I. J. Computer Network and Information Security, **2026, 4, 130-149**
Published Online on August 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2026.04.07

Neuromorphic RISC-V Systems for Bio-inspired Computing Applications

Yamini Devi Ykuntam*
ECE, Jawaharlal Nehru Technological University (JNTU), Kakinada, India
Email: yaminieie@gmail.com
ORCID ID: https://orcid.org/0000-0001-8043-8288
*Corresponding author

M. V. Nageswara Rao
ECE, GMR Institute of Technology, India
Email: nageswararao.mv@gmrit.edu.in
ORCID ID: https://orcid.org/0000-0003-3590-3501

Leela Kumari. B.
ECE, UCEK, JNTUK, India
Email: leela8821@yahoo.com
ORCID ID: https://orcid.org/0000-0002-4551-5863

Received: January 25, 2025; Revised: May 20, 2025; Accepted: September 2, 2025; Published: August 8, 2026

Abstract: Neuromorphic computing is a paradigm based on the computational mechanisms of the human brain and has received considerable attention as a real-time technique with low energy requirements. Present systems, however, are limited in their ability to scale traditional processors to a neuromorphic architecture, leading to issues with latency, power consumption, and smooth data flow. To address these problems, this paper proposes the ACORISC-VbSNN framework, comprising a modular RISC-V architecture, Spiking Neural Networks (SNNs), and Ant Colony Optimization (ACO). The system uses a shared-memory architecture to maximize communication between traditional and neuromorphic processors, ensuring data is managed effectively. The postulated framework processes the sensory data by pre-processing and encoding it using rate coding, and dynamically optimizing memory access. SNNs are also used to process spike trains in real-time, whereas ACO is used to determine the best data paths to minimize bottlenecks. Experimental analysis shows that the system performs better, with ultra-low power consumption of 0.0095 mW, very low latency of 0.000544 s, and 99.2% accuracy. These findings indicate that the ACORISC-VbSNN model has the potential to advance the field of bio-inspired computing, providing a highly accurate, energy-efficient, and low-latency system for real-world use.

Index Terms: Neuromorphic Computing, Latency, Power Efficiency, Spiking Neural Networks, Ant Colony Optimization, RISC-V Architecture.

1. Introduction

Neuromorphic systems are designed to emulate the fundamental computational principles of biological nervous systems, thereby overcoming the limitations of traditional computer architectures, in which the central processing unit (CPU) and memory must continuously exchange data [1]. They can change their connections dynamically by learning and adapting over time. Many operations are handled by neural networks (NNs) at high speed, while conventional networks often process tasks one at a time [2]. In fields like image recognition and language processing, deep neural networks play a significant role in artificial intelligence (AI) research. The majority of these AI systems, referred to as narrow AI, focus on specific tasks and problems [3]. Energy consumption is reduced during computation (which occurs in memory) because there is no need to go and come back repeatedly [4]. Energy-efficient computing is essential in the Internet of Things (IoT), where devices connect to sensors, process data, and transmit it wirelessly. Signal processing algorithms are used to simplify and compress data, thereby improving power consumption, performance, and bandwidth in IoT networks [5]. Deep Learning (DL) has been applied to decode 3-dimensional sEMG signals into instructions, classifying either discrete gestures or continuous variables for more consistent control [6].

As a foundation for neuromorphic computing systems, the RISC-V architecture, with its open-source, highly customizable instruction set, is gaining attention [7]. Unlike proprietary architectures, RISC-V has enabled researchers and developers to develop specialized hardware to realize bio-inspired applications [8]. This flexibility is essential for allowing neuromorphic capabilities, including parallel processing, sparse coding, and energy-efficient computation, in the hardware [9]. Neuromorphic systems can be created using the open-source RISC-V to suit certain applications, such as autonomous robots and wearable biomedical equipment [10], while also improving performance, power efficiency, and scalability [11]. RISC-V is a promising solution for bio-inspired computing applications [12], where energy efficiency and real-time performance are critical [13].

These structures enable the efficient processing of complex sensory information, including vision, sound, and touch, through a mechanism that directly mimics brain-like computation [14]. For example, neuromorphic systems based on RISC-V may be used in autonomous robotics to interpret incoming event data (e.g., from a DVS camera) to detect patterns and provide a decision with minimal latency and power consumption [15]. Similarly, bio-inspired prosthetics or assistive devices that react speedily to neural signals can also be facilitated by such systems in the healthcare setting [16]. As RISC-V is flexible, developers can continue optimizing these systems to suit various bio-inspired applications and avoid the constraints of traditional computing architectures [17]. Relevant works include a RISC-V based fault-tolerant processor [18], AGILER [19], and von Neumann architectures [20].

The study's main objective is to develop a high-performance, energy-efficient neuromorphic computing platform, called ACORISC-VbSNN, that combines the open-source RISC-V architecture with SNNs and ACO for use in bio-inspired computer systems. The suggested system should address the weaknesses of classic von Neumann architectures by facilitating parallel, event-driven processing and effective data communication through a shared common memory structure between traditional and neuromorphic processors. Sensory information is preprocessed and rate-coded into spike trains, followed by dynamic ACO optimization to achieve low latency and low power consumption. This study is novel since it integrates RISC-V, SNN, and ACO in a hybrid manner within a unified architecture that not only enhances communication efficiency between heterogeneous processors but also achieves superior accuracy (99.2%), ultra-low power consumption (0.0095 mW), and minimal latency (0.000544 s). For real-time intelligent systems like robotics, medical equipment, and edge AI applications, this novel framework offers a scalable, bio-inspired approach.

2. Related Works

2.1. Review of Related Work

Dong et al. [21] propose a new processing system for smart homes that mimics the human brain's sensory-motor processing, using memristor circuits to create a low-cost, eco-friendly solution. A memristor crossbar array made of eco-friendly 2D materials reduces system complexity by enabling parallel connections, which lowers system costs. A neuromorphic sensory-processing module that effectively gathers and processes a variety of sensory inputs is used for monitoring. A fully hierarchical, biomimetic learning module further improves smart home features.

Nazari et al. [22] developed a system inspired by the brain to recognize patterns using SNNs, modeled after real brain neurons that send signals (spikes) to one another. The research aims to accurately classify images from popular datasets (MNIST and CIFAR-10) using this hardware system. To make the system faster and less expensive, they reduce the amount of data processed by using Principal Component Analysis (PCA). This helps design the system to optimize performance and reduce delays.

Gillet et al. [23] focus on making fast AI tasks with SNNs more practical for low-power devices. SNNs work best with event-based data from specialized neuromorphic sensors, but these are hard to find. To solve this, regular sensors can be used along with a process called "encoding" to make the data compatible with SNNs. However, doing this in software can slow things down and use more energy. The researchers propose a method for creating and testing efficient hardware systems to handle this encoding process. They use tools (HLS and Python) to design and evaluate different hardware setups quickly. By testing two different encoding methods on FPGAs (a type of hardware), they measure energy usage and speed. This approach could reduce reliance on special sensors and advance low-power SNN hardware.

By offering a unified, ultra-low-power solution for edge AI applications, Rutishauser et al.'s ColibriES platform [24] addresses the challenges of integrating event-based sensors (e.g., DVS cameras) with neuromorphic processors (e.g., Loihi). ColibriES maximizes performance and energy economy by supporting both event-based SNNs and frame-based ternary convolutional neural networks (TCNNs). It is based on the Kraken System-on-Chip (SoC) with a heterogeneous PULP processor. It is ideal for real-time, battery-powered devices like wearables and UAVs, achieving 7.7 mJ per inference and 164.5 ms latency in gesture recognition applications. However, wider adoption and scalability may be hampered by its dependency on specialist technology and narrow evaluation scope.

To achieve flexibility and lower power consumption, Schiavone et al. [25] proposed an eFPGA-Augmented RISC-V SoC for IoT nodes. The RISC-V circuit operates at a power of 46.83 μW/MHz and a voltage between 0.5 and 0.8 V. The platform comprises a customizable FPGA, a flexible microcontroller, and a potent 22nm FinFET CPU. It achieves ultra-low sleep power (20.5 μW) by using body-biasing to reduce the embedded FPGA's leakage power by up to 18×. Compared to comparable reconfigurable SoCs in its class, the SoC exhibits 3.4× greater performance and 2.9× better energy efficiency.

Hassan et al. [26] proposed hyperdimensional computing for robotic grasping perception, which helps determine an object's size and solidity to calculate grasping force. Data preprocessing is complete, which is essential for computing hyperdimensional operations. The accuracy of the suggested method is calculated and compared with that of other current techniques. The efficiency of the implemented design is also calculated. Efficiency and speed are improved by 26× and 10×, respectively, compared to the existing methods. Apart from all these, it effectively reduced environmental noise.

With a focus on near-sensor data analysis, Leone et al. [27] proposed an approach that presents SYNtzulu, an SNN processing element designed for low-cost, low-power FPGA devices [28]. To increase the system's versatility, it incorporates a RISC-V subsystem that manages input/output functions and sets runtime settings. Using a Lattice iCE40UP5K FPGA, SYNtzulu is tested in a variety of scenarios and shows accuracy levels on par with the most advanced models. In terms of power efficiency, dynamic power management strategies can reduce the system's maximum consumption during SNN inference from 12.05 mW to an average of 1.45 mW. Because this method guarantees low power consumption, it can be used in edge AI applications that require energy-efficient operation.

Table 1 details existing work challenges. It includes the methods used in the work, along with their advantages and disadvantages.

Table 1. Problems with the previous approach.

