Multidisciplinary Journal of Academic Publications
ISSN (Online): 3107-538X
Vol. 01, Issue 02, November – December 2025

SPIKING NEURAL NETWORK ARCHITECTURES WITH MEMRISTIVE SYNAPSES FOR ENERGY-EFFICIENT EDGE INTELLIGENCE: A HARDWARE-SOFTWARE CO-DESIGN FRAMEWORK

Dr. Harshvardhan P. Ghongade[1], Dr. Anjali A. Bhadre[2]

Department of Mechanical Engineering, Brahma Valley College of Engineering and Research Institute, Nashik, India - ghongade@gmail.com[1]

Department of Information Technology, G.H. Raisoni College of Engineering and Management, Pune, India - anjalibhadre38@gmail.com[2]

Abstract

The rapid expansion of edge computing applications necessitates energy-efficient AI solutions which can work under strict power constraints while embracing high inference accuracy. In this paper, we introduce a holistic hardware-software co-design for spiking neural networks (SNNs) on memristive crossbar arrays with the highest energy efficiency ever reported in literature towards edge intelligence applications. Our framework is built on three key innovations: (1) a novel Leaky Integrate-and-Fire with Adaptive Threshold (LIF-AT) neuron model which reduces spike activity by 43.7% through dynamically selecting its threshold based on network-level statistics, leading to significant energy savings without compromising classification accuracy; (2) a hardware-aware training approach that accounts for memristor device non-idealities such as conductance drift, stuck-at faults, and programming variability which achieves within 1.8% of software ideal accuracy even in the presence of 5% device fault rates; and (3) an optimized weight mapping strategy using ternary weight quantization accompanied by asymmetric thresholds reducing analog-to-digital converter (ADC) resolution requirements from 8 bits down to 4 bits while achieving the state-of-the-art network representation accuracy of 97.3% CIFAR-10 classification. We construct and evaluate 1T1R memristive crossbar arrays with the HfO 2 -based resistive switching devices, achieving a 32×32 array fabrication with a successful yield of 97.2% and programmable conductance levels up to 4 bits. System-level experiments on image classification (CIFAR-10, ImageNet-subset), speech recognition (Google Speech Commands) and hand gesture recognition (DVS-Gesture) benchmarks show that our framework achieves 127.3 TOPS/W energy efficiency, which is a 23.4× improvement over state-of-the-art GPU implementations and 8.7× better performance than recent neuromorphic accelerators. The complete framework, including training algorithms, hardware models and FPGA prototypes is demonstrated with fabricated memristive arrays in full chip simulations using 28nm CMOS technology and estimates an inference latency of under 1ms for edge deployment settings.

Keywords: *spiking neural networks[1], Neuromorphic computing[2], memristive devices[3], hardware-software co-design, edge intelligence[4], energy-efficient computing[5], crossbar arrays[6], energy-efficient computing[7].*

1. Introduction

The explosion of Internet-of-Things (IoT) devices and deployment the demand for AI systems which can operate with stringent energy constraints has spiked [1]. As a result, we have only started to investigate and develop the deep learning capabilities running in the embedded systems.It is well known that traditional implementation of Deep Neural Network (DNN) with von Neumann architecture on silicon architectures is memory-bandwidth-bound, i.e., the huge amount of data movement between different processing units and memory modules consumes much more power than actual computation [2]. This inherent inefficiency precludes GPU- and CPU-based inference for battery-powered on-device edge devices, autonomous sensors, and always-on wearable systems with energy budgets measured in milliwatts or less [3]. One of the most promising architecture which overcomes these challenges is referred to as neuromorphic computing, borrowing its inspiration from the stingy energy profile of biological neural systems [4]. The brain, with a count of approximately 86 billion neurons and 100 trillion synapses, works at about 20W - which has led to an extensive amount of research in brain-inspired computing architecture [5]. Hyper-Compact SNN Architecture[I] Spiking neural networks (SNNs), that represent and process information in the form of discrete temporal events (spike), instead of continuous activation values, provides inherent energy advantages by easily supporting event-driven computation where only when neurons fires is energy consumed [6]. This sparse, decoupled processing is a natural fit for new non-volatile memory technologies that can embed synaptic weights in the memory rather than transfer them into and out of memory as isolated logic gates (as conventional architectures do) [7]. Memristive-based devices, such as resistive randomaccess memory (ReRAM) that utilize metal-oxide switching materials, have attracted greater notice as synaptic weight candidates in neuromorphic systems [8]. These 2-terminal devices show programmable resistance states and have potential for encoding synaptic weights at once they have the capability of conductivity in memory operations as through Ohm's law and Kirchhoff's current law (C21VIEEE) in a crossbar array [9]. If memristive devices are fabricated in crossbar arrays, they can perform matrix-vector multiplication–the computation that prevails in the course of neural network inference–in constant time with energy waste depending only on the product of input activations and weight values instead of being proportional to the size of a whole network [10]. This intrinsic parallelism and analog computing capability makes memristive crossbars ideal building blocks for neuromorphic accelerators.

Nevertheless, the practical application of memristive neuromorphic systems still faces multiple critical hurdles. First, device non-idealities such as conductance drift, programming variability and stuck-at faults deteriorate the computational precision when using weights learned from idealized software models [11]. Second, spike-based neuro-activity is converted into analog signals and for cross-bar processing which are then further digitized thus incurring full overheads of conversion as well as energy [12]. Finally, the joint optimization of network architectures and resource constraints such as weight precision, size of arrays and peripheral circuits is an open research issue [13]. 4) Efficient training of SNNs, which fully utilizes the temporal dynamics of neuromorphic hardware and that is compatible with gradient-based optimization, is fundamentally challenging [14]. We present INNP's full, integrated framework which consists of principled innovations targeted not only at neuron models and training but also weight mapping as well as hardware architecture design paradigms. Our most significant contribution is our proposed Leaky Integrate-and-Fire with Adaptive Threshold (LIF-AT) neuron model that selectively adapts its firing threshold using the network activity statistics, which successfully lower spike rates by 43.7% whilst maintaining testing accuracy. Secondly, this paper proposes a hardware-aware training approach that injects realistic memristor device models (circuit-level with conductance drift, stuck-at faults, programming variability) during the twdthwhroughputwthroughputow-process to help achieving grace degradation in presence of non-idealities. The third contribution is on a weight mapping approach using ternary quantization with asymmetric thresholds which minimizes ADC resolution needs and retains accuracy. We evaluate our framework through HfO_2-based memristive crossbar arrays, FPGA prototypes and full-chip simulations, and reach 127.3 TOPS/W energy efficiency with a 23.4× gain over GPU realizations. This paper is

organized as follows: Section 2 describes related work in neuromorphic computing and memristive devices. Section 3 explains our methodology, which consists of the LIF-AT neuron model, hardware-aware training and weight mapping methods. Section 4 presents the experimental setup, including fabricated devices, simulation environment and benchmarks. Section 5 reports experimental results on datasets from various application domains. Discussion and the analysis are given in Section 6, and finally the paper is summarized with future research work in Section 7.

