

Multi-scroll hopfield neural network excited by memristive self-synapses and its application in image encryption

Ting He(何婷), Fei Yu(余飞), Yue Lin(林越), Shaoqi He(何邵祁), Wei Yao(姚卫), Shuo Cai(蔡烁), and Jie jin(金杰)
Citation: Chin. Phys. B, 2025, 34 (**12**): 120506. DOI: 10.1088/1674-1056/adfeff
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Coexisting and multiple scroll attractors in a Hopfield neural network with a controlled memristor

Qing-Qing Ma(马青青), An-Jiang Lu(陆安江), and Zhi Huang(黄智)
Chin. Phys. B, 2024, 33 (**12**): 120502. DOI: 10.1088/1674-1056/ad8148

Fractional-order heterogeneous memristive Rulkov neuronal network and its medical image watermarking application

Dawei Ding(丁大为), Yan Niu(牛炎), Hongwei Zhang(张红伟), Zongli Yang(杨宗立), Jin Wang(王金), Wei Wang(王威), and Mouyuan Wang(王谋媛)
Chin. Phys. B, 2024, 33 (**5**): 050503. DOI: 10.1088/1674-1056/ad322b

Coexistence behavior of asymmetric attractors in hyperbolic-type memristive Hopfield neural network and its application in image encryption

Xiaoxia Li(李晓霞), Qianqian He(何倩倩), Tianyi Yu(余天意),Zhuang Cai(才壮), and Guizhi Xu(徐桂芝)
Chin. Phys. B, 2024, 33 (**3**): 030505. DOI: 10.1088/1674-1056/acf281

FPGA implementation and image encryption application of a new PRNG based on a memristive Hopfield neural network with a special activation gradient

Fei Yu(余飞), Zinan Zhang(张梓楠), Hui Shen(沈辉), Yuanyuan Huang(黄园媛), Shuo Cai(蔡烁), and Sichun Du(杜四春)
Chin. Phys. B, 2022, 31 (**2**): 020505. DOI: 10.1088/1674-1056/ac3cb2

New results for exponential synchronization of linearly coupled ordinary differential systems

Ping Tong(童评), Shi-Hua Chen(陈士华)
Chin. Phys. B, 2017, 26 (**5**): 050503. DOI: 10.1088/1674-1056/26/5/050503

SPECIAL TOPIC — Biophysical circuits: Modeling & applications in neuroscience

Multi-scroll hopfield neural network excited by memristive self-synapses and its application in image encryption

Ting He(何婷)[1], Fei Yu(余飞)[2], Yue Lin(林越)[1], Shaoqi He(何邵祁)[2], Wei Yao(姚卫)[2], Shuo Cai(蔡烁)[2], and Jie jin(金杰)[3,†]

[1] *School of Computer Science and Technology, ChangSha University of Science and Technology, Changsha* 410114, *China*
[2] *School of Physics and Electronic Science, ChangSha University of Science and Technology, Changsha* 410114, *China*
[3] *School of Information Engineering, Changsha Medical University, Changsha* 410219, *China*

(Received 6 August 2025; revised manuscript received 20 August 2025; accepted manuscript online 26 August 2025)

The functionality of the biological brain is closely related to the dynamic behavior generated by synapses in its complex neural system. The self-connection synapse, as a critical form of feedback synapse in Hopfield neurons, plays an essential role in understanding the dynamic behavior of the brain. Synaptic memristors can bring neural network models closer to the complexity of the brain's neural networks. Inspired by this, this study incorporates the nonlinear memory characteristics of synapses into the Hopfield neural network (HNN) by replacing a single self-synapse in a four-dimensional HNN model with a novel cosine memristor model, aiming to more realistically reproduce the dynamical behavior of biological neurons in artificial systems. By performing a dynamical analysis of the system using numerical methods, we find that the model exhibits infinitely many equilibrium points and can induce the formation of rare transient attractors, as well as an arbitrary number of multi-scroll attractors. Additionally, the model demonstrates complex coexisting attractor dynamics, including transient chaos, periodicity, decaying periodicity, and coexisting chaos. Furthermore, the feasibility of the proposed HNN model is verified using a field-programmable gate array (FPGA). Finally, an electronic codebook (ECB)–mode block cipher encryption algorithm is proposed for image encryption. The encryption performance is evaluated, with an information entropy value of 7.9993, demonstrating the excellent randomness of the system-generated numbers.