S.No	Author Name	Technique	Advantage	Disadvantage
1	Dong, et al [21]	Multimodal Neuromorphic for smart homes	The performance accuracy is improved.	Need to improve the fabrication process.
2	Nazari, et al [22]	Spike-Timing Dependent Plasticity (STDP)-based Spiking Neural Networks	The power consumption is very low, with a slow rate of convergence.	Need to reduce design cost.
3	Gillet, et al [23]	Digital Spike Encoding Architecture Generation	It is very flexible in optimising design.	With increased throughput, power consumption has increased slightly.
4	Rutishauser, et al. [24]	Embedded system with Neural Network	It showed reduced latency.	Need to improve the design's performance.
5	Schiavone, et al [25]	eFPGA-Augmented RISC-V SoC	Offers high versatility and energy efficiency in a compact, low-power IoT device.	Its limited size may restrict the complexity of hardware accelerators.
6	Hassan, et al [26]	Hyperdimensional computing of robotic grasping perception	It has good processing speed and improved efficiency.	Methods to reduce noise should be included.
7	Leone, et al [27]	Processor of SNN with tiny RISC-V	It is very flexible and efficient.	Need to improve energy efficiency.

Based on the literature reviewed, there has been significant progress in combining neuromorphic processing with energy-efficient architectures. Nonetheless, the vast majority of available systems involve trade-offs among quality, response time, and power usage. For example, Dong et al. [21] and Nazari et al. [22] achieved better pattern recognition and lower power consumption, but at the cost of increased design complexity and slower convergence. Low-power SNN hardware was developed by Gillet et al. [23] and Rutishauser et al. [24], but, due to specialized sensors and poor optimization, their designs made adaptability less attainable. In the same vein, Schiavone et al. [25] and Leone et al. [27] have shown flexible RISC-V-based SoCs for IoT and edge applications, but did not provide efficient co-optimization of neural and processing units. In contrast, the suggested ACORISC-VbSNN system is the only one that unifies the RISC-V architecture, Spiking Neural Networks, and Ant Colony Optimization, achieving the highest accuracy (99.2%), lowest power consumption (0.0095 mW), and lowest latency (0.000544 s). In this hybrid design, the shortcomings of previous designs have been addressed, as it provides optimal data flow, real-time responsiveness, and scalable, energy-efficient performance capabilities of bio-inspired and neuromorphic designs.

2.2. Among the major contributions of the proposed method, the following should be mentioned

- Sensors capture visual, auditory, or tactile inputs, enabling the system to receive sensory data.
- A high-level architecture integrates neuromorphic components (like spiking neurons) with the RISC-V core processor.
- Data preprocessing is included to remove unwanted data or redundant information from the sensory image.
- Encoding is applied to this data to convert it into spike trains suitable for the SNN using rate coding.
- Neurons are activated through SNN and are updated by adjusting synaptic weights based on the incoming spikes. This enables the SNN to learn over time and adapt to new information.
- ACORISC-VbSNN Proposed method: Generates a shared memory space, where both the RISC-V core and the SNN co-processor can access the same memory. The ACO assists in the efficient exchange of information between traditional and neuromorphic processors to maintain data flow.
- Decoding converts the obtained spike train, which is then used to perform the required tasks.
- Finally, the performance is evaluated in terms of speed, power efficiency, accuracy, and latency.

3. Motivation and Problem Statement

The accelerated development of intelligent systems, such as autonomous robots, IoT systems, and edge AI applications, requires energy-efficient, real-time computing architectures that replicate the information-processing nature of the brain. Traditional von Neumann systems have bottlenecks in data transfer, consume large amounts of power, and are not very adaptable for processing bio-inspired sensory data. Despite these achievements, neuromorphic computing often lacks coordination, optimization, and scalability, even as both RISC-V architectures and SNNs have advanced separately. Current solutions cannot be used in dynamic real-world settings, as they cannot balance accuracy, latency, and power efficiency. To address these challenges, this work proposes the ACORISC-VbSNN model, which incorporates a RISC-V architecture, SNN processing, and ACO to enable optimal data flow, low energy consumption, and real-time decision-making in next-generation bio-inspired computing systems.

4. Proposed Methodology

The proposed ACORISC-VbSNN method combines the ACO algorithm with a shared-memory design that allows the RISC-V core (traditional processor) and the SNN co-processor (neuromorphic processor) to communicate via the same memory space. This shared-memory model enables seamless, effective communication between the classical computing tasks performed by the RISC-V core and the neuromorphic computing tasks performed by the SNN. ACO also plays a vital role in this system by dynamically determining the most efficient routes for transferring information between processors, much as ants determine the most efficient route to food sources. ACO assists in minimizing bottlenecks and improving overall system performance. The Proposed Architecture diagram is given in Figure 1.

The proposed ACORISC-VbSNN methodology starts with sensor data acquisition. The raw sensory information is then forwarded through the preprocessing step, which includes cleaning and formatting to make it compatible with the neuromorphic system. Image noise removal methods, such as Gaussian filters, and audio noise removal methods, such as band-pass filters, are used, and normalization is then applied to standardize the data. The previously processed data is transformed into spike trains at the encoding stage using rate coding, which encodes the data based on the rate at which neurons fire—the ACORISC-VbSNN stage, which combines the RISC-V processor and SNNs, further processes the encoded spike trains. The Leaky Integrate-and-Fire (LIF) mechanism is modeled in the SNN, which receives and processes spikes by updating the neuron's membrane potential and releasing a spike when the threshold is exceeded. This is done by exploiting the dynamism of neurons to process information. The ACO algorithm is an important element of this stage, as it optimizes information exchange between the RISC-V core and the SNN. As an expression of the collective memory space, ACO dynamically updates pheromone trails based on solution quality, determining the most effective paths for information flow, reducing bottlenecks, and increasing communication efficiency. After the spike trains are decoded, the decoding phase interprets the neural outputs into actionable data. The RISC-V processor then uses the decoded information to perform tasks such as decision-making, control, and feedback.

Lastly, the system is assessed based on key metrics, such as accuracy, power consumption, and latency. The dataset is tested on Neuromorphic-MNIST (N-MNIST), where the accuracy of the tasks, power consumption, and response time are compared with existing approaches such as LSVM and RISC-VbCNN. The overall effect of this combined working process is a highly efficient neuromorphic operating system that processes information in real time, making it well-suited to bio-inspired applications.

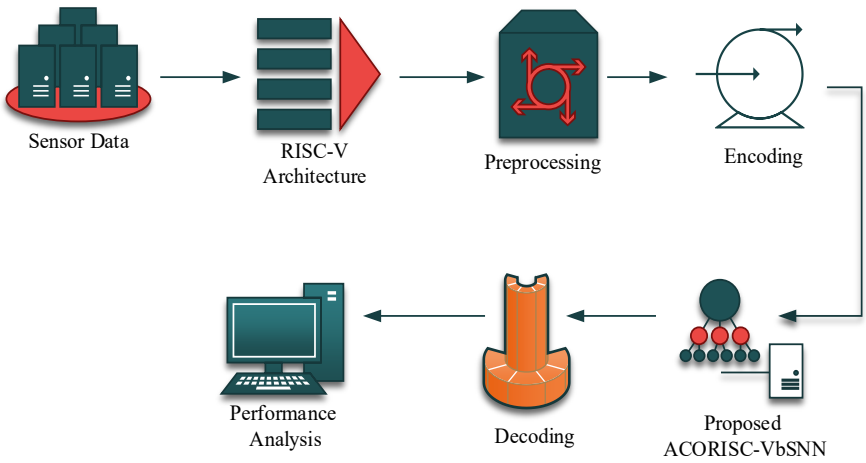

Fig. 1. Proposed architecture.

4.1. Sensor Data

To collect sensory information for bio-inspired work, different types of sensors are used to obtain visual, auditory, and tactile data about the surrounding environment. Image data is gathered with cameras, such as RGB, depth, and infrared cameras, and microphones or audio sensors capture auditory signals. Physical interactions, such as touch, are measured by tactile sensors, e.g., pressure, proximity, or force sensors. The other sensors are thermal, bio-signals (EEG or EMG), and environmental (humidity and light intensity). It is expressed in Equation (1).

$$S = \{s_1, s_2, \ldots s_n\} \tag{1}$$

The acquired information is usually packaged as images (JPEG, PNG), video (MP4, AVI), audio waveforms (WAV), spectrograms, or numeric arrays of tactile measurements.

4.2. RISC-V Architecture

An open-source instruction set architecture (ISA) based on RISC is called RISC-V (Reduced Instruction Set Computer - Five). Researchers at the University of California, Berkeley originally presented it in 2010. Since its release, numerous contributions have been made to develop it. RISC-V was designed for research and education, and today has several industry applications, including the production of commercial chips. Figure 2 shows the RISC-V CPU.

Fig. 2. RISC-V processor.

4.2.1. Key Attributes of RISC-V

- Open-Source Nature: Anyone can use, alter, and implement it for free without paying a license price. Encourages innovation and collaboration across academia and industry.
- Modular Design: A 32-bit, 64-bit, or 128-bit integer base is one of the fundamental instruction sets. It is possible to add additional optional extensions (such as vector processing and floating-point), allowing tailored implementations across different applications.
- Simplicity and Scalability: It emphasizes a simplified set of instructions to be efficient and perform well. It is meant to scale across a wide range of devices, from microcontrollers to high-performance computing systems.
- 4 Versatility: It can be used in IoT, mobile, and data center applications, as well as in AI accelerators. It also facilitates cross-platform compatibility and long-term sustainability.

4.2.2. Instruction Set

The set of instructions, the description of the operations to be performed, and the operands that should be used form the lexicon of a computer. The computer can only understand these instructions because they are encoded in binary machine language. Assembly language, a symbolic representation, is used to improve readability.