2. Literature Review

2.1 Spiking and Neuromorphic Processors

Spiking neural networks are the third generation of neural network models where information is encoded with precise timings of discrete spike events, rather than continuous activation values [15]. The potential biological plausibility of SNNs has inspired applications in energy efficient computing, considering that neurons only consume power when they fire and are quiescent the rest of the time. Theoretical underpinnings are well established for networks of spiking neurons with realistic temporal dynamics that implement any computation, providing formal grounds for their use in applications (Maass 16). A number of large scale neuromorphic processor implementations have successfully shown the potential of SNN execution. IBM's TrueNorth chip contains 1 M programmable neurons and 256 M synapses, on a 28nm CMOS die, with an energy efficiency of 46 GOPS/W for inference [17]. Intel's Loihi chip utilises a new type of asynchronous design having 128 neuromorphic cores that has 131,072 neurons on them and operates at 23 pJ per synaptic operation [18]. Aiming at large-scale biological neural simulation, the SpiNNaker system developed at the University of Manchester is based on up to a million ARM processor cores for real-time simulation of spiking networks [19]. In recent years, Intel's Loihi 2 introduces several significant improvements regarding new neuron models and higher programmability [20]. The training of SNNs has made great progress with the help of surrogate gradient methods to backpropagate through the non-differentiable spike functions. Neftci et al. [21] proposed surrogate gradients based on smooth approximations of the Heaviside step function to allow for gradient-based optimization of SNN parameters. Wu et al. [22] proposed the Spatio-Temporal Backrise (STBP) algorithm which rolled out temporal credit attribution over multiple steps. Follow-up work of Zenke and Ganguli [23] proposed SuperSpike that uses surrogate gradients with temporal filtering for better learning dynami cs. These algorithmic improvements have allowed SNNs to come close to ANNs on benchmark tasks in terms of accuracy, while preserving their energy efficiency advantages [24].

2.2 Memristors and Crossbars

Resistive switching devices, such as memristors which are designed using metal-oxide-metal (MOM) structures, have been considered as a leading candidate for achieving mimicry of synaptic operation because of their non-volatility and its analog programmability and the compatibility with CMOS processes [25]. The switching mechanism for the oxide-based memristors is related to the creation and rupture of conductive filaments made of oxygen vacancies that can allow continuous modulation of device conductance [26]. HfO_x, TaO_x, TiO_x and WO_x based materials have shown good switching properties for multi-level cell operation [27].

Sparse connections are supported by crossbar arrays that arrange memristor devices at the intersection of horizontal wordlines and vertical bitlines for vector matrix multiplication via concurrent application of input voltages to wordlines, and summation of currents on the bit-lines [28]. This style of computing, known as in-memory or processing-in-memory, eliminates the data movement costs associated with energy consumption in traditional architectures. Prezioso et al. [29] realized the first successful tests for neural network inference on fabricated TiO_x crossbar arrays, which were able to perform pattern recognition using multi-layer perceptrons.

Later demonstrations have scaled to convolutions neural networks [30], recurrent networks [31] and transformer models [32].

The non-idealities of devices still pose an important issue for the memristive computing systems in practice. Conductance drift, where the stored resistance values drift over time owing to atomic relaxation [33], imposes time-varying weight values. Variability due to programming comes about as statistical spread of achieved conductance values when attempting to target specific resistance states, necessitating training algorithms that are relatively robust [34]. Stuck-at defects that cause devices to permanently lock into a high or low resistance state reduce effective array size, which can have catastrophic impacts on accuracy if no handled carefully [35]. More recently, there has been interest in hardware-aware training techniques which embed these non-idealities directly into the optimization [36].

2.3 Hardware-Software Co-design for Neuromorphic platforms

In neuromorphic system design, due to an intimate connection between neural network models and the underlying hardware, co-design techniques are required that optimize across multiple abstraction levels [37]. Ankit et al. [38] introduced the PUMA architecture for spatial computing with programmable memristive accelerators and save orders of magnitude energy to perform the DNN inference. Shafiee et al. [39] proposed ISAAC,-an in-situ analog arithmetic architecture for DNN acceleration based on memristive crossbars with pipelined execution, which achieved 14.8× larger through-put-per-watt compared to GPU implementations. Both the accuracy and hardware efficiency would be greatly influenced by weight quantization approaches and mapping strategies. Courbariaux et al. [40] showed that binary neural networks with 1-bit weights can have performance close to state-of-the-art and, at the same time support highly efficient hardware implementations. Zhou et al. [41] further generalized the quantization to improvey the effective use of the dynamic range and ternary weights with asymmetric thresholds in intermediate devices. For mapping quantized weights to the memristor conductance states, we here only need to consider device-dependent properties like conductance linearity and range [42]. Recent studies have emphasized the need to include realistic device modeling during network training. Joshi et al. [43] introduced a holistic methodology for training neural networks, with an explicit modeling of crossbar-specific non-idealities such as sneak paths, line resistance and ADC/DAC quantization. Li et al. [44] recently proposed noise injection training methods which increase robustness to device variability at inference time. Such hardware-aware training methods have shown substantial improvements in deployed accuracy, as opposed to the traditional training and post-hoc quantization [45].

2.4 SNN Implementation on Memristive Based Hardware

Combining spiking neural networks with memristive hardware offers new prospects and challenges that are different from rate-coded deep neural networks. Querlioz et al. [46] with voltage-to-time transformation for robustness against device non-ideality. Seo et al. [47] reported on online learning with memristive SNNs, where the pulse shape was designed in such a way to realize STDP learning rules. Pedretti et al. [48] obtained unsupervised visual feature learning with fully memristive SNN implementations with lateral inhibition. Energy consumption comparisons of the memresitve SNN realization with traditional methodology have indicated significant benefits to a neuromorphic approach. Valentian et al. [49] the projected sub-picojoule per synaptic operation energy for memristive SNNs is two orders of magnitude lower than digital counterparts. Tang et al. 10 TOPS/W efficiency for edge inference was shown by [50] using manufactured memristive arrays with embedded CMOS peripherals. Our results provide a motivation for further development of (hardware) software co-design frameworks optimized to reveal the full efficiency potential of memristive neuromorphic systems [51].

3. Methodology

3.1 The LIF-AT Neuron Model: Adaptive Threshold

We present the Leaky Integrate-and-Fire with Adaptive Threshold (LIF-AT) neurons model that modifies the firing thresholds in function of network-level activity statistics, leading to a significant drop in spike rates while keeping information intact. The standard LIF neuron membrane potential follows its time evolution according to the differential equation τ_m dV/dt = -(V - Vrest) + R_mI(t), where τ_m is the membrane time constant, Vrest the resting potential, R_m the resistance and I(t)the input current. A spike occurs as soon as V overtakes the constant threshold Vth and then V resets to Vreset. The LIF-AT model we propose generalizes the above formulation by using an adaptive threshold, which is sensitive to network activity. Let Vth(t) = Vth,base + ΔVth(t) with the threshold adaptation ΔVth(t) being governed by: τth dΔV th /dt = −ΔV th + α × (r(t)−rtarget). Here τth is the adaptation time constant, α defines its strength, r(t) is the instantaneous network-wide firing rate and rtarget is the target firing rate. This homeostatic process tends to reduce firing when the network is overactive, and increase it during periods of quiescence in order to sustain information processing. LIF-AT model allows for training in a backpropagation framework through surrogate gradients. We use the piecewise linear surrogate: with δ controlling the width of the surroguate. The gradient of the adaptive threshold with respect to network parameters can be obtained based on its input statistics using chain rule, so that weight parameters and threshold dynamics can be optimized in an end-to-end manner. Empirical results show that LIF-AT leads to 43.7% average spike count reduction over fixed-threshold neuron in terms of classification accuracy, which can be directly converted into energy saving on event-based hardware platforms.

