stability in categorizing the workloads with high efficiency. Moreover, to prove the performance of the proposed framework, time of categorizing is calculated, which is shown in the figures,

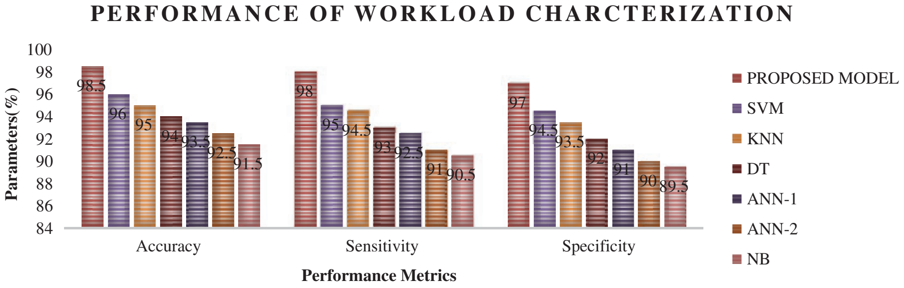

Figure 5: Performance analysis for different learning models for categorizing the very heavy workloads using IoMT benchmarks

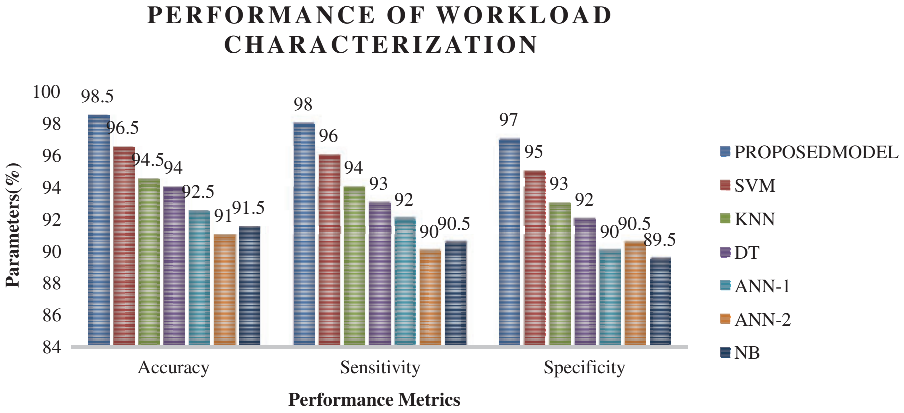

Figure 6: Performance analysis for different learning models for categorizing the heavy workloads using IoMT benchmarks

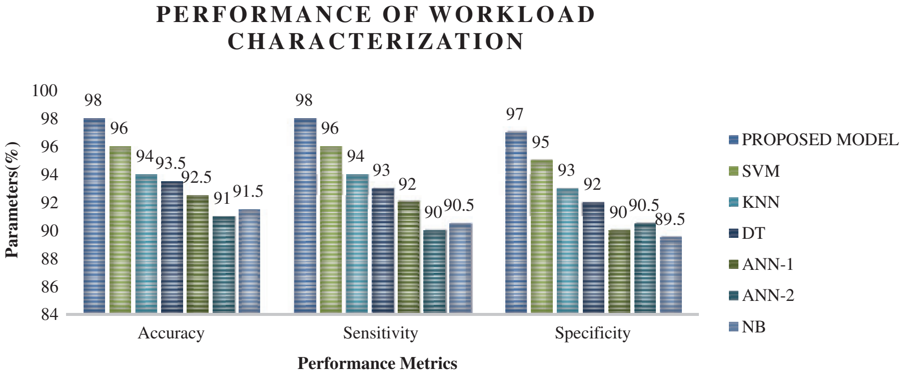

Figure 7: Performance analysis for different learning models for categorizing the medium workloads using IoMT benchmarks

From the above analysis, it is found that the proposed model consumes less time with high performance, which proves that these models can be integrated into the hardware, which can consume lesser energy and lesser overhead.