- Instruction Encoding: RISC-V instructions follow a simplified RISC philosophy, consisting of 40 unique instructions in the base ISA, RV32I. For ease of use and efficiency, each instruction is encoded as a 32-bit word that is aligned to a 4-byte memory boundary.
- Instruction Format: It includes four primary instruction formats. They are detailed below:

1. R-Type (Register): Three registers are used for these operations: one destination register, rd; two source registers, rs1 and rs2.
2. I-Type (Immediate): These work with a 12- or 13-bit immediate value, one source register (rs1), and one destination register (rd).
3. S-Type (Store): For memory storage operations, it uses a 12- or 13-bit signed immediate and two

registers (rs1, rs2) as sources.

4. U-Type (Upper Immediate): It uses a 20- or 21-bit immediate value and a destination register (rd).

Figure 3 provides the encoding format for instruction types. Every instruction form has the same encoding format, which simplifies decoding. The opcode (7 bits), funct3 (3 bits), and funct7 (7 bits) fields are used to specify operations; registers and immediates are fixed-bit fields.

RISC-V registers are small-capacity, high-speed, and directly-accessible storage locations accessible by instructions. The architecture has 32 registers (x0-x31), which are 32 bits each. Table 2 explains these registers.

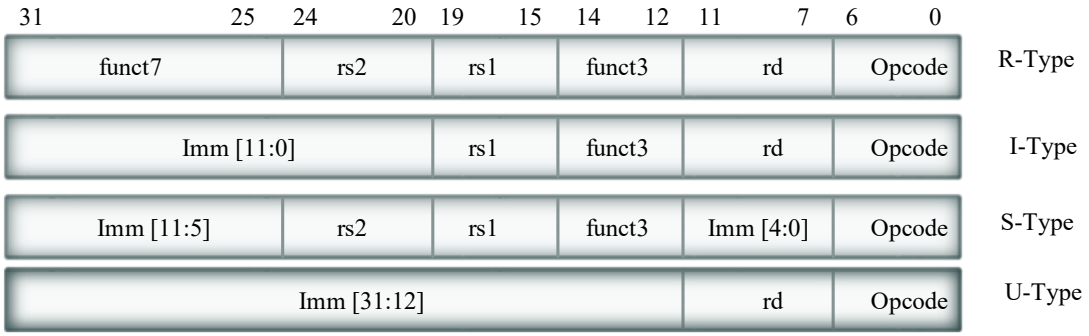

Fig. 3. Encoding of instruction type.

Table 2. Registers in RISC-V.

Register Group	Register Names	Description
Zero Register	x0	Hard-wired zero
Control Registers	x1-x4	Return address, stack pointer, global pointer, thread pointer
Temporary Registers	x5-x7, x28-x31	Temporaries
Saved Registers	x8, x9, x18-x27	Saved registers for function calls and context switching
Argument Registers	x10-x17	Function arguments and return values

The major benefit of this design is its consistency: across all instruction types and in every location, registers appear in the same bit positions; hence, accessing data is faster than accessing memory.

Registers are limited in storage capacity despite being fast. In larger data sets, memory is employed. The RV32I architecture provides 4 GB (2^{32} bytes) of addressable memory, organized as an array of 32-bit words. Although this organization is addressable by bytes, each 32-bit word consists of four 8-bit bytes. The addresses of words are aligned to multiples of 4, which leads to efficient memory access. Because RISC-V instructions can operate only on registers, processing memory-stored data requires transferring it into registers first, after which it can be processed.

As the RISC-V architecture enables efficient calculations, the second task is to feed the sensor data to be encoded and processed.

4.3. Pre-processing

Data pre-processing aims to clean, format, and prepare raw sensory data for encoding into spike trains or further analysis by the neuromorphic system. The process begins with noise removal, where techniques such as Gaussian filters for images, as represented in Equation (2) [29], and band-pass filters for audio are applied to eliminate unwanted noise, and denoising autoencoders can be used for advanced cleaning.

$$X(p,h) = \frac{1}{2\pi\vartheta^2} e^{-\frac{p^2+h^2}{2\vartheta^2}} \tag{2}$$

Gaussian function '$X(p,h)$' over a position '(p,h)' which specifies the filter size is typically selected to be odd (e.g., 3×3, 5×5) to guarantee a central reference point. The standard deviation ϑ controls the Gaussian curve's breadth, or spread: a small ϑ value yields a sharper filter that preserves fine details, while a larger ϑ value results in broader smoothing. The exponential term $e^{-\frac{p^2+h^2}{2\vartheta^2}}$ assigns weights to positions in the kernel, with values decreasing as the distance from the center increases, creating the characteristic Gaussian profile. Lastly, the normalizing factor $\frac{1}{2\pi\vartheta^2}$ ensures that the sum of all kernel weights equals 1, thereby maintaining the data's overall intensity. By carefully selecting the filter size ϑ, the Gaussian filter can effectively balance detail preservation with noise reduction.

Normalization: Once a pixel standard has been resized, the pixels are brought to a given range (0 to 1 or -1 to 1).

In addition to stabilizing the training process, normalization may improve the model's performance by reducing its sensitivity to variations in image brightness and contrast. The formula provided in Equation (3) is used to normalize a typical image with pixel values ranging from 0 to 255 to a 0 to 1 range [30].

$$\hat{a} = \frac{a_n - \min(a)}{\max(a) - \min(a)} \tag{3}$$

Normalization $\hat{a}$ is performed on the pixel a_n values; the minimum and maximum values range from 0 to 255. Once the sensory data has been pre-processed and refined, encoding this data into a format appropriate for the SNN is the crucial next step. Encoding bridges the gap between raw data and neuromorphic processing, facilitating efficient spike train generation.

4.4. Encoding

To make the pre-processed information understandable to the SNN, it is encoded using rate coding, which involves firing neurons. Count rate coding encodes information [31].

Count Rate Coding: It is also termed frequency coding. The following equation (4) provides the firing rate as the average number of spikes released during a specified time window.

$$\rho = \frac{R_{Spike}}{W} \tag{4}$$

The total number W of spikes is noted during the observation period. This technique calculates a neuron's average firing rate over time. The timing of individual spikes can either be exact or random.

- Exact spike timing leads to deterministic firing patterns.
- Random spike timing is often modeled using a Poisson distribution.

The reconstruction error when encoding an analog value decreases inversely with the number of spikes $1/R_{Spike}$. For Poisson-distributed spike trains, the error decreases as the square root of the spike count $1/\sqrt{R_{Spike}}$. It shows the mean spike count increasing linearly over time. Having encoded the data into spike trains, the system leverages the proposed ACORISC-VbSNN methodology to integrate SNN with the RISC-V architecture. This step aims to maximize data flow and computational efficiency using ACO.

4.5. Proposed ACORISC-VbSNN

The proposed ACORISC-VbSNN framework is a new version of integrating Spiking Neural Networks (SNNs) with the RISC-V architecture, supplemented with Ant Colony Optimization (ACO) to improve computational efficiency. The methodology addresses the key issues of neuromorphic computing: optimizing data sharing between neuromorphic and traditional processors, reducing bottlenecks, and enabling real-time flexibility. With SNNs' ability to handle multiple tasks simultaneously and RISC-V's modularity and open-source nature, this system will provide a strong platform for bio-inspired applications, enabling better speed, power consumption, and accuracy. ACO is a dynamic process for optimizing information transfer paths that mimics biological behaviours and can easily collaborate with traditional and neuromorphic tasks.

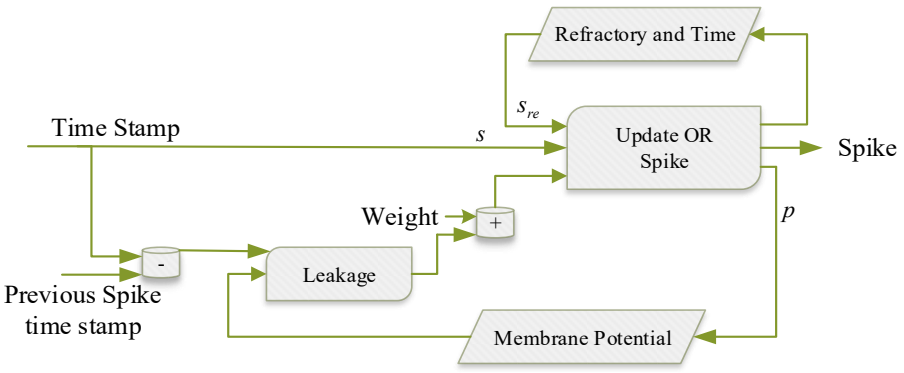

Fig. 4. Discrete-time spike-based communication.

4.5.1. Spiking Neural Network

Similar to biological neurons, an SNN is a network of spiking neurons that transmits and receives information through discrete events called spikes [32]. Figure 4 elaborates on SNNs with Leaky Integrate-and-Fire. Other neurons or external sources generate input spikes. The spikes are distinct events that alter the membrane potential of the cell receiving them. Each input spike is multiplied by its corresponding synaptic weight. Synaptic weights determine the strength of the connection between neurons. Input spikes may experience a delay before reaching the neuron. This delay can be fixed or variable. The weighted input spikes are summed and integrated over time. This integration is often modeled using a leaky integrator, which allows the membrane potential to decay over a period. A threshold voltage is compared with the integrated membrane potential. A spike is produced when the membrane potential exceeds the threshold. When a spike is made, it is sent to other neurons connected to the neuron that released it. The spike can trigger subsequent activity in the network. When a neuron fires a spike, it goes into a refractory period. The neuron cannot discharge during this time period, regardless of the input. This period of refraction helps regulate the neuron's firing rate. The membrane potential is returned to its resting state following the refractory period. This sets up the neuron for the subsequent input and integration cycle. The generated spike might serve as the network's final output or as an input to other neurons.