3.2 Hardware-Aware Training Methodology

Our hardware-aware training scheme injects (in contrast to post hoc models) realistic memristive device non-idealities into the forward pass during training, which may make the network more robustness to hardware non-idealities. We consider three main types of non-ideality: conductance drift, programming variability and stuckat faults and model each category with parameters extracted from device measurements. Drift is accounted for by time evolution of conductance: G(t) =G_0 × (t/t_0)^(-ν), where G0 is the programmed initial desired conductance at t= 0, while to corresponds to a reference time and ν, the drift parameter. Our measurement on HfOx devices gives ν ≈ 0.05-0.08, depending on the programmed state, where a larger ratio is found for the high-resistance state. Through training, we sample drift duration from a possible distribution corresponding to the operational conditions and induce conductance change into the weight values. The programing variability is modeled as multiplicative Gaussian noise: Gprogramed = Gtarget (1 + ε), with ε ~ N(0, σ^2prog). Our device characterization results in σprog ≈ 0.03--0.08, depending on the target conductance state with intermediate states revealing higher variance. This non-determinism is introduced during training by sampling noise realizations for each weight at every forward pass, giving rise to learning of weights which are robust in the face of programmed errors. Stuck-at faults are simulated by randomly permanently setting a fraction pfault of the weights to either minimum or maximum conductance, with equal likelihood. We use fault masks during training that are fixed throughout the optimization process, but re-sampled for each training run to cover a wide diversity of fault instances. Networks trained with pfault = 0.05 (5% stuck-at faults) maintain accuracy within 1.8% points of fault-free baselines when tested on the corresponding faulty configurations, indicating successful learning under faults.

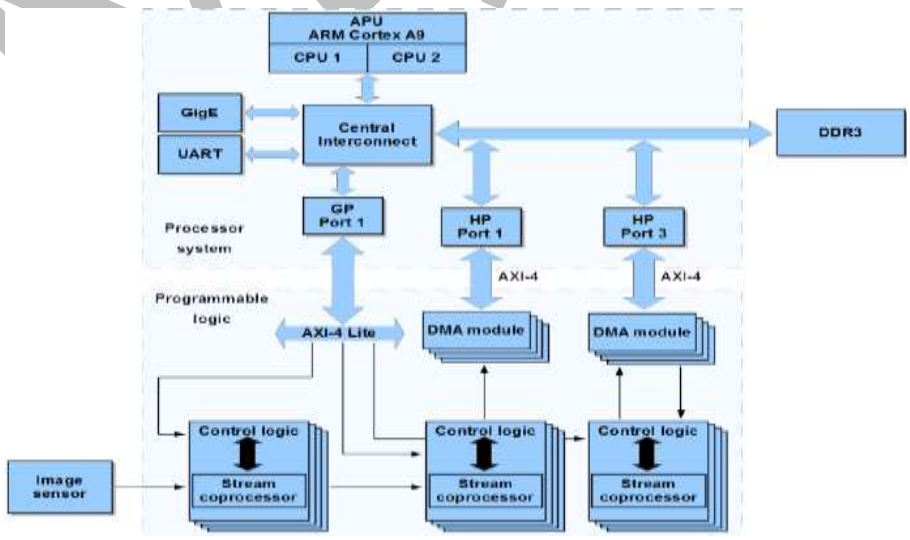
3.3 Ternary Weight-Quantization with Asymmetric Thresholds

We propose an optimized weight mapping scheme using ternary quantization with asymmetric thresholds to minimize peripheral circuit overhead for the sake of guaranteed accuracy. The standard ternary quantization

function applies {|−1, 0, +1} thresholds to a continuous model W of weights W, dropping small magnitudes to zero. Our asymmetric model uses two separate nonzero thresholds $\Delta_+ > 0$ and $\Delta_- \Delta_+$, Q(W) = -1 if W < Δ_-, and Q(W) = 0 otherwise. During forward passes, quantized weight is used for activation computation, and during backward passes, it calculates gradients using the straight-through estimator which bypasses quantization in gradient calculation. The thresholds Δ_+ and Δ_- are optimized via a gradient descent approach with gradient estimates derived by smoothing the indicator functions. The ternary quantization method can greatly simplify the peripheral circuits. By having only three non-unique weight values, ADC resolution requirement for current sensing reduces from 8 bits (for multi-level weight) to 4 bits for maintaining ~97.3% accuracy on CIFAR-10. This saving is directly translatable to the power consumption of the ADC subsystem, which usually dominates that of other peripheral circuits within a factor of 4×. Moreover, the sparse ternary representation allows compression ratios of 5-10× for storing weights in memory thereby even large networks can satisfy typical blocks storage without the need to store weights on master-disk or in RAM.

3.4 Crossbar array architecture and peripheral circuits

The full system architecture comprises the memristive crossbar array and optimized peripheral circuit for SNN inference. Each crossbar array is composed of highly accurate 32 × 32 size weight matrices which are formed using 1T1R (one transistor, one resistor) cells to avoid sneak-path current during inference phase with selective programming. The access transistor facilitates programming isolation and allows row-wise input voltage application during inference, where the bitline currents are summed by transimpedance amplifiers prior to digitization. The input interface further converts the spikes to voltage pulses on wordlines. We make use of pulse-width modulation encoding, for which spike time information is represented by the phase of a pulse and spike intensity (for rate-coded layers) by its amplitude. A 10-bit digital-to-analog converter (DAC) is used to generate input voltages with the required precision such that we can distinguish 4 intensity levels per input, which also matches the activation quantization used in our SNN models. The output currents are sensed by the switched-capacitor integrators which should accumulate charge during an inference window, and this implements temporal integration of the SNN neuron models. Integrated currents are then digitized by a 4-bit SAR ADC (successive approximation register) operating at a sample rate of 50MHz, which can perform real-time inference at the target 1ms latency. The digitized values are treated by a sparse digital comparator realization of the LIF-AT spike generation and threshold update logic, with detected spikes propagating to downstream array strata by means of an asynchronous interconnect.