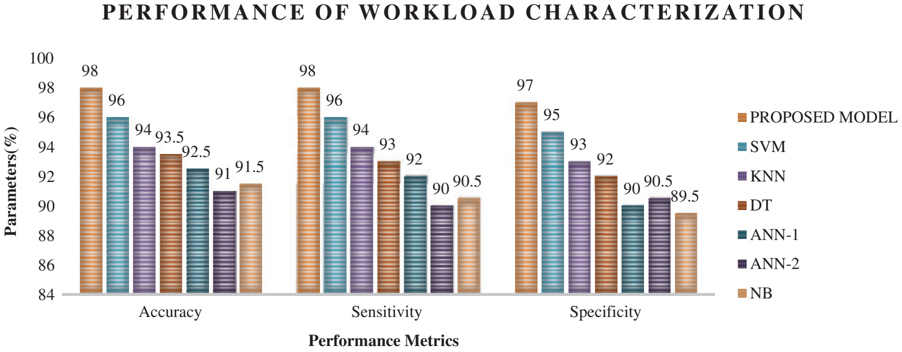

Figure 8: Performance analysis for different learning models for categorizing the normal workloads using IoMT benchmarks

6 Conclusion

The paper proposes the first-ever hybrid machine-learning-based approaches for increasing the endurance in NVRAM. The proposed WHEAL methodology incorporates different stages such as workload categorization, compression technique and a memory allocator. The first phase is workload categorization in which the new energy-saving optimized ELM technique is used for the workload categorization where the energy is taken as the major threshold. Also, the paper proposes the dual compression technique for compressing the bits and the memory allocator stores the loads by using adaptive write cycles in different caches. Extensive experimentations have been conducted and performance metrics such as accuracy of categorization, write frequency ratio and write latencies were analyzed and compared with other existing architectures such as Baseline architecture and A-CACHE controllers. As a result, the proposed model has increased the lifetime of NVRAMS by 50% greater than the A-CACHE controller and even > 90% greater than baseline architectures. Eventhough this technique can increase NVRAM's endurance, computational overheads need its improvisation for less complexity implementation.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Z. Li, R. Zhou and T. Li, "Exploring high-performance and energy proportional interface for phase change memory systems," in *Int. Symposium on High Performance Computer Architecture*, Shenzhen, China, pp. 210–221, 2013.

[2] L. Wilson, "International technology roadmap for semiconductors," *In Semiconductor Industry Association,* Washington, DC, USA, 2011.

[3] H. Zhang, X. Chen, N. Xiao and F. Liu, "Architecting energy-efficient STT-RAM based register file on GPGPUs via delta compression," *IEEE Design Automation Conference,* pp. 1–6, Article No.: 119, 2016.

[4] B. C. Lee, E. Ipek, O. Mutlu and D. Burger, "Phase change memory architecture and the quest for scalability," *Communications of ACM,* vol. 53, no. 7, pp. 99–106, 2010.

[5] B. C. Lee, P. Zhou, J. Yang, Y. Zhang and B. Zhao *et al.*, "Phase-change technology and the future of main memory," *IEEE Micro,* vol. 30, no. 1, pp. 143, 2010.

[6] P. Zhou, B. Zhao, J. Yang and Y. Zhang, “A durable and energy efficient main memory using phase change memory technology,” *International Symposium on Computer Architecure,* vol. 37, no. 3, pp. 14–23, 2009.

[7] Z. Li, F.Wang, D. Feng, Y. Hua and W. Tong *et al.*, “Tetris write: Exploring more write parallelism considering PCM asymmetries,” in *Int. Conf. on Parallel Processing*, Philadelphia, PA, USA, pp. 159–168, 2016.

[8] L. Jiang Du, Y. Zhao, B. Zhang, Y. Childers and J. Yang *et al.*, “Hardware-assisted cooperative integration of wear levelling and salvaging for phase change memory,” *ACM Transactions on Architecture and Code Optimization,* vol. 10, no. 2, pp. 1–25, 2013.

[9] S. Cho and H. Lee, “Flip-N-write: A simple deterministic technique to improve PRAM write performance, energy and endurance,” in *Int. Symposium on Microarchitecture*, New York, NY, USA, pp. 347–357, 2009.

[10] A. R. Alameldeen and D. A. Wood, *Frequent Pattern Compression: a Significance-Based Compression Scheme for L2 Caches,* Department of computer science, University of Wisconsin-Madison, Madison, WI, USA, Rep. 1500, 2004.