Keywords: self-connected synapses, Hopfield neural network, multi-scroll attractor, field programmable gate array, image encryption

PACS: 05.45.–a, 05.45.Gg, 07.05.Mh, 87.19.lj

DOI: 10.1088/1674-1056/adfeff **CSTR:** 32038.14.CPB.adfeff

1. Introduction

The nervous system in the human brain can be described as a highly complex nonlinear system composed of multiple artificial neurons interconnected with each other. Neural networks can exhibit complex dynamic characteristics through interactions between neurons and are highly nonlinear dynamical systems with adaptive, self-organizing, and real-time learning capabilities.[1–4] In particular, the unique memory mechanisms and learning abilities of the brain provide an important foundation for exploring neurodynamics. To this end, researchers have simulated and studied the neural mechanisms and biological structures of the brain by combining multiple neurons, thereby developing artificial neural networks composed of many interconnected neurons.[5–7] Biological experiments have shown that chaos exists in the nervous system of the brain, and neural networks are closely associated with chaotic dynamics.[8] Therefore, chaotic neural networks are regarded as a new generation of information processing technology and have attracted interest in many fields, such as secure communication,[9] pattern recognition,[10] and emotional learning.[11,12]

The Hopfield neural network (HNN), as an associative memory model with self-feedback characteristics and capable of simulating key mechanisms of memory storage and retrieval in the brain, has been widely studied in the field of chaotic neural networks.[13–15] Recent studies have shown that different HNN models can exhibit various dynamic behaviors, such as chaos,[16–18] hyperchaos,[19–21] transient chaos,[22–24] and hidden chaos,[25–27] as well as special coexistence states, including multistability,[28,29] extreme multistability,[30,31] and others. Bao *et al.*[32] proposed a two-dimensional discrete HNN and discovered the coexistence of Neimark–Sacker bifurcation behavior induced by stability evolution and hyperchaotic attractors. Chen *et al.*[33] replaced the hyperbolic tangent activation function of a two-neuron HNN with the ReLU function, revealing the existence of periodic bifurcations, spike/burst patterns, fast–slow effects, and coexistence of multistable modes. Peng *et al.*[34] demonstrated that applying time-varying stimuli to the HNN can alter the number of generated vortices or disrupt its dynamic behavior.

Memristors, the fourth fundamental electronic component, exhibit powerful nonlinear characteristics and offer

†Corresponding author. E-mail: jj67123@hnust.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

unique advantages in nonlinear dynamics research, including applications in neural networks,[35–38] chaotic systems,[39–41] and memory storage.[42,43] In particular, in the field of neural networks, memristors can store effective data by adjusting their internal parameters, exhibiting memory properties and enabling a closer approximation to biological synapses.[44–46] The integration of memristors with HNN enhances the bio-inspired capabilities of the HNN, allowing it to better simulate the complex dynamic behaviors generated by the brain.[47] In biological neural networks, synapses exhibit dynamic plasticity, where the synaptic connection strength changes over time to enable neurons to better respond to external stimuli. Memristive synapses mimic this process by utilizing the change in resistance of memristors in a circuit to achieve similar learning and adaptive functions. Unlike traditional electronic components such as capacitors and inductors, memristors have the ability to "remember" their historical state, giving them a unique advantage when simulating neural networks. The simplified HNN diagram with the constructed self-synapse is shown in Fig. 1. Mou *et al.*[48] employed discrete capacitive memristors as synaptic couplings in a discrete neuron mapping model, discovering multiple discharge modes and extreme multistability in their model. Xu *et al.*[49] constructed a locally active memristor based on the rectified linear unit (ReLU) function and applied it to a third-order circuit, finding that the circuit could generate periodic/chaotic spike behaviors. Sun *et al.*[50] proposed a locally active magnetically controlled hyperbolic memristor coupled with a four-dimensional HNN, which produced complex dynamic behaviors. Lin *et al.*[51] replaced the synapses of the HNN with piecewise memristors, obtaining an arbitrary number of multi-structure chaotic attractors through position variation and parameter control. To achieve the interaction between the electric field and the magnetic field, Zhang *et al.*[52] used flux-controlled memristors as biomimetic synapses to connect a three-neuron HNN, systematically proposing three topological brain-like functional area networks, including the memristor-cascaded dual HNN, chain HNN, and lattice HNN. Meanwhile, to simulate synaptic plasticity and self-regulation, the authors also proposed an adaptive regulation scheme that controls memristor synapses through the energy difference between adjacent HNNs.