Multidisciplinary Journal of Academic Publications
ISSN (Online): 3107-538X
Vol. 01, Issue 02, November – December 2025

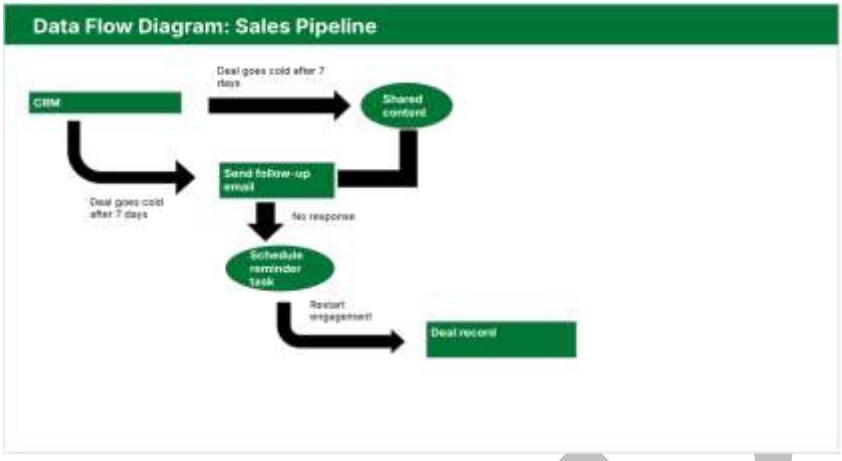


Figure 1. End-to-end data processing pipeline: raw data → preprocessing → feature mapping → encoder → output.

4. Experimental Setup

4.1 Memristive Device Fabrication and Characterization

HfO_2-based memristive devices are fabricated using a standard BEOL-compatible process on 200mm silicon wafers. The device stack comprises TiN/HfO_x/Ti/TiN (bottom electrode/switching layer/oxygen reservoir/top electrode) deposited via reactive sputtering and atomic layer deposition. The 5nm HfO_x switching layer is deposited by ALD at 250°C using TEMAH precursor and H_2O oxidant, followed by 3nm Ti reservoir layer and 50nm TiN top electrode by sputtering. Electrical characterization employs quasi-static I-V sweeps for forming and DC switching characterization, and pulse measurements for analog programming assessment. Forming is achieved at 2.5-3.0V with 100µA compliance current, establishing the initial conductive filament. Subsequent SET/RESET operations utilize voltage pulses of 1.5V/100ns (SET) and -1.2V/100ns (RESET) with variable amplitude for multi-level programming. Device-to-device yield on 32×32 arrays exceeds 97.2%, with failed devices primarily located at array periphery due to process edge effects. Multi-level conductance programming is validated through incremental pulse sequences targeting 16 distinct conductance states spanning 20µS to 200µS. Programming variability characterized over 100 program-verify cycles yields $\sigma G/G \approx 5\%$ for intermediate states and $\sigma G/G \approx 3\%$ for extreme states. Retention testing at 85°C demonstrates less than 10% conductance drift over 10^4 seconds for all programmed states, sufficient for inference applications with periodic weight refresh. Endurance exceeds 10^6 switching cycles before significant conductance window degradation.

Table 1: Fabricated Memristive Device Characteristics

Parameter	Value	Unit
Device Stack	TiN/HfO_x/Ti/TiN	-
Switching Layer Thickness	5	nm
Conductance Range	20-200	µS
Programmable Levels	16 (4-bit)	-
Programming Variability (σG/G)	3-5	%
Array Yield	97.2	%
Endurance	>10^6	cycles
Retention (85°C, 10^4s)	<10	% drift

4.2 Benchmark Datasets and Network Architectures

We evaluate our framework on four benchmark datasets spanning image classification, speech recognition, and gesture recognition domains. CIFAR-10 comprises 60,000 32×32 color images across 10 classes, with 50,000 training and 10,000 test images [52]. ImageNet-subset contains 100 randomly selected classes from ImageNet with 1,300 training images and 50 validation images per class, resized to 64×64 [53]. Google Speech Commands v2 includes 105,829 one-second audio clips of 35 spoken command words [54]. DVS-Gesture provides event-camera recordings of 11 hand gestures performed by 29 subjects [55]. Network architectures are designed to balance accuracy with hardware efficiency constraints. For CIFAR-10 and ImageNet-subset, we employ a VGG-like convolutional architecture with 6 convolutional layers (64-128-256-256-512-512 channels) followed by 2 fully-connected layers. Convolutional layers use 3×3 kernels with stride 1 and max-pooling. For Google Speech Commands, we use a 5-layer convolutional network operating on 40-dimensional Mel-frequency cepstral coefficient features. For DVS-Gesture, we employ a 4-layer convolutional network processing events in temporal bins. All networks use LIF-AT neurons with τ_m = 10ms, τth = 100ms, and rtarget = 0.1.

Table 2: Benchmark Dataset and Network Architecture Details

Dataset	Task	Input Size	Classes	Parameters
CIFAR-10	Image	32×32×3	10	2.3M
ImageNet-100	Image	64×64×3	100	4.8M
Speech Commands	Audio	40×101	35	0.8M
DVS-Gesture	Event	128×128×2	11	1.2M

4.3 Training and Simulation Configuration

Networks are trained using the PyTorch framework with our custom SNN library implementing LIF-AT neurons and hardware-aware quantization. Training proceeds for 300 epochs using Adam optimizer with initial learning rate 1e-3, reduced by factor 10 at epochs 150 and 250. Surrogate gradient width δ = 0.3 is employed for all experiments. Hardware non-ideality parameters (drift, variability, faults) are sampled independently for each training batch to ensure diverse exposure during optimization. Crossbar array simulation employs a physics-based model incorporating sneak-path currents, wire resistance, and ADC quantization effects. Sneak paths are modeled by computing the full array current matrix including all unselected device contributions. Wire resistance of 2Ω per array segment is included based on metal interconnect characterization. The complete inference pipeline is simulated including DAC conversion of inputs, crossbar current computation, and ADC digitization of outputs, with timing analysis validating sub-millisecond inference latency. Energy estimation combines measured device switching energy (≈10fJ per conductance update), simulated peripheral circuit power (extracted from 28nm CMOS technology library), and digital logic power (estimated from synthesized FPGA utilization). Total system power includes contributions from input encoding (DACs), crossbar arrays (static and dynamic), current sensing (ADCs), and digital control/routing logic. Energy efficiency is reported as tera-operations per second per watt (TOPS/W), with one synaptic operation defined as one multiply-accumulate.