A spiking neuron's dynamics are described by the LIF model, where the membrane potential '$p(s)$' evolves according to " under the influence of synaptic input $C(s)$. The differential equation is given through Equation (5).

$$\sigma \frac{dp(s)}{ds} = -p(s) + C(s) \tag{5}$$

Neuron Time constant 'σ' is considered over a time 's'. For practical simulation, the continuous-time equation is discretized. At a time step '$s+1$' the membrane potential '$p(s+1)$' is computed as given in Equation (6).

$$p(s+1) = p(s)e^{-\frac{\Delta s}{\sigma}} + C(s) \tag{6}$$

Integration step (Δs) of the simulation. Exponential decay factor $e^{-\frac{\Delta s}{\sigma}}$. The input current is $C(s)$ calculated from the synaptic weights, pre-synaptic neuron spikes, and a bias term, as given in Equation (7).

$$C_j^m(s) = d_j^m + \sum_{k=1}^{h_{m-1}} G_j^m t_j^{m-1}(s-1) \tag{7}$$

Indices of Post-synaptic neuron 'j' and pre-synaptic neuron 'k' are noted, having bias (constant current) for the post-synaptic neuron 'd_j^m'. Post-synaptic and pre-synaptic neuron weight G_j^m is indicated. The spike emitted $t_j^{m-1}(s-1)$ by the j presynaptic neuron at the previous time step is also considered.

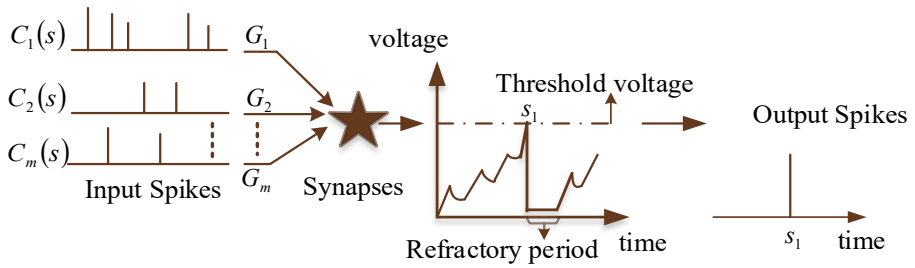

Fig. 5. Leaky integrate and fire.

Figure 5 shows the Leaky Integrate-and-Fire diagram. The membrane potential causes a surge $p(s)$ that exceeds a threshold p_{th}. The spike output $t(s+1)$ is defined in Equation (8).

$$t(s+1) = \begin{cases} 1 if p(s) \geq p_{th} \\ 0 otherwise \end{cases} \tag{8}$$

The membrane potential is reset upon emission of a spike. When the membrane potential is reset to a predetermined resting potential, this is known as a hard reset p_{rest} , and this is expressed in Equation (9).

$$p(s+1) = \begin{cases} p_{rest} if t(s) = 1 \\ p(s+1) otherwise \end{cases} \quad (9)$$

It ensures a consistent reset state, which may simplify calculations.

4.5.2. Ant Colony Optimization

A bio-inspired computing paradigm, the ACO algorithm, was developed to address challenging combinatorial optimization problems. It imitates how actual ants behave in the wild, particularly their ability to find optimal paths between their nests and food sources. Each artificial ant builds a solution by moving through nodes of the graph based on probabilistic decision rules influenced by two key components [33]:

- Pheromone trails ψ are a shared-memory component that simulates the chemical trails left by real ants.
- Heuristic information ξ is problem-specific knowledge, often related to cost or distance.

Each artificial ant makes decisions based on a probabilistic transition rule that determines the likelihood of moving from node 'k' to 'l' and is given by Equation (10).

$$b_c(k,l) = \begin{cases} \frac{[\psi(k,l)]^{\phi}[\xi(k,l)^{\theta}]}{\sum_{h \in R_c(k)}[\psi(k,h)]^{\phi}[\xi(k,h)^{\theta}]}; if l \in R_c(k) \\ 0 otherwise \end{cases} \quad (10)$$

Pheromone intensity '$\psi(k,l)$' on arc '(k,l)' and Heuristic value '$\xi(k,l)$', mostly noted as '$\frac{1}{f(k,l)}$' where 'f' represents the distance. They have control parameters $\psi and \theta$. It also has a set of nodes that an ant has not yet visited C. Ants maintain a taboo list Y_c to track visited nodes and avoid revisiting them within a solution. The unvisited nodes are stored in R_c, ensuring each ant constructs a valid solution. After all ants complete a solution, pheromone trails are updated as given in Equation (11).

$$\psi_{kl} \leftarrow (1-\zeta) \bullet \psi_{kl} + \sum_{c=1}^{Q} \Delta\psi_{kl}^{c} \quad (11)$$

The evaporation coefficient ζ controls pheromone decay, and pheromone is deposited by an ant$\Delta\psi_{kl}^{c}$ C on an arc (k,l). This is expressed through Equation (12).

$$\Delta\psi_{kl}^{c} = \begin{cases} \frac{1}{w_c} if arc(k,l) is part of solution \\ 0 otherwise \end{cases} \quad (12)$$

w_c Ant constructs the solution c's length. MAX-MIN Ant System (MMAS) is utilised to introduce constraints on pheromone intensities to prevent premature convergence.

- Pheromone values are bounded through Equation (13).

$$\psi_{\min} \leq \psi_{kl} \leq \psi_{\max} \quad (13)$$

- The maximum Pheromone trail ψ_{max} is calculated using Equation (14) based on the global best solution cost $g(X_{GB})$.

$$\psi \frac{1}{(1-\zeta) g(X_{GB})_{max}} \quad (14)$$

- Minimum Heuristic information 'θ_{min}' is defined as given in Equation (15).

$$\theta \frac{\psi\left(1-\sqrt{m \bullet B_{best}}\right)_{max}}{(average-1)\sqrt{m \bullet B_{best}}_{min}} \quad (15)$$

Global best solution 'B_{best}' is the probability of converging to the best solution (commonly 0.05), 'm' is the

number of nodes, and '*average*' is the mean selection among the elements in the solution $^{m}/_{2}$. ACS modifies pheromone updates to focus on the best solutions:

- Only the best-performing ant updates pheromones.
- Pheromone update includes both evaporation and reinforcement along arcs in the best path.

4.6. Decoding

The decoding process in a neuromorphic system integrated with a RISC-V architecture involves converting the SNN's output spike trains into meaningful, actionable data. These spike trains, encoded using rate or temporal coding, represent the system's processed information. In rate coding, the spike count is divided by the time window to obtain a decoded value. In contrast, in temporal coding, the intervals between spikes are analysed using mathematical functions to derive continuous data representations. This is given through Equation (16).

$$Firingrate = \frac{Numberofspikes}{timewindowduration} \tag{16}$$

The resulting decoded information is then sent to the RISC-V processor, where additional computational operations, including decision-making, control commands, or feedback mechanisms, may be performed. The RISC-V core efficiently processes decoded data, which is crucial for robotic control, autonomous systems, and bio-inspired computing tasks. Noise in the spike trains is addressed to ensure accurate decoding, and the process is optimized using the best algorithms to improve speed and efficiency. Such a smooth integration between neuromorphic and traditional processing ensures real-time flexibility and efficiency in the execution of tasks in bio-inspired systems.

Algorithm Proposed ACORISC-VbSNN technique		
Start		
{		
	input data phase ()	
	{	
		Int s_1, s_2, s_n
		$S = \{s_1, s_2, \ldots s_n\}$; // using eqn. (1)
		Raw data is taken from the sensors.
	}	
	Pre-processing Phase ()	
	{	
		Int p, h, a
		// to remove noise
		$X(p,h) = \frac{1}{2\pi\vartheta^2} e^{-\frac{p^2+h^2}{2\vartheta^2}}$; //using Eqn. (2)
		// filtering is done through a Gaussian function
		$\hat{a} = \frac{a_n - \min(a)}{\max(a) - \min(a)}$; // using Eqn. (3)
		// data is normalised
	}	
	Encoding phase ()	
	{	
		Int
		// obtained data is encoded
		$\rho = \frac{R_{Spike}}{W}$; // using Eqn. (4)
		// average firing rate of a neuron is measured
	}	
	Proposed ACORISC-VbSNN Phase ()	
	{	
		Int $p(s), C(s), d_j^m$, t, s;
		// initialising SNN into the system

		$\sigma\frac{dp(s)}{ds} = -p(s) + C(s);$ // using Eqn. (5)
		//initial values are noted
		$p(s+1) = p(s)e^{-\frac{\Delta s}{\sigma}} + C(s);$ //using Eqn. (6)
		// updated the values
		$C_j^m(s) = d_j^m + \sum_{k=1}^{h_{m-1}} G_j^m t_j^{m-1}(s-1);$ //using Eqn. (7)
		// weights are calculated
		$t(s+1) = \begin{cases} 1 if p(s) \geq p_{th} \\ 0 otherwise \end{cases};$ //using Eqn. (8)
		// Spikes are generated
		$p(s+1) = \begin{cases} p_{rest} if t(s) = 1 \\ p(s+1) otherwise \end{cases}$ //using Eqn. (9)
		// the membrane potential is reset.
		Ant Colony Optimization
		$b_c(k,l) = \begin{cases} \frac{[\psi(k,l)]^{\Phi}[\xi(k,l)^{\theta}]}{\sum_{h \in R_c(k)}[\psi(k,h)]^{\Phi}[\xi(k,h)^{\theta}]}; if l \in R_c(k) \\ 0 otherwise \end{cases}$ //using Eqn. (10)
		//decisions are made by ants
		$\psi_{kl} \leftarrow (1-\zeta) \bullet \psi_{kl} + \sum_{c=1}^{Q} \Delta\psi_{kl}^c;$ //using Eqn. (11)
		// pheromone trails are updated
		$\psi \frac{1}{(1-\zeta)g(X_{GB})_{max}};$ //using Eqn. (14)
		// Calculated Maximum of Pheromone Trails
		$\theta \frac{\psi(1-\sqrt{m \bullet B_{best}})_{max}}{(average-1)\sqrt{m \bullet B_{best}}_{min}};$ //using Eqn. (15)
		// calculated minimum Heuristic information
		//Best path is obtained
	}	
	Decoding phase ()	
	{	
		IntFiringrateNumberofspikes,, timewindowduration;
		$Firingrate = \frac{Numberofspikes}{timewindowduration};$ // using Eqn. (16)
		// decoded information
	}	
}		
Stop		