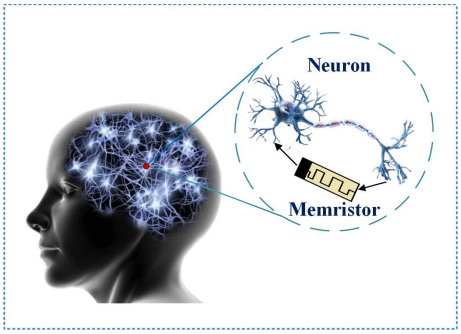


Fig. 1. Simplified schematic diagram of the self-synaptic HNN.

The topological complexity of attractors is closely related to their high unpredictability. In recent years, multi-scroll attractors, compared with simple single-scroll and double-scroll attractors, have exhibited more complex key spaces and stronger randomness, gradually becoming a research focus in neural networks. Yu *et al.*[53] proposed a non-polynomial memristor that simulates neuron synapses or electromagnetic radiation, satisfies the Lipschitz condition, and can control unidirectional or multidirectional multi-scroll attractors. Zhang *et al.*[54] coupled a magnetically controlled hyperbolic memristor with an HNN and discovered that the system could generate an arbitrary number of multi-double-scroll chaotic attractors in multiple directions. Li *et al.*[55] introduced a multi-level logic pulse into the memristor, which extended the multi-scroll behavior of the original HNN model into a symmetric structure. Yu *et al.*[56] demonstrated that by introducing different numbers of memristors, the system could produce multi-dimensional chaotic attractors with an arbitrary number of multi-double-scroll structures and applied the generated sequences to industrial internet of things (IIoT) video encryption, further enhancing communication security.

Transient chaos refers to a dynamical phenomenon in which a system exhibits chaotic-like behavior for a finite period before eventually converging to a periodic orbit or a steady-state point. It typically arises from boundary crises and represents a highly specific and intricate state in chaotic dynamics.[57] In neural networks, this behavior is characterized by an initial phase of highly fluctuating chaotic signals that eventually stabilize into periodic firing patterns. In brain-inspired neural networks, such dynamical phenomena are often associated with abnormal brain activity.[58] Therefore, investigating these dynamic characteristics is essential. Bao *et al.*[59] proposed a system that simulates a transient chaotic neural network (TCNN) using memristors to solve the traveling salesman problem (TSP). Compared with previous mixed analog-digital systems, the proposed design achieved an eightfold reduction in total energy consumption for continuous optimization tasks. Manchein *et al.*[60] investigated the dynamical properties of FitzHugh–Nagumo (FHN) neurons under oscillatory current drive and Gaussian noise perturbations of varying intensities. By tuning the parameter space, they observed transient chaotic behavior near the boundary between periodic and chaotic regimes, demonstrating that noise-induced perturbations can elicit more complex nonlinear dynamics in the FHN model. Li *et al.*[61] studied a non-cascaded abnormal neural network model with three neurons. By varying a single synaptic weight, they identified the presence of limit cycles, single-scroll chaotic attractors, and double-scroll attractors. Further adjustment of two synaptic weights revealed the coexistence of chaotic and periodic attractors, burst ignition, and long-term transient chaos, highlighting the rich dynamical complexity of their model.

However, studies on the coexistence of multi-scroll attractors and transient chaotic attractors induced by self-feedback synapses within a single model have not yet been reported. In this work, we investigate the dynamical behavior of a four-neuron Hopfield neural network in which the self-feedback synapse is replaced by a memristor. The main contributions are as follows:

(i) A novel cosine memristor capable of generating multi-scroll attractors is proposed. Unlike most memristors with piecewise functions, this memristor is continuous and smooth.

(ii) A cosine-based memristor is introduced to replace the self-synapse of neuron 1. Through analysis, it is revealed that the system possesses infinitely many equilibrium points and is capable of generating rare dynamical phenomena such as multi-transient behaviors, multi-scroll attractors, and complex coexistence of multiple attractors.

(iii) A novel image encryption algorithm based on Electronic Codebook mode is designed, with the information entropy approaching the ideal value, demonstrating the excellent randomness of the chaotic sequences generated by the system.