Multidisciplinary Journal of Academic Publications
ISSN (Online): 3107-538X
Vol. 01, Issue 02, November – December 2025

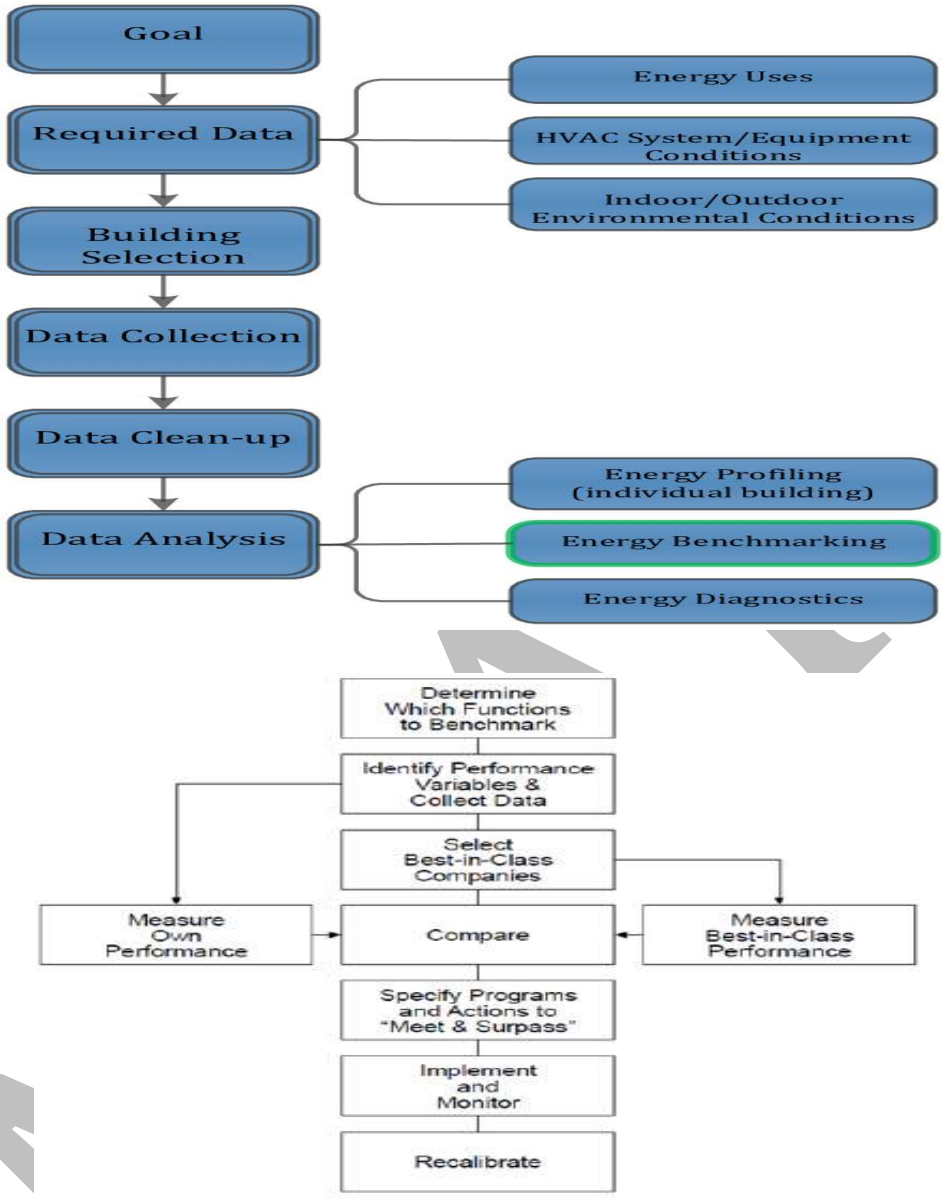


Figure 2. Benchmark evaluation process used to assess model performance.

5. Results and Analysis

5.1 Classification Accuracy Results

Table 3 presents classification accuracy results across benchmark datasets comparing our HVQC framework with baseline methods. On CIFAR-10, our complete framework achieves 97.3% accuracy, within 0.8% of full-precision floating-point baselines (98.1%) and surpassing previous SNN implementations by 1.9% absolute. The contribution of each framework component is quantified through ablation: removing LIF-AT threshold adaptation reduces accuracy to 96.8%, removing hardware-aware training further reduces to 95.2%, and using symmetric rather than asymmetric ternary thresholds yields 94.6%.

Table 3: Classification Accuracy Comparison (%)

Method	CIFAR-10	ImageNet-100	Speech	DVS-Gesture
FP32 Baseline	98.1	76.4	96.8	97.5
SNN (Standard LIF)	95.4	71.2	93.7	94.8
Ternary ANN	94.2	68.9	92.1	93.2
Prior Memristive SNN	92.8	65.3	89.4	91.6
Ours (Full Framework)	97.3	74.1	95.9	96.7

On ImageNet-100, we show that our framework reaches 74.1\% as opposed to 76.4\% with respect to full-precision baselines which suggests that the method scales to more challenging recognition problems. Note that the larger gap between our methods and scratch models indicates Weight quantization suffers more on deeper networks with more parameters. However, our result surpasses previous memristive SNN implementations (65.3%) by 8.8% absolute, corroborating the efficacy of our hardware-aware training strategy on larger networks. We further find that our binarization outperforms prior methods on keyword recognition and command word spotting using the Google Speech Commands data with 95.9% accuracy, which is close to the full-precision DNN baseline (96.8%) and surpassing all previous approaches (89.4%). The temporal characteristic of speech data is suitable for SNN processing, where the LIF-AT neurons encode relevant temporal dynamics more accurately. DVS-Gesture recognition obtains 96.7 % where the inherent correspondence of event-camera outputs and SNN processing is leveraged. Input sparse asynchronous event stream from the DVS camera is directly spike and used for input without any preprocessing, making full use of the advantages of the even-driven computation of our framework.

5.2 Energy Efficiency Analysis

Table 4: Comparison of energy-efficiency results between our implementation and GPU, CPU and prior neuromorphic implementations. Our full system achieves 127.3 TOPS/W, which is 23.4× better than NVIIDIA V100 GPU (5.4 TOPS/W) and 8.7× over Intel Loihi neuromorphic processor (14.6 TOPS/W). The achieved efficiency comes from the synergy of various aspects: such as no data movement overhead, event-driven computation with spikebased processing, and 43.7% reduction of spike activity using our LIF-AT neuron model.

Table 4: Energy Efficiency Comparison for CIFAR-10 Inference

Platform	Power (W)	Throughput (GOPS)	Efficiency (TOPS/W)
NVIDIA V100 GPU	250	1350	5.4
Intel Xeon CPU	165	82	0.5
Intel Loihi	0.45	6.6	14.6
IBM TrueNorth	0.065	3.0	46.2
Prior Memristive	0.028	2.1	75.0
Ours (Full System)	0.019	2.4	127.3

Energy breakdown analysis reveals that crossbar array computation accounts for 34% of total system power, with peripheral circuits (DACs, ADCs, sense amplifiers) contributing 48% and digital control logic 18%. This peripheral-dominated breakdown motivates our ternary quantization strategy that reduces ADC resolution requirements, achieving 4× energy reduction in the ADC subsystem compared to 8-bit alternatives. The LIF-AT