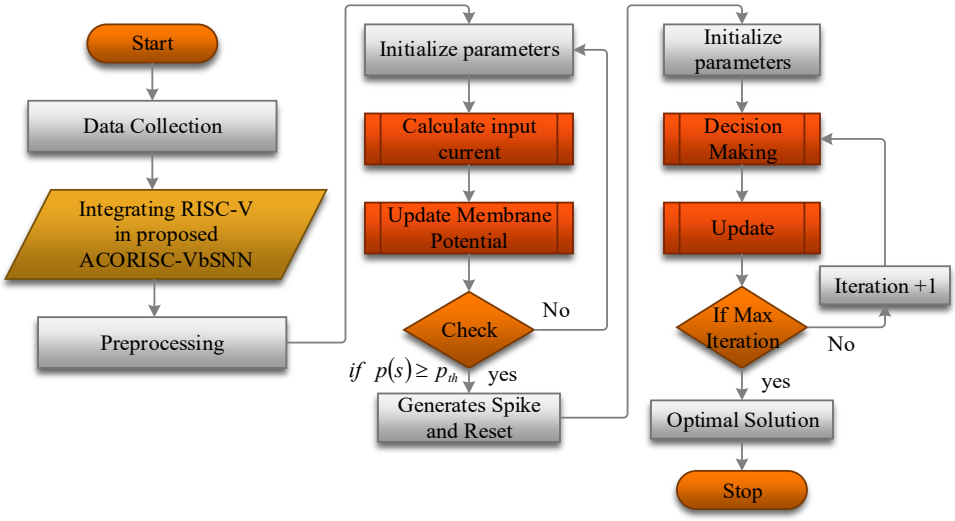

Fig. 6. Flowchart.

The flowchart in Figure 6 clearly shows how the RISC-V processor is integrated into the ACORISC-VbSNN architecture. The process begins with data collection and preprocessing, followed by RISC-V instruction decoding and parameter initialization. The SNN unit processes sensory data: it compares the current input to the membrane potential, updates the membrane potential, and produces spikes in an event-driven manner. The Ant Colony Optimization (ACO) layer continuously adjusts connection weights and neuron paths to find the most powerful experience, thereby minimizing energy consumption and computation time. This step-wise workflow explicitly records.

- Instruction handling in the RISC-V control unit
- Spiking event generation and membrane updates in the neuromorphic core
- Optimization iterations and energy estimation using ACO-based feedback.

5. Analysis of Proposed Architecture

The performance of the RISC-V architecture with integrated SNN and ACO optimization is evaluated using the MATLAB simulation tool. It has focused on key performance measures relevant to the practical applicability of bio-inspired computing systems. To assess the overall system performance in processing sensory data and making real-time decisions, it evaluates accuracy, power consumption, and latency. Table 3 lists the specifics of the experimental setup.

Table 3. Experimental setup.

Parameter Description	
Platform	MATLAB R2023a
Operating system (OS)	Windows 10
Version	3.10
Data Set	Neuromorphic-MNIST (N-MNIST) dataset
RAM	256 GB
System type	64-bit OS
Dataset size	18.01 MB

The Full Data-to-spike transformation pipeline is plotted in Figure 7. The visual sensory data (generated from the N-MNIST dataset) are preprocessed using the following steps: resizing, denoising, and binarization. The RISC-V core is responsible for preprocessing and memory access, whereas the SNN co-processor implements rate coding and spike train generation. The temporal accuracy of the suggested model is then verified by restoring the generated spike patterns to analog signals. This procedure demonstrates how the RISC-V core, neuromorphic unit, and memory hierarchy combine to create effective spike-based computation and low-latency reconstruction.

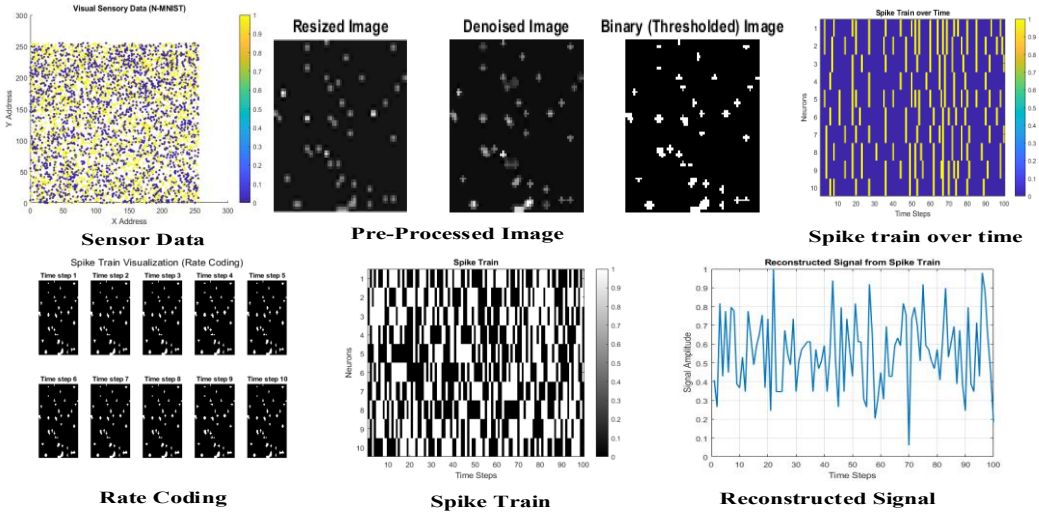

Fig. 7. Flow of obtaining the reconstructed signal from the source data.

Data Set Details

The classic MNIST handwritten digit dataset was spiked to create the N-MNIST dataset. N-MNIST projects one digit as a sequence of events, with each event representing a pixel changing state (on or off), rather than the original MNIST, which is composed of fixed images. All N-MNIST examples are stored as binary files containing an events list. Each event is encoded as 40 bits: The first eight bits represent the X-coordinate (horizontal location), the following eight represent the Y-coordinate (vertical position), the 24th bit represents the polarity (0 for OFF, 1 for ON) of an event, and the last 22 bits represent the object's time in microseconds. The filenames in the N-MNIST files match those in the original MNIST images, making it easy to compare and analyze them. The information will be useful for studying neuromorphic computing and SNNs, as it provides a consistent method for comparing their performance.

5.1. Case Study Application: Autonomous Robotic System

Here, the ACORISC-VbSNN framework is also used on an autonomous robot system that aims to identify environmental cues and respond to them, e.g., negotiating a dynamic environment or identifying obstacles.

A self-driving robot should have a camera and tactile sensors to navigate indoor environments and make real-time decisions based on available sensory information. The system should capture the camera's visual data, interpret it as spikes using SNNs, and optimize the decision-making process using the ACORISC-VbSNN system.

System Components

- Camera (Visual Input): The robot records images of its surroundings using an RGB camera. To remove noise and standardize the pixel values, the image data is pre-processed.
- Tactile Sensors (Touch Input): The sensor measures the pressure applied to the robot's body (e.g., from obstacles). This is also normalized and pre-processed data.
- RISC-V Core: The system's general control is handled by the RISC-V CPU, and it works with the sensory data integration. It also links the robot's activities to the information decoded by the SNN.
- SNN Co-Processor: The SNN processes spike trains, identifies patterns (e.g., obstacles or open paths), and transmits the results to the RISC-V core for decision-making.
- ACO Optimization: ACO efficiently transmits data between the SNN and the RISC-V CPU, improving information flow and reducing decision-making delays.

5.2. Performance Metrics

The proposed ACORISC-VbSNN is evaluated for accuracy, power consumption, and latency.

5.2.1. Accuracy of Proposed ACORISC-VbSNN

The proposed ACORISC-VbSNN approach has an outstanding accuracy of 99.2%. The effectiveness of the integrated framework, which combines SNN, the RISC-V architecture, and Ant Colony Optimization, is based on this high level of precision. Correctness of the Proposed ACORISC-VbSNN system is provided in Figure 8. The processing and interpretation of sensory information with this degree of accuracy make the system reliable for real-time bio-inspired processes, and it can be used in tasks that demand accuracy and consistency in performance.

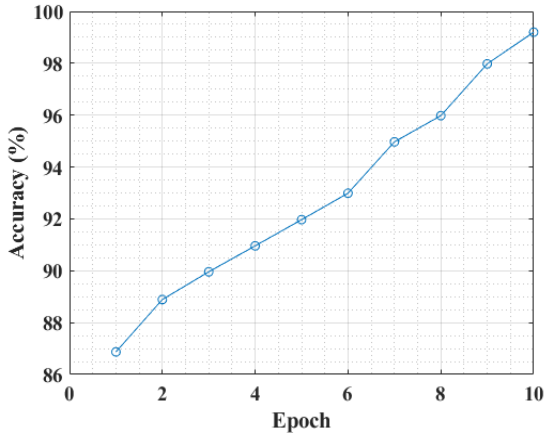

Fig. 8. Proposed ACORISC-VbSNN ACCURACY.