(iv) Field-programmable gate array (FPGA) technology is employed to implement the proposed self-synapse HNN, validating the feasibility of the theoretical model.

In Section 2, we compare this study with previous related work. In Section 3, we introduce the proposed cosine memristor model and analyze its equilibrium points. Section 3 then provides a detailed analysis of the dynamic behavior exhibited by the self-synapse HNN system. In Section 4, we outline the implementation of the proposed self-synapse HNN system on an FPGA, validating its theoretical feasibility. Section 5 presents an image encryption method based on the electronic codebook (ECB) approach. Finally, Section 6 concludes the paper.

2. Related works

This subsection compares the present study with previous related works, highlighting its advantages in three main aspects: the dynamic behaviors induced by the design of the memristive neural network, hardware implementation, and image encryption.

2.1. System comparison

As shown in Table 1, under the same continuous cosine memristor, the self-synaptic neural network proposed in this study exhibits significantly richer dynamic behaviors compared with previous models. For example, the transient coexistence phenomenon does not appear in the comparative literature. Moreover, unlike earlier works, the hardware implementation in this study is carried out using FPGA-based digital circuits. Compared with software-based circuit simulations such as Circuit and Multisim, FPGA-based digital circuit implementations are closer to real application scenarios. They not only validate the feasibility of theoretical models but also support large-scale parallel computation, offering higher engineering practicality and greater research value.

Table 1. Comparison of systems with different memristor models.

Properties	2022[62]	2023[63]	2024[64]	This paper
Continuity	yes	yes	yes	yes
Multiscroll attractor	no	no	no	yes
Special dynamic behavior	no	no	no	transient coexistence
Hardware implementation	no	circuit	multisim	FPGA

Table 2. Comparison of image encryption algorithms across representative studies.

Category	Expert systems with applications[66]	IEEE access[65]	Information sciences[67]	This work
Encryption algorithm	An encryption scheme based on a hyperchaotic system	A color image encryption algorithm combining KAA mapping and multiple chaotic mapping	Chaos-based image encryption combined with compressed sensing	Block cipher image encryption based on ECB mode
NPCR/UACI (%)	99.6101/33.4751	99.6254/33.0704	99.6085/33.4776	99.6399/33.6029
Encryption time (s)	0.2656	2.63711	0.0475	0.003526
Robustness analysis	Central cropping	No	Noise attack	Gaussian + salt-and-pepper + central + bordercropping
Advantages	Novel framework to generate chaotic seeds	The near-null pixel correlation coefficients of the encrypted images as well as the very large key space	Efficient in data quality and computation	High efficiency with strong security and decryption quality
Future directions	Tailored encryption for domain-specific images	No	Extend to video encryption	Real-time video secure transmission field, autonomous driving secure communication

2.2. Comparison of image encryption effects

In recent years, image encryption algorithms based on chaotic systems have made significant progress. To comprehensively evaluate the innovation and advancement of block cipher-based image encryption algorithms under the ECB mode, this paper selects and systematically compares three representative and relevant studies in the field, as detailed in Table 2. Reference [65] proposes a color image encryption algorithm by integrating the KAA map with multiple chaotic maps, resulting in a large key space and strong resilience against various types of attacks, thus validating its robust performance. The algorithm in Ref. [66] is based on a 3D discrete hyperchaotic system and enhances resistance to known-plaintext attacks by modifying the LSBs of randomly selected pixels in the plaintext image. However, it suffers from excessive encryption time. In contrast, Ref. [67] introduces a data segmentation and reconstruction method based on circulant matrices, which significantly reduces encryption time and offers a promising approach for video encryption applications. Under identical image encryption conditions, the proposed algorithm achieves a processing time of only 3.526 ms using the ECB-mode block cipher, which is 13.5 times faster than the 47.5 ms reported in Ref. [67], representing a 93% improvement in encryption efficiency. This breakthrough in computational efficiency highlights the potential of the proposed algorithm for real-time image encryption tasks and underscores its applicability in critical areas such as secure video transmission and autonomous driving communication systems.