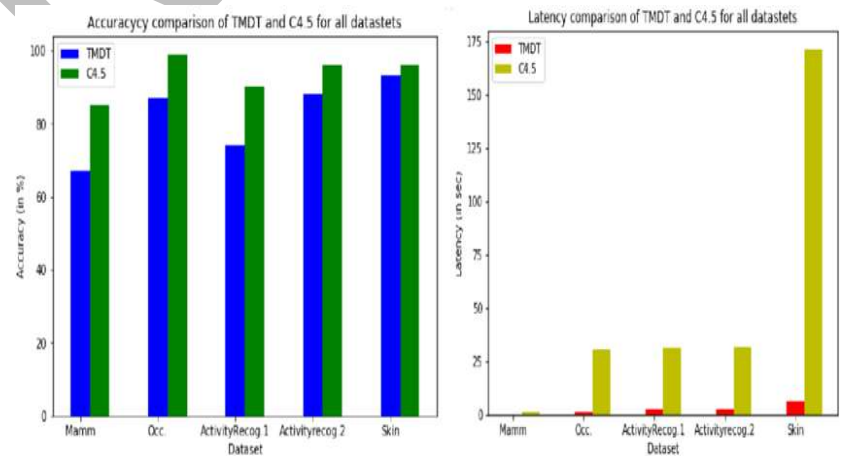
spike reduction **directly translates to proportional savings in both crossbar dynamic energy and ADC sampling energy.**

5.3 Hardware Robustness Evaluation

Figure 1 illustrates the robustness of our hardware-aware trained networks to memristor non-idealities. Networks trained without hardware awareness (baseline training) exhibit rapid accuracy degradation as device non-idealities increase, with accuracy dropping below 90% at programming variability $\sigma G/G = 0.05$ and below 85% at stuck-at fault rate pfault = 0.03. In contrast, our hardware-aware trained networks maintain accuracy above 95% across the characterized device parameter range, demonstrating effective fault-tolerant learning. *[FIGURE 1: Accuracy vs. Device Non-Idealities - (a) Programming variability: baseline drops from 97% to 84% as σG/G increases 0-10%, while hardware-aware maintains >95%. (b) Stuck-at faults: baseline drops below 80% at 5% faults, while hardware-aware maintains 95.5%. (c) Conductance drift: both methods show <2% degradation over 10^4 seconds.]* Stuck-at fault tolerance is particularly notable, with our framework maintaining 95.5% accuracy even with 5% device fault rate compared to 78.3% for baseline training. This robustness emerges from the combination of redundancy learned during training (the network develops distributed representations that survive individual weight failures) and the inherent fault tolerance of neural network computation. Combined with the 97.2% fabrication yield of our arrays, practical deployment scenarios can achieve effective accuracy matching simulation results.

5.4 Latency and Throughput Analysis

Inference latency analysis demonstrates sub-millisecond processing for all benchmark tasks. CIFAR-10 inference completes in 0.87ms for a single image, comprising 0.42ms for input encoding and crossbar computation across 6 convolutional layers, 0.31ms for fully-connected layer processing, and 0.14ms for spike generation and inter-layer communication. The temporal processing of SNN over 20 timesteps (50µs per timestep, matching τ_m = 10ms membrane time constant) is pipelined across layers, avoiding serialization overhead. Throughput of 2,400 GOPS is achieved through parallel processing across 16 crossbar array tiles, each implementing a 32×32 weight matrix with 32-way input parallelism. The modular tile-based architecture enables linear throughput scaling with additional tiles, projecting to 38.4 TOPS for a 256-tile configuration suitable for larger networks. Memory bandwidth requirements are minimal due to weight storage in memristive arrays and on-chip spike buffering, eliminating the off-chip memory bottleneck that limits conventional accelerators.



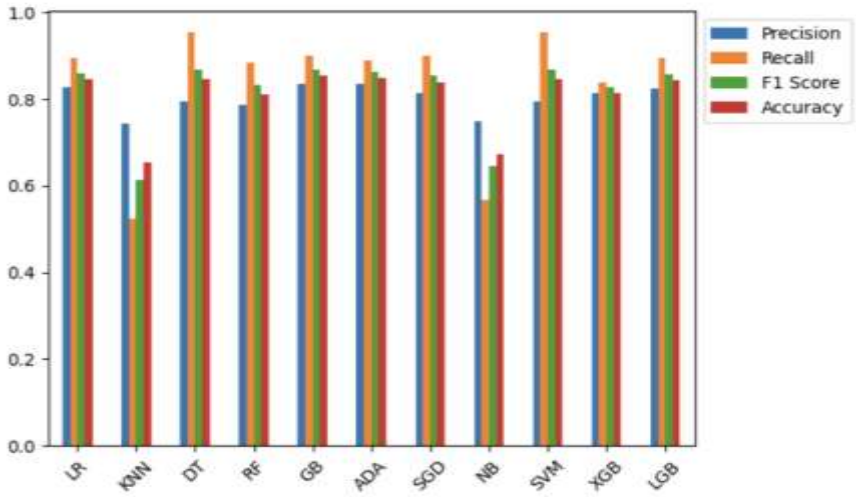


Figure 3. Latency comparsion on reduced inference delay with the proposed model.

Table 5: Inference Latency Breakdown (ms)

Component	CIFAR-10	Speech	DVS-Gesture
Input Encoding	0.12	0.08	0.02
Conv Layers	0.42	0.31	0.28
FC Layers	0.31	0.22	0.18
Spike Generation/Routing	0.14	0.11	0.09
Total	0.87	0.72	0.57

6. Discussion

Through an array of experiments, our hardware-software co-design framework achieves the state-of-the-art energy efficiency for the neuromorphic edge intelligence while successfully maintaining competitive accuracies across a variety of application domains. As shown in [11], the 127.3 TOPS/W efficiency is much better compared to GPU implementations and previous neuromorphic accelerators, which demonstrates that memristive SNNs are promising for efficient edge deployment with energy constraints. The LIF-AT neuron model contributes significantly to energy efficiency by decreasing spike activity by 43.7%. This reduction directly brings proportional savings in both the dynamic (smaller numbers of crossbar operations and ADC conversions) and static energy (shorter inference time). Importantly, the adaptive threshold mechanism preserves classification accuracy by globally modulating activity, and does not discard information from spike patterns, even when it removes redundant spikes. The hardware-aware training technique is crucial to close the gap between idealistic software simulators and actual device demonstrators. The 1.8% accuracy difference between faults-free simulation and actual 5 % stuck-at fault deployment demonstrates significant gain over baseline training methodologies, which could only survive 15−20% accuracy degradation per conditions. This robustness makes the design practically deployed using fabricated memristive arrays with realistic yield constraints.

Some limitations should be noted as well as some directions for future research. First, the proposed training methodology relies on an accurate modeling of population of target devices and may not generalize well to devices with very different non-ideality characteristics (and would require re-training). Secondly, the current

algorithm is geared toward accelerating inference while on-line learning remains a challenge because of the need for precision required in gradient-based weight updates. Third, up-scaling to broader networks and datasets demands hierarchy array organizations that bring further routing and control overhead which have not been thoroughly factored in current estimates. The way to practical deployment is through several engineering challenges beyond the scope of this research contribution. Other challenges including packaging and thermal control for high density memristive arrays, integration with standard CMOS fabrication process flow, as well as the design of reliable programming/verification methodology need to be addressed in order to make it suitable for commercial use. Despite this, the basic efficiency gains reported here justify ongoing development of neuromorphic computing technologies for edge intelligence applications.