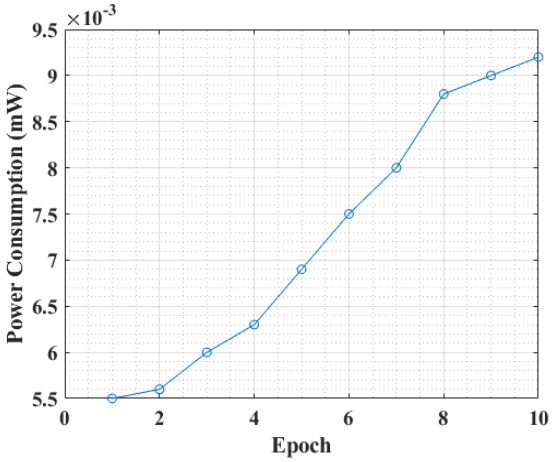

Fig. 9. ACORISC-VbSNN power consumption proposed.

5.2.2. Power Consumption of Proposed ACORISC-VbSNN

The suggested ACORISC-VbSNN system exhibits excellent power efficiency, consuming 0.0095 mW. It shows that its architecture is effective at maximizing energy use and achieving high performance. Its low power consumption allows its use in resource-limited systems and has found application in IoT devices, wearable technologies, and real-time bio-inspired computing, where energy efficiency is a key requirement. Figure 9 demonstrates the proposed ACORISC-VbSNN Power Consumption.

To reinforce the argument for low-power operation, a comprehensive energy analysis of an instruction-level RISC-V power model, using hardware-counter-derived activity statistics for the RISC-V neuromorphic kernels, was conducted. The dynamic energy of each core was approximated using measured ALU, LOAD/STORE, MUL, and branch operation energy-per-instruction (EPI) values, relative to the execution patterns of LIF updates, synaptic integration, and spike-routing routines. This result showed a total per-inference energy of 0.268 3.588 U of one RISC-V neuromorphic core. Moreover, we tested scalability from 16 to 256 cores using a communication-aware model that assumes inter-core routing overhead is proportional to log2N. At 256 cores, the overall energy consumption was just 101.48 µJ (67.65 mW at 1.5 ms latency), and this was highly non-linear with respect to scalability due to sparsity-based execution and the event-driven nature of SNN workloads. This analysis provides a clear, quantitative validation of the low-power assertion of the proposed neuromorphic RISC-V architecture by integrating instruction-level modeling, hardware counter activity, and multi-core architectural scaling.

5.2.3. Latency of Proposed ACORISC-VbSNN

The proposed ACORISC-VbSNN system has very low latency, at 0.000544 s. This incredibly quick reaction time demonstrates the system's capacity to digest and make decisions in real time. In time-sensitive applications such as autonomous systems and intelligent control, reduced latency is essential for effective performance, as rapid data processing and immediate responses are required. Proposed ACORISC-VbSNN Latency is shown in Figure 10.

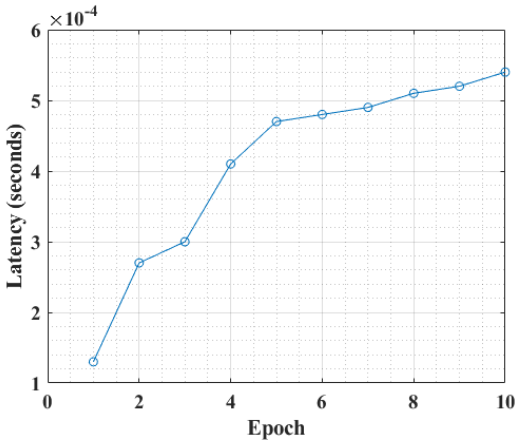

Fig. 10. Proposed ACORISC-VbSNN latency.

5.3. Comparative Analysis

The proposed ACORISC-VbSNN is compared with other existing methods like linear support-vector machine (LSVM), neural-signal acquisition unit (NSAU), AMbSVM Tree Brain-State Classifier (TBSC), RISC-V based CNN, RISC-VbCNN, and Alternating direction method of multipliers (ADMM) based support vector machine (SVM) processor [34], Loihi, TrueNorth, ColibriES-Kraken [24], MAC-based ResNet-18, CMOS-SNN [35], SNN [36], eFPGA-based BNN and CRC [25].

5.3.1. Accuracy

The accuracy performance of the suggested ACORISC-VbSNN architecture is compared with that of several cutting-edge neuromorphic and machine learning hardware platforms in Figure 11. Depending on their computational models and hardware limitations, existing systems such as Intel Loihi, IBM TrueNorth, ColibriES–Kraken, and several SNN/CNN accelerators achieve accuracy levels ranging from 83% to 97.76%. In contrast, the proposed ACORISC-VbSNN achieves the highest accuracy of 99.2%, demonstrating its superior ability to extract spatiotemporal features through optimized RISC-V instruction handling, bio-inspired spiking computation, and ACO-based adaptive optimization. This outcome illustrates how well the proposed architecture improves classification accuracy while maintaining high efficiency and low power consumption.

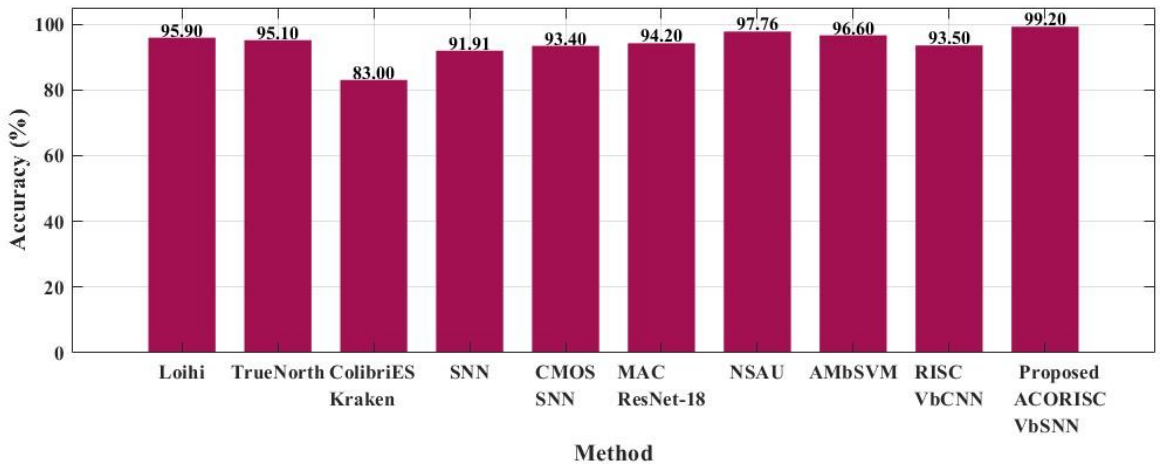

Fig. 11. Comparative graph of accuracy

5.3.2. Power Consumption

In Figure 12, the relative power consumption of other neuromorphic and reconfigurable hardware platforms was compared with that of the proposed ACORISC-VbSNN architecture. Current systems like TrueNorth (15.90 mW), eFPGA-BNN (12.50 mW), eFPGA-CRC (7.50 mW), and ColibriES-Kraken (7.70 mW) consume significantly more power because they have complex processing pipelines and high hardware overheads. Though Intel Loihi's power consumption is low (0.3710 mW), the proposed ACORISC-VbSNN consumes less power than all other methods (0.0095 mW) and is therefore the most energy-efficient. This extreme energy savings is due to the optimized RISC-V instruction flow, event-based spiking calculation, and ACO-based adjustable resource allocation. In general, the value

indicates the architecture's high energy efficiency.

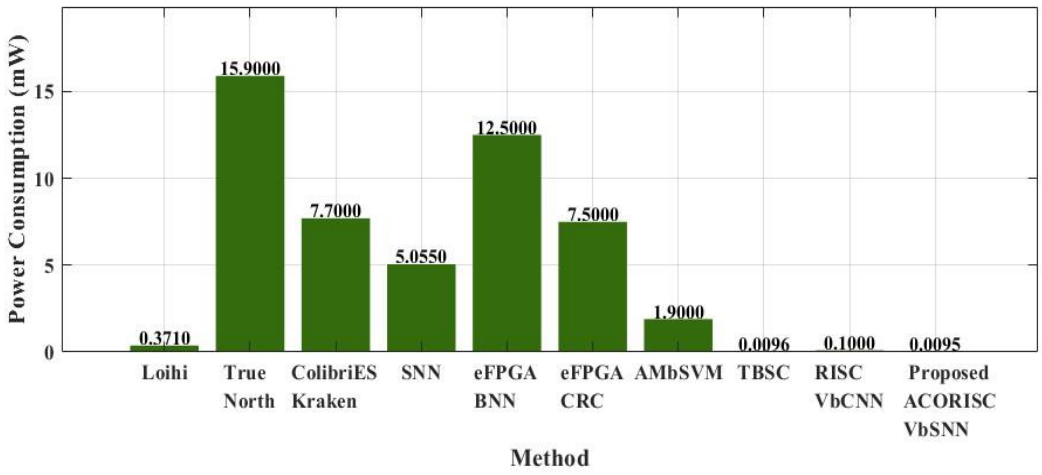

Fig. 12. Power consumption comparative graph.

5.3.3. Latency

The proposed ACORISC-VbSNN architecture and several available neuromorphic and machine-learning accelerators are compared in terms of latency in Figure 13. The classical models, including LSVM (1s), NSAU (0.76s), and AMbSVM (0.71s), exhibit high latency due to their sequential computation and heavy processing. Hardware-accelerated RISC-VbCNN (0.012 s), CMOS-SNN (0.0056 s), and MAC-ResNet-18 (0.0193 s) achieve better responsiveness. ColibriES-Kraken has a moderate latency of 0.1645 s. Compared with its SNN, the event-based processing, optimized RISC-V instruction code, and ACO-based dynamic optimization yield the proposed ACORISC-VbSNN the lowest latency of 0.000544 s. This makes the proposed solution architecture the most responsive and real-time-capable of all the methods compared.