3. Model construction and analysis

3.1. Memristor model

Based on memristor theory, the general mathematical model of a non-ideal magnetically controlled memristor can be expressed as

$$\begin{cases} i = W(\varphi)v, \\ \dot{\varphi} = f(\varphi, v). \end{cases} \tag{1}$$

In the equation, v, i, and φ represent the voltage across the memristor, the current passing through the memristor, and the magnetic flux variable, respectively. $W(\varphi)$ and $f(\varphi, v)$ denote the memductance function and the state equation. According to existing research, memristors can be used to simulate the sub-synapses of neurons. Based on Eq. (1), a new non-ideal magnetically controlled memristor model is proposed, and its expression is given by

$$\begin{cases} i = W(\varphi)v = (a + b\varphi)v, \\ \dot{\varphi} = cv - d\cos(\pi\varphi). \end{cases} \tag{2}$$

In this model, a, b, c, and d are the four adjustable parameters of the memristor. A sinusoidal voltage source $V_m = A\sin(Ft)$ is applied to the input port of the memristor, where A and F represent the amplitude and frequency, respectively. Fixing the parameters of the memristor model at $a = 0.3$, $b = 0.004$, $c = 2.1$, and $d = 1.6$, we first choose the frequency $F = 1$. When $A = 20$, 15, and 8, the v–i curves of the memristor are shown in Fig. 2. As A decreases, the area of the hysteresis side lobes increases. When the amplitude is set to $A = 10$, and the frequency is varied as $F = 1$, 3, and 15, the compressed hysteresis loops near the origin gradually shrink for increasing frequency. As the frequency increases, the side lobe area diminishes and eventually approaches a single-valued function. Thus, the proposed memristor is fully capable of exhibiting the essential characteristics of a memristor.

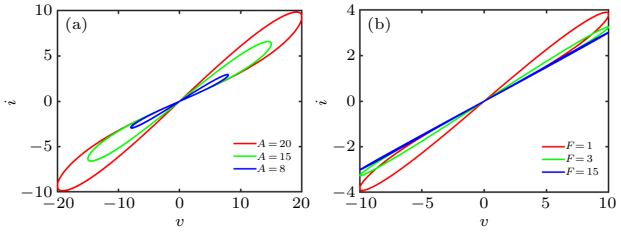


Fig. 2. Pinched hysteresis loops of the memristor. (a) Frequency $F = 1$, (b) amplitude $A = 10$.

3.2. HNN model with autosynaptic memristor

The HNN, as an efficient artificial neural network, features a complex interconnected structure similar to that of the biological nervous system. The classical HNN model, consisting of n neurons, is described by the following set of differential equations:

$$C_i\dot{x}_i = -\frac{x_i}{R_i} + \sum_{j=1}^{n} w_{ij}\tanh(x_j) + I_i, \qquad (i, j \in \mathbb{N}^*). \tag{3}$$

In this model, C_i and R_i denote the capacitance and resistance associated with neuron i. The coefficient w_{ij} represents the synaptic weight describing the connection strength from neuron j to neuron i, and $\tanh(\cdot)$ is the neuronal activation function. I_i denotes the external input current. Memristors are widely employed to emulate biological synapses. As shown in Fig. 3, the proposed memristor model is incorporated into a four-neuron HNN, where the memristor is used to simulate the autosynapse of neuron 1, yielding a memristive HNN model.

Based on the original HNN formulation (3), we set $C_i = 1$, $R_i = 1$, and $I_i = 0$ ($i = 1, 2, 3$), and choose appropriate synaptic weights to obtain the following four-neuron system:

$$\begin{cases} \dot{x} &= -x + k_1(a_1 + b_1 w)\tanh(x) - 1.6\tanh(y) \\ &\quad +0.75\tanh(z) - 20\tanh(u), \\ \dot{y} &= -y + 2.5\tanh(x) + 2\tanh(y) + 0.95\tanh(z) \\ &\quad +5\tanh(u), \\ \dot{z} &= -z - 0.5\tanh(x) + 3.8\tanh(y) + 0.2\tanh(z) \\ &\quad -6.5\tanh(u), \\ \dot{u} &= -u + 2\tanh(x) + 2.9\tanh(z) + 4\tanh(u), \\ \dot{w} &= c_1\tanh(x) - d_1\cos(\pi w). \end{cases} \tag{4}$$

Here, k_1 represents the coupling strength introduced by the memristor, while a_1, b_1, c_1, and d_1 are adjustable parameters of the memristive synapse.