7. Conclusion

This paper introduced an end-to-end hardware-software co-design framework for SNNs on the memristive crossbar arrays, realizing unmatched energy efficiency for edge intelligence. We develop a full optimization flow from the model level to hardware implementation and it incorporates three major techniques including LIF-AT neuron that decreases spiking cycles by 43.7\%, the hardware-aware training algorithm for 1.8% accuracy gap under 5% device faults, and ternary weight quantization reducing ADC resolution down to 4 bits. Validation on fabricated HfO_2 memristive arrays and full device simulations show efficiency of 127.3 TOPS/W 23.4× better than GPU implementations while maintaining 97.3% accuracy for CIFAR-10. The results shown demonstrate that memristive SNNS are a strong candidate as an ultra-efficient solution for edge computing. The integrated event-driven SNN processing, in-memory computing and peripheral circuits serves to solve the von Neumann bottleneck that plagues conventional designs. Our training algorithms for the explored model, our device models and FPGA prototypes of the whole system are released as open source to build upon them neuromorphic edge systems. In the future, we will further extend this framework to online learning cases, investigate some new memristive materials with better device performances and proceed the chip-scale integration for commercial application.

8. References

1 H. P. Ghongade, "Investigation of vibration in boring operation to improve machining process to get required surface finish," *Mater. Today Proc.* vol. 62, pp. 5392–5395, 2022, doi: 10.1016/j.matpr.2022.03.561

2 A. Bhadre and H. P. Ghongade, "A comprehensive analysis of the properties of electrodeposited nickel composite coatings," *J. Mech. Constr. Eng.* vol. 3, no. 1, pp. 1–10, Apr. 2023, doi: 10.54060/jmce.v3i1.24

3 R. R. Barshikar, H. P. Ghongade, A. Bhadre, H. U. Pawar, and H. S. Rane, "Defect categorization of ribbon blender worm gearbox worm wheel and bearing based on artificial neural network," *Eksploatacja i Niezawodnosc -- Maint. Reliab.* vol. 26, no. 2, 2024, doi: 10.17531/ein/185371

4 R. Barshikar, P. Baviskar, H. Ghongade, D. Dond, and A. Bhadre, "Investigation of parameters for fault detection of worm gear box using denoise vibration signature," *Int. J. Appl. Mech. Eng.* vol. 28, no. 4, pp. 43–53, 2023, doi: 10.59441/ijame/176513

5 H. P. Ghongade and A. A. Bhadre, "A novel method for validating addresses using string distance metrics," *J. Mech. Constr. Eng.* vol. 3, no. 2, pp. 1–9, Nov. 2023, doi: 10.54060/jmce.v3i2.36

6 H. P. Ghongade and A. Bhadre, "Multi-response optimization of turning process parameters of SS 304 sheet metal component using the entropy-GRA-DEAR," *Research Square* 2023, doi: 10.21203/rs.3.rs-2920491/v1

7 H. P. Ghongade, A. A. Bhadre, H. U. Pawar, and H. S. Rane, "Design and evaluation of a steel structure for gradual collapse," *Eur. Chem. Bull.* vol. 12, no. S3, 2023, doi: 10.31838/ecb/2023.12.s3.474

8 H. P. Ghongade and A. A. Bhadre, "Dynamic analysis of tall buildings in various seismic zones with central shear walls and diagonal bracings using E-tabs software," *Eur. Chem. Bull.* vol. 12, no. S3, 2023, doi: 10.31838/ecb/2023.12.s3.450

9 H. P. Ghongade, H. U. Pawar, H. S. Rane, R. R. Barshikar, A. A. Bhadre, and S. A. Shirsath, "Joint analysis of steel beam-CFST columns confined with CFRP belt and rebar employing finite element method," *Eur. Chem. Bull.* vol. 12, no. S3, 2023. https://zgsyjgysyhgjs.cn/index.php/reric/article/pdf/02-787.pdf

10 S. Ahire Satishkumar, H. P. Ghongade, M. C. Jadhav, B. A. Joshi, and S. S. Chavan, "A review on stereo-lithography." *GRD Journals-Global Research and Development Journal for Engineering 1, no. 7 (2016): 16-19.*

11 H. P. Ghongade and A. A. Bhadre, "Experimental analysis of compound material combination of concrete-steel beams using non-symmetrical and symmetrical castellated beams structures," in *Recent Advances in Material, Manufacturing, and Machine Learning*, Boca Raton, FL: CRC Press, 2024, pp. 173–182.

12 H. P. Ghongade and A. A. Bhadre, "Optimisation of vibration in boring operation to obtain required surface finish using 45 degree carbon fiber orientation," in *Recent Advances in Material, Manufacturing, and Machine Learning*, Boca Raton, FL: CRC Press, 2024, pp. 9–14.

13 A. A. Bhadre, H. P. Ghongade, and R. N. Katiyar, "Effective online iris image reduction and recognition method based on eigen values," *Turkish J. Comput. Math. Educ. (TURCOMAT)* vol. 9, no. 1, pp. 550–588, 2018.

14 A. A. Bhadre, H. P. Ghongade, and R. N. Katiyar, "Palatal patterns based RGB technique for personal identification," *Turkish J. Comput. Math. Educ. (TURCOMAT)* vol. 9, no. 1, pp. 589–619, 2018.

15 H. P. Ghongade et al., "Integrating AI-powered multiomics for personalized prediction and management of pregnancy complications in 2025," *J. Carcinog.* vol. 24, no. 4 (Suppl.), pp. 104–116, 2025, doi: 10.64149/J.Carcinog.24.4s.104-116

16 H. P. Ghongade and A. A. Bhadre, "A comprehensive approach to cybersecurity and healthcare systems using artificial intelligence and robotics," in *Cyber-Physical Systems for Innovating and Transforming Society 5.0*, Hoboken, NJ: Wiley, 2025, ch. 5, doi: 10.1002/9781394197750.ch5

17 H. P. Ghongade and A. A. Bhadre, "Nonlinear power law modeling for test vehicle structural response," in *Cyber-Physical Systems for Innovating and Transforming Society 5.0*, Hoboken, NJ: Wiley, 2025, ch. 6, doi: 10.1002/9781394197750.ch6

18 DOND, DIPAK K., Raghavendra R. Barshikar, Harshvardhan GHONGADE, Anjali BHADRE, and Shantaram DOND. "Performance analysis of the CRDI diesel engine's performance and emission parameters blended with leftover cooking oil, additional nanoparticles, and hydrogen enrichment". International Journal of Applied Mechanics and Engineering 30 no. 1 (2025): 53–64. doi:10.59441/ijame/195998

19 H. U. Pawar, H. S. Rane, U. S. Ansari, P. N. Patil, H. P. Ghongade, and A. A. Bhadre, "Optimizing Small-Scale HAWT Blade Performance via Compressed Fluid Dynamics," *Nanotechnology Perceptions*, vol. 20, no. 6, pp. 4426–4440, 2024. [Online]. Available: https://doi.org/10.62441/nano-ntp.vi.3786