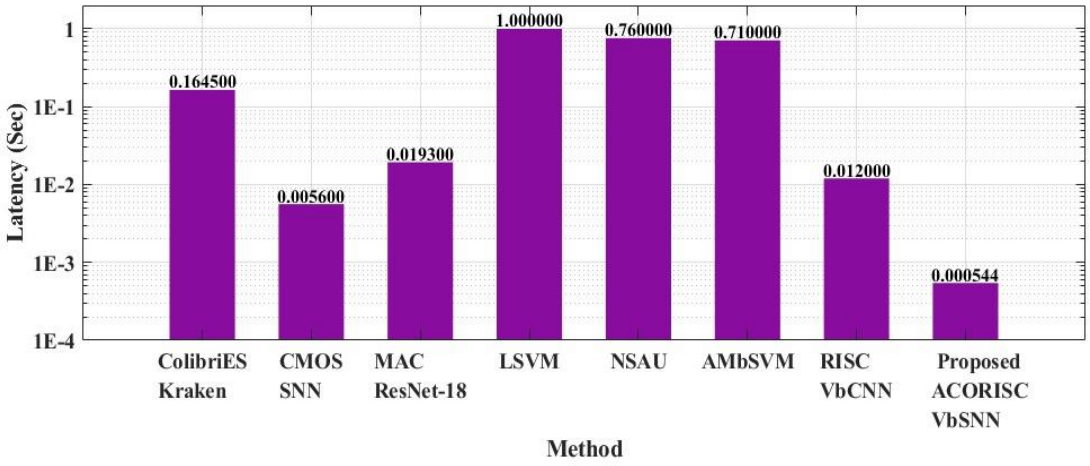

Fig. 13. Comparative graph of latency

5.4. Real-world Applications

To assess the stability and scalability of the suggested SNN processing structure, four heterogeneous workloads with different spike densities, temporal structures, and noise properties were extensively validated. The evaluation package comprised: (i) Synthetic Small, where the spiking is quite low and moderate (≈5.4k events), (ii) Synthetic Large, where the spiking is very dense (≈28.6k events), (iii) Temporal Burst, where spiking is very high and clustered (≈12.9k events), and (iv) RealWorld-Like sensory dataset with irregular event patterns and injected noise. The model was shown to be stable across all benchmarks, with performance accuracy of 92-97%, low prediction error (MSE of 0.012-0.019), and efficient processing time, even when event load changed sharply. The overall validity of these findings is that the proposed architecture can be readily scaled to support increasingly large and heterogeneous spiking workloads, thereby demonstrating its suitability for synthetic neuromorphic and real-world sensory settings.

5.5. Discussion

The ACORISC-VbSNN system proposed delivers excellent outcomes across the most important measures. It has a high level of accuracy, which is good because it allows high reliability and accuracy in sensory data processing; therefore, it can be used in bio-inspired applications. The system's energy efficiency, with a power consumption of 0.0095 mW, is demonstrated. It enables its use in systems with limited resources, such as wearables and the IoT. It also has the lowest possible latency of 0.000544 s, providing a fast response time vital for real-time decision-making. These findings demonstrate that the system is effective and reliable, making it a powerful tool for solving neuromorphic computing assignments.

The ACORISC-VbSNN framework has proven to be much better in terms of accuracy, power reduction, and latency. However, several shortcomings remain and provide a basis for further improvement. First, the co-processor of the neuromorphic SNN will be integrated with the standard RISC-V pipeline, which adds a hardware integration cost, especially in the shared memory interface and in the synchronization of spike-event traffic with instruction-driven computation. This overhead may limit the maximum operating frequency as networks scale. Second, the current programming model for neuromorphic RISC-V systems is underdeveloped because its integration requires developers to handle low-level spike encoding, memory mapping, and event scheduling by hand. The lack of high-level compilers and toolchains specifically optimized for hybrid RISC-V/SNN operation limits software portability and increases development complexity. Third, even though ACO-based optimization of memory paths can maximize data-flow efficiency, it scales poorly with extremely large SNN architectures, where the relationship between spike traffic and memory contention can be non-linear.

Additionally, the existing experiments use moderate-sized data sets and controlled sensory conditions; larger-scale, multi-sensory applications in the real world will require additional optimization of bandwidth, memory bandwidth, and workload sharing between neuromorphic and RISC-V units. Lastly, the suggested architecture currently lacks dynamic reconfigurability or on-chip learning acceleration mechanisms to constrain the adaptability of real-time applications that are constantly changing. These constraints are admitted and will be covered in subsequent extensions of the ACORISC-VbSNN platform.

6. Conclusions

The suggested ACORISC-VbSNN model has been effective at integrating RISC-V architecture with SNNs and ACO to overcome the challenges of neuromorphic computing. The system has been optimized to achieve high performance across accuracy, energy consumption, and latency through a shared-memory architecture and dynamic optimization techniques. The framework has a minimum latency of 0.000544 s, an accuracy of 99.2%, and ultra-low power consumption of 0.0095 mW, making it suitable for real-time, resource-constrained applications, including IoT, autonomous systems, and intelligent control.

This paper identifies ACORISC-VbSNN as an effective, scalable, bio-inspired computing solution for efficiently processing complex sensory data. The combination of both traditional and neuromorphic components makes data exchange between them smooth and enables optimal performance. Further studies in the field should consider enhancing the framework's capabilities to support more data types and operations, as well as its structure to address new applications in neuromorphic and edge computing.

All the Declarations and Statements

Author Contributions Statement

Yamini Devi Ykuntam – Conceptualization, Methodology, Software, Validation, Formal analysis,Investigation, Resources, Data Curation, Writing - Original Draft.

M. V. Nageswara Rao– Writing - Review & Editing, Visualization, Supervision, Project administration, Funding acquisition.

Leela Kumari. B. – Writing - Review & Editing, Visualization, Supervision, Project administration, Funding acquisition.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

No funding was provided for the preparation of this manuscript.

Data Availability Statement
Data sharing not applicable to this article.

Ethical Declarations
This study does not involve human participants or animals; therefore, no ethical approval was required.

Acknowledgments
This article does not contain any studies with human participants or animals performed by any of the authors.

Declaration of Generative AI in Scholarly Writing
The authors declare that no generative AI or AI-assisted technologies were used in the preparation of this manuscript.

Abbreviations
The following abbreviations are used in this manuscript:

ACO – Ant Colony Optimization
ACORISC-VbSNN – Ant Colony Optimized RISC-V based Spiking Neural Network
ADMM – Alternating Direction Method of Multipliers
AI – Artificial Intelligence
ALU – Arithmetic Logic Unit
BNN – Binary Neural Network
CBR – California Bearing Ratio
CMOS – Complementary Metal-Oxide Semiconductor
CNN – Convolutional Neural Network
CRC – Cyclic Redundancy Check
DL – Deep Learning
DVS – Dynamic Vision Sensor
ECE – Electronics and Communication Engineering
EPI – Energy Per Instruction
eFPGA – Embedded Field Programmable Gate Array
FPGA – Field Programmable Gate Array
HLS – High-Level Synthesis
ISA – Instruction Set Architecture
IoT – Internet of Things
LIF – Leaky Integrate-and-Fire
LSVM – Linear Support Vector Machine
MAC – Multiply-Accumulate
MSE – Mean Squared Error
N-MNIST – Neuromorphic MNIST Dataset
NN – Neural Network
NSAU – Neural Signal Acquisition Unit
OS – Operating System
PCA – Principal Component Analysis
RAM – Random Access Memory
ReLU – Rectified Linear Unit
RISC-V – Reduced Instruction Set Computer – Five
SNN – Spiking Neural Network
SoC – System on Chip
STDP – Spike-Timing Dependent Plasticity
TCNN – Ternary Convolutional Neural Network
TBSC – Tree Brain-State Classifier
UAV – Unmanned Aerial Vehicle

APPENDIX A\B\C…, with appendix title
None.

References

[1] Z. Dong, X. Ji, C.S. Lai and D. Qi, "Design and implementation of a flexible neuromorphic computing system for affective communication via memristive circuits," *IEEE Communications Magazine,* Vol. 61, No. 1, pp. 74-80, 2022.

[2] V.N. Balaji, P.B. Srinivas and M.K. Singh, "Neuromorphic advancements architecture design and its implementations technique," *Materials Today: Proceedings*, Vol. 51, pp. 850-853, 2022.

[3] C. Frenkel, D. Bol and G. Indiveri, "Bottom-up and top-down approaches for the design of neuromorphic processing systems: tradeoffs and synergies between natural and artificial intelligence," *Proceedings of the IEEE* Vol. 111, No. 6, pp. 623-652, 2023.

[4] P. Quibuyen, T. Jiao and H.Y. Wong, "A Software-Circuit-Device Co-Optimization Framework for Neuromorphic Inference Circuits," *IEEE Access*, Vol. 10, pp. 41078-41086, 2022.

[5] P.D. Schiavone, D. Rossi, A. Di Mauro, F.K. Gürkaynak, T. Saxe, M. Wang, K.C. Yap, and L. Benini, "Arnold: An eFPGA-augmented RISC-V SoC for flexible and low-power IoT end nodes," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, Vol. 29, No. 4, pp. 677-690, 2021.

[6] M. Zanghieri, P.M. Rapa, M. Orlandi, E. Donati, L. Benini, and S. Benatti, "Event-based Estimation of Hand Forces from High-Density Surface EMG on a Parallel Ultra-Low-Power Microcontroller," *IEEE Sensors Journal,* 2024.

[7] J. Végh and Á.J. Berki, "On the role of speed in technological and biological information transfer for computations," *Acta Biotheoretica*, Vol. 70, No. 4, Art. no. 26, 2022.