[11] D. B. Dgien, P. M. Palangappa, N. A. Hunter, J. Li and K. Mohanram, “Compression architecture for bit-write reduction in non-volatile memory technologies,” in *Int. Symposium on Nanoscale Architectures*, Paris, France, pp. 51–56, 2014.

[12] Y. Guo, Y. Hua and P. Zuo, “DFPC: A dynamic frequent pattern compression scheme in NVM-based main memory,” in *Design, Automation & Test in Europe Conference & Exhibition*, pp. 1622–1627, 2018.

[13] C. H. Wu and T. W. Kuo “An adaptive two-level management for the flash translation layer in embedded systems,” in *Int. Conference on Computer-Aided Design*, San Jose, CA, USA, pp. 601–606, 2006.

[14] Y. H. Chang, J. W. Hsieh and T. W. Kuo, “Endurance enhancement of flash-memory storage systems: an efficient static wear leveling design,” in *IEEE Design Automation Conf.*, San Diego, CA, USA, pp. 212–217, 2007.

[15] Y. Wang, D. Liu, M. Wang, Z. Qin and Z. Shao *et al.* “RNFTL: A reuse-aware NAND flash translation layer for flash memory,” in *Languages, Compilers and Tools for Embedded Systems Conference*, Stockholm, Sweden, pp. 163–172, 2010.

[16] Y. Wang, D. Liu, Z. Qin and Z. Shao “An endurance-enhanced flash translation layer via reuse for NAND flash memory storage systems,” in *Conference on Design, Automation and Test in Europe*, Grenoble, France, pp. 1–6 152 D, 2011.

[17] Z. Qin, Y. Wang, D. Liu D and Z. Shao “A two-level caching mechanism for demand-based page-level address mapping in NAND flash memory storage systems,” in *IEEE Real-Time and Embedded Technology and Applications Symposium*, Chicago, IL, USA, pp. 157–166, 2011.

[18] Z. Qin, Y. Wang, D. Liu, Z. Shao and Y. Guan “MNFTL: an efficient flash translation layer for MLC NAND flash memory storage systems,” in *IEEE Design and Automation Conf.*, San Diego, CA, USA, pp. 17–22, 2011.

[19] B. C. Lee, E. Ipek, O. Mutlu and D. Burger, “Architecting phase change memory as a scalable dram alternative,” *International Symposium on Computer Architecture,* vol. 37, no. 3, pp. 2–13, 2009.

[20] J. Wang, X. Dong, G. Sun, D. Niu and Y. Xie, “Energy-efficient multi-level cell phase-change memory system with data encoding,” in *IEEE Int. Conf. on Computer Design,* USA, pp. 175–182, 2011. https://doi.org/10.1109/ICCD.2011.6081394.

[21] S. R. Shishira, A. Kandasamy and K. Chandrasekaran, “Workload characterization: Survey of current approaches and research challenges,” in *Int. Conf. on Computer and Communication Technology*, Allahabad, India, pp. 151–156, 2017.

[22] H. Maria Calzarossa, A. Luisa Massari and R. Daniele Tessera, “Workload characterization issues and methodologies,” *Lecture Notes in Computer Science,* pp. 459–482, 2000.

[23] A. Alexander Maksyagin, “Modeling multimedia workloads for embedded system design,” *A dissertation submitted to the Swiss Federal Institute of Technology (ETH) Zurich for the degree of Doctor of Sciences Diss*. ETH No. 16285, 2005.

[24] F. Zhang, J. Cao, W. Tan, S. U. Khan and K. Li *et al.*, "Evolutionary scheduling of dynamic multitasking workloads for big-data analytics in elastic cloud," *IEEE Transactions on Emerging Topics in Computing,* vol. 2, pp. 338–351, 2014.

[25] S. K. Baruah, "Dynamic- and static-priority scheduling of recurring real-time tasks,"*Real-Time Systems,* vol. 24, no. 1, pp. 93–128, 2003.

[26] A. Writam Banerjee, "Challenges and applications of emerging nonvolatile memory devices," *Electronics,* vol. 9, pp. 1029, 2020.

[27] E. Jishen Zhao, D. Cong Xu, C. Ping Chi and H. Yuan Xie, "Memory and storage system design with nonvolatile memory technologies," *IPSJ Transactions on System LSI Design Methodology,* vol. 8, pp. 2–11, 2015.