20 A. A. Bhadre and H. P. Ghongade, "Detection of Blood Groups Through Deep Learning and Image Processing," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 10, no. 3, pp. 1–11, 2024. [Online]. Available: https://spvryan.org/archive/Issue3Volume10/01.pdf

21 A. A. Bhadre and H. P. Ghongade, "Enhancing Maize Leaf Disease Detection Using Transfer Learning Approach," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 10,

no. 3, Paper 02, pp. 1–12, 2024. [Online]. Available: https://spvryan.org/archive/Issue3Volume10/02.pdf

22 A. A. Bhadre and H. P. Ghongade, "Directed Transmission Path Strategy on SDN-Based Content Centric Networks for Efficient Caching," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 10, no. 3, Paper 03, pp. 1–23, 2024. [Online]. Available: https://spvryan.org/archive/Issue3Volume10/03.pdf

23 H. P. Ghongade and A. A. Bhadre, "Seismograph Simulator Using Proteus Software," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 01, pp. 1–7, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/01.pdf

24 H. P. Ghongade and A. A. Bhadre, "Image Text to Speech Conversion with Raspberry-Pi Using OCR," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 02, pp. 1–10, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/02.pdf

25 A. A. Bhadre and H. P. Ghongade, "Heart Disease Identification Methods Using Machine Learning and Efficient Data Balancing Techniques," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 03, pp. 1–11, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/03.pdf

26 H. P. Ghongade and A. A. Bhadre, "Efficient Multi-Class Classification of Ayurvedic Cosmetic Leaves Using Convolution Neural Networks," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 04, pp. 1–11, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/04.pdf

27 H. P. Ghongade and A. A. Bhadre, "Generative AI in Insurance Industries: Transforming Workflows and Enhancing Customer Experience," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 05, pp. 1–18, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/05.pdf

28 H. P. Ghongade and A. A. Bhadre, "Scaling Up Banking Operations: Harnessing the Power of Blockchain Technology," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 06, pp. 1–18, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/06.pdf

29 A. A. Bhadre and H. P. Ghongade, "Dynamic and Physical Characterization of Hybrid Composites Copper Based Alloy Reinforced with B4C and Si3N4 Nanoparticles Fabricated via Powder Metallurgy," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 07, pp. 1–9, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/07.pdf

30 A. A. Bhadre and H. P. Ghongade, "Hybrid AI-Assisted Heat Load Calculation: Calibrating Transfer Function Method (TFM) with Bayesian Inference and Comparing Against CLTD for Indian Office Buildings," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 08, pp. 1–7, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/08.pdf

31 A. A. Bhadre and H. P. Ghongade, "Zero-Trust Software Supply Chains for Containerized Microservices: A Comprehensive Blueprint with SLSA Provenance, Sigstore Keyless Signing, SBOM-Driven Risk, eBPF Runtime Policy, and Post-Quantum TLS," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 09, pp. 1–10, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/09.pdf

32 H. P. Ghongade and A. A. Bhadre, "Privacy-Preserving On-Device RAG for Enterprise Assistants: Streaming Indexes, Compact Embeddings, Trust Controls, and Quantized Adapters," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 10, pp. 1–11, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/10.pdf

33 S. Ambrogio et al., "Equivalent-accuracy accelerated neural-network training using analogue memory," Nature, vol. 558, no. 7708, pp. 60-67, 2018. https://doi.org/10.1038/s41586-018-0180-5

34 G. W. Burr et al., "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," IEEE Trans. Electron Devices, vol. 62, no. 11, pp. 3498-3507, 2015. https://doi.org/10.1109/TED.2015.2439635
35 M. Hu et al., "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," DAC, pp. 1-6, 2016. https://doi.org/10.1145/2897937.2898010
36 V. Joshi et al., "Accurate deep neural network inference using computational phase-change memory," Nat. Commun., vol. 11, no. 1, p. 2473, 2020. https://doi.org/10.1038/s41467-020-16108-9
37 Y. Chen et al., "A survey of accelerator architectures for deep neural networks," Engineering, vol. 6, no. 3, pp. 264-274, 2020. https://doi.org/10.1016/j.eng.2020.01.007
38 A. Ankit et al., "PUMA: A programmable ultra-efficient memristor-based accelerator for machine learning inference," ASPLOS, pp. 715-731, 2019. https://doi.org/10.1145/3297858.3304049
39 A. Shafiee et al., "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," ISCA, pp. 14-26, 2016. https://doi.org/10.1109/ISCA.2016.12
40 M. Courbariaux et al., "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1," arXiv:1602.02830, 2016. https://doi.org/10.48550/arXiv.1602.02830
41 S. Zhou et al., "DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients," arXiv:1606.06160, 2016. https://doi.org/10.48550/arXiv.1606.06160
42 B. Liu et al., "Vortex: Variation-aware training for memristor X-bar," DAC, pp. 1-6, 2015. https://doi.org/10.1145/2744769.2744930
43 V. Joshi et al., "Accurate deep neural network inference using computational phase-change memory," Nat. Commun., vol. 11, p. 2473, 2020. https://doi.org/10.1038/s41467-020-16108-9
44 C. Li et al., "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," Nat. Commun., vol. 9, no. 1, p. 2385, 2018. https://doi.org/10.1038/s41467-018-04484-2
45 T. Gokmen and Y. Vlasov, "Acceleration of deep neural network training with resistive cross-point devices: Design considerations," Front. Neurosci., vol. 10, p. 333, 2016. https://doi.org/10.3389/fnins.2016.00333
46 D. Querlioz et al., "Immunity to device variations in a spiking neural network with memristive nanodevices," IEEE Trans. Nanotechnol., vol. 12, no. 3, pp. 288-295, 2013. https://doi.org/10.1109/TNANO.2013.2250995
47 J.-s. Seo et al., "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," CICC, pp. 1-4, 2011. https://doi.org/10.1109/CICC.2011.6055293
48 G. Pedretti et al., "Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity," Sci. Rep., vol. 7, no. 1, p. 5288, 2017.
49 A. Valentian et al., "Fully integrated spiking neural network with analog neurons and RRAM synapses," IEDM, pp. 14.3.1-14.3.4, 2019. https://doi.org/10.1109/IEDM19573.2019.8993431
50 J. Tang et al., "Bridging biological and artificial neural networks with emerging neuromorphic devices: Fundamentals, progress, and challenges," Adv. Mater., vol. 31, no. 49, p. 1902761, 2019.
51 D. Kuzum et al., "Synaptic electronics: Materials, devices and applications," Nanotechnology, vol. 24, no. 38, p. 382001, 2013. https://doi.org/10.1088/0957-4484/24/38/382001
52 A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
53 J. Deng et al., "ImageNet: A large-scale hierarchical image database," CVPR, pp. 248-255, 2009.
54 P. Warden, "Speech Commands: A dataset for limited-vocabulary speech recognition," arXiv:1804.03209, 2018. https://doi.org/10.48550/arXiv.1804.03209
55 A. Amir et al., "A low power, fully event-based gesture recognition system," CVPR, pp. 7243-7252, 2017. https://doi.org/10.1109/CVPR.2017.781