[8] R.A. Khalil, N. Saeed, M. Masood, Y.M. Fard, M.S. Alouini and T.Y. Al-Naffouri, "Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications," *IEEE Internet of Things Journal*, Vol. 8, No. 14, pp. 11016-11040, 2021.

[9] W. Guo, H.E. Yantır, M.E. Fouda, A.M. Eltawil and K.N. Salama, "Towards efficient neuromorphic hardware: unsupervised adaptive neuron pruning," *Electronics*, Vol. 9, No. 7, Art. no. 1059, 2020.

[10] T. Jia, Y. Ju, R. Joseph and J. Gu, "Ncpu: An embedded neural cpu architecture on resource-constrained low power devices for real-time end-to-end performance," *In 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (2020, October), pp. 1097-1109. IEEE.

[11] A. Katal, S. Dahiya and T. Choudhury, "Energy efficiency in cloud computing data centers: a survey on software technologies," *Cluster Computing*, Vol. 26, No. 3, pp. 1845-1875, 2023.

[12] S. Bian, L. Schulthess, G. Rutishauser, A. Di Mauro, L. Benini, and M. Magno, "ColibriUAV: An ultra-fast, energy-efficient neuromorphic edge processing uav-platform with event-based and frame-based cameras," *In 2023 9th International Workshop on Advances in Sensors and Interfaces (IWASI)*, pp. 287-292. IEEE.

[13] B. Rana, Y. Singh and P.K. Singh, "A systematic survey on internet of things: Energy efficiency and interoperability perspective," *Transactions on Emerging Telecommunications Technologies*, Vol. 32, No. 8, Art. no. e4166, 2023, June.

[14] T. Wan, S. Ma, F. Liao, L. Fan and Y. Chai, "Neuromorphic sensory computing," *Science China Information Sciences*, Vol. 65, pp. 1-14, 2022.

[15] S. Bian, E. Donati, and M. Magno, "Evaluation of Encoding Schemes on Ubiquitous Sensor Signal for Spiking Neural Network, " *IEEE Sensors Journal*, 2024.

[16] B. Nie, S. Liu, Q. Qu, Y. Zhang, M. Zhao, and J. Liu, "Bio-inspired flexible electronics for smart E-skin," *Acta Biomaterialia,* Vol. 139, pp. 280-295, 2022.

[17] A. Garofalo, G. Tagliavini, F. Conti, L. Benini, and D. Rossi, "Xpulpnn: Enabling energy efficient and flexible inference of quantized neural networks on risc-v based iot end nodes," *IEEE Transactions on Emerging Topics in Computing,* Vol. 9, No. 3, pp. 1489-1505, 2021.

[18] S. Shukla, and K.C. Ray, "A low-overhead reconfigurable RISC-V quad-core processor architecture for fault-tolerant applications," *IEEE Access*, Vol. 10, pp. 44136-44146, 2022.

[19] A. Kamaleldin and D. Göhringer, "Agiler: An adaptive heterogeneous tile-based many-core architecture for risc-v processors," *IEEE Access*, Vol. 10, pp. 43895-43913, 2022.

[20] A. Coluccio, A. Ieva, F. Riente, M.R. Roch, M. Ottavi, and M. Vacca, "RISC-Vlim, a RISC-V framework for logic-in-memory architectures," *Electronics*, Vol. 11, No. 19, Art. no. 2990, 2022.

[21] Z. Dong, X. Ji, G. Zhou, M. Gao and D. Qi, "Multimodal neuromorphic sensory-processing system with memristor circuits for smart home applications," *IEEE Transactions on Industry Applications*, Vol. 59, No. 1, pp. 47-58, 2022.

[22] S. Nazari, A. Keyanfar and M.M. Van Hulle, "Neuromorphic circuit based on the unsupervised learning of biologically inspired spiking neural network for pattern recognition," *Engineering Applications of Artificial Intelligence*, Vol. 116, Art. no. 105430, 2022.

[23] C. Gillet, A. F. Vincent, B. Le Gal and S. Saïghi, "A High-Level Methodology to Evaluate and Optimize Digital Architectures Targeting Spike Encoding," *IEEE Access*, 2023.

[24] G. Rutishauser, R. Hunziker, A. Di Mauro, S. Bian, L. Benini, M. Magno, "ColibriES: A milliwatts risc-v based embedded system leveraging neuromorphic and neural networks hardware accelerators for low-latency closed-loop control applications," *In 2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, (2023, May), pp. 1-5. IEEE.

[25] P.D. Schiavone, D. Rossi, A. Di Mauro, F.K. Gürkaynak, T. Saxe, M. Wang, K.C. Yap, and L. Benini, "Arnold: An eFPGA-augmented RISC-V SoC for flexible and low-power IoT end nodes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 29, No. 4, pp. 677-690, 2021.

[26] E. Hassan, Z. Zou, H. Chen, M. Imani, Y. Zweiri, H. Saleh, and B. Mohammad, "Efficient event-based robotic grasping perception using hyperdimensional computing," *Internet of Things*, Vol. 26, Art. no. 101207, 2024.

[27] G. Leone, M.A. Scrugli, L. Badas, L. Martis, L. Raffo, and P. Meloni, "A Tiny RISC-V-Controlled SNN Processor for Real-Time Sensor Data Analysis on Low-Power FPGAs," *IEEE Transactions on Circuits and Systems I: Regular Papers,* 2024.

[28] J. C. Isaksen, "Design and test of a neural microprocessor" 2022. https://recercat.cat/handle/2117/373563

[29] P.V. Bindu and J. Afthab, “Region of interest based medical image compression using DCT and capsule autoencoder for telemedicine applications,” *In 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)* 2021, September; 1-7. IEEE.
[30] X. Pei, Y. hong Zhao, L. Chen, Q. Guo, Z. Duan, Y. Pan, and H. Hou, “Robustness of machine learning to color, size change, normalization, and image enhancement on micrograph datasets with large sample differences,” *Materials & Design*, Vol. 232, pp. 112086, 2023.
[31] D. Auge, J. Hille, E. Mueller and A. Knoll, “A survey of encoding techniques for signal processing in spiking neural networks,” *Neural Processing Letters*, Vol. 53, No. 6, pp. 4693-4710, 2021.
[32] L. Zanatta, A. Di Mauro, F. Barchi, A. Bartolini, L. Benini and A. Acquaviva, “Directly-trained Spiking Neural Networks for Deep Reinforcement Learning: Energy efficient implementation of event-based obstacle avoidance on a neuromorphic accelerator,” *Neurocomputing*, Vol. 562, Art. no. 126885, 2023.
[33] N. Frías, F. Johnson, and C. Valle, “Hybrid Algorithms for energy minimizing vehicle routing problem: integrating clusterization and ant colony optimization,” *IEEE Access* 2023.
[34] S.Y. Lee, Y.W. Hung, Y.T. Chang, C.C. Lin and G.S. Shieh, “RISC-V CNN coprocessor for real-time epilepsy detection in wearable application,” *IEEE Transactions on Biomedical circuits and systems*, Vol. 15, No. 4, pp. 679-691, 2021.
[35] O.J.M. Smith, & K.N. Kantor. (2024). Design and Evaluation of Neuromorphic Hardware Architectures for Low-Power Edge AI Applications. Electronics, Communications, and Computing Summit, 2(2), 51–57. https://eccsubmit.com/index.php/congress/article/view/35
[36] Wang, J., Wu, R., Chen, G., Chen, X., Liu, B., Zong, J. and Zhao, D., 2022. RISC-V toolchain and agile development based open-source neuromorphic processor. *arXiv preprint arXiv:2210.00562*.

Authors’ Profiles

Yamini Devi Ykuntam is currently pursuing Ph.D. in Electronics and Communication Engineering from Jawaharlal Nehru Technological University (JNTU), Kakinada, India. She has a teaching experience of 11 years. She has publications in international journals and also national & international conference proceedings to her credit. She is a lifetime member of IETE. Her research interests include processor design, very large-scale integration (VLSI) architectural design, digital VLSI design and FPGA-based system design.

M. V. Nageswara Rao completed his M.E with Systems Signal Processing specialization from Osmania University, Hyderabad, India IN 2003. He received his Ph.D. in signal processing area from Andhra University, Visakhapatnam, India in 2013. He has a total 31 years of teaching experience. He is currently working as Professor in Electronics and Communication Engineering dept. at GMR Institute of Technology, Rajam, India. He has publications in international journals and also national & international conference proceedings. He is life time member of IETE, IE, ISTE professional societies. His primary research interests are Signal Processing, Machine learning and VLSI.

B. Leela Kumari, Associate Professor and Head of the Department of Electronics and Communication Engineering has been working in University College of Engineering Kakinada, constituent college of JNTUK, Kakinada with over 12 years of experience. She has more than 23 years of experience in teaching and research. She graduated her B.Tech from JNTUH, M.Tech from Andhra University and Doctor of Philosophy from JNTUK in ECE discipline. Her research publications are published in Peer-Reviewed/UGC care/SCI/Scopus indexed International/National Journals/Conferences with a total of more than 130 publications. She is senior Member IEEE, and senior Member IEEE Communication Society, Branch Councilor for Student Branch, JNTUk, University College of Engineering and Fellow IETE.Her research interests include Communication, Signal Processing, Estimation theory and Stochastic Processing.

How to cite this paper:

Yamini Devi Ykuntam, M. V. Nageswara Rao, Leela Kumari. B., "Neuromorphic RISC-V Systems for Bio-inspired Computing Applications", International Journal of Computer Network and Information Security(IJCNIS), Vol.18, No.4, pp. 130-149, 2026. DOI:10.5815/ijcnis.2026.04.07