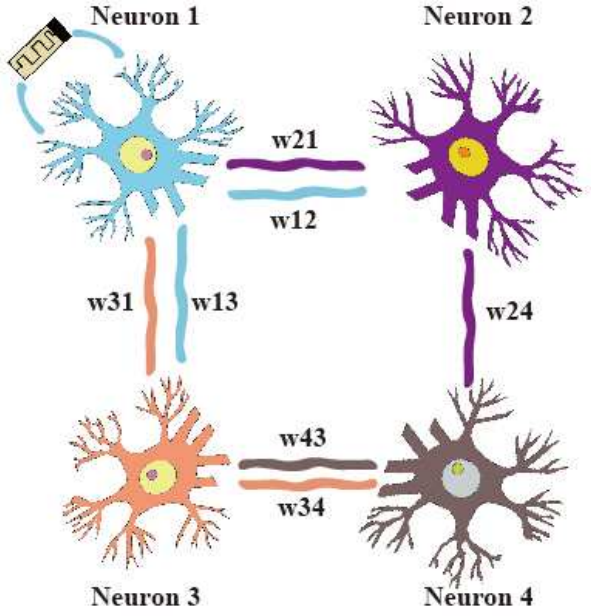


Fig. 3. Four-dimensional autosynaptic HNN topology.

3.3. Equilibrium and stability analysis

The chaotic behavior of the system is closely associated with the equilibrium points and their stability in the HNN. Therefore, the equilibrium characteristics of Eq. (4) are analyzed using both theoretical and numerical methods. Let $E = (x^*, y^*, z^*, u^*, w^*)$ denote an equilibrium point of this memristive HNN. By setting $\dot{x} = \dot{y} = \dot{z} = \dot{u} = \dot{w} = 0$, the equilibrium points are obtained by solving the following system:

$$\begin{cases} -x + k_1(a_1 + b_1 w)\tanh(x) - 1.6\tanh(y) + 0.75\tanh(z) \\ \quad -20\tanh(u) = 0, \\ -y + 2.5\tanh(x) + 2\tanh(y) + 0.95\tanh(z) \\ \quad +5\tanh(u) = 0, \\ -z - 0.5\tanh(x) + 3.8\tanh(y) + 0.2\tanh(z) \\ \quad -6.5\tanh(u) = 0, \\ -u + 2\tanh(x) + 2.9\tanh(z) + 4\tanh(u) = 0, \\ c_1\tanh(x) - d_1\cos(\pi w) = 0. \end{cases} \tag{5}$$

To analyze the equilibrium states and their stability, both theoretical reasoning and numerical computation are applied to Eq. (5). The first four equations yield $x^* = y^* = z^* = u^* = 0$. Using MATLAB, the curves corresponding to $x = 0$ and $\cos(\pi w)$ are plotted for graphical inspection, as shown in Fig. 4. The intersection points of these curves give $w^* = \frac{1}{2} \pm k$, $(k \in \mathbb{Z})$.

Thus, the system possesses infinitely many equilibrium points, expressed as

$$E = (0, 0, 0, 0, \tfrac{1}{2} \pm k), \qquad (k \in \mathbb{Z}).$$

This result indicates that if each equilibrium point is unstable, the model can generate self-excited multi-scroll attractors whose number and configuration depend on the distribution of these equilibrium points.

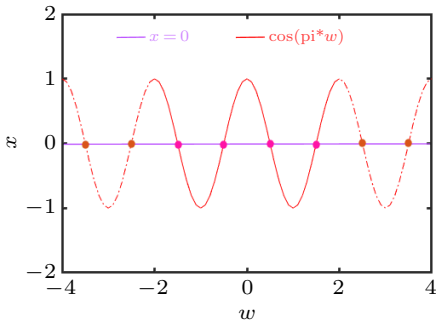


Fig. 4. Equilibrium point schematic diagram.

Taking the single-scroll chaotic attractor as an example, this study investigates the stability of the equilibrium point. The parameters are set as $a_1 = 1.5$, $b_1 = 0.02$, $c_1 = 1$, $d_1 = 1$, and $k_1 = 1.3$. The initial condition is given by $[0, 0, 0, 0, \frac{1}{2}]$. Setting the left-hand side of the system to zero, the Jacobian matrix of the model is expressed as follows:

$$\begin{bmatrix} -1 + k_1^2 n & -1.6k_2 & 0.75k_3 & -20k_4 & b_1 k_1 \tanh(x) \\ 2.5k_1 & -1 + 2k_2 & 0.95k_3 & 5k_4 & 0 \\ -0.5k_1 & 3.8k_2 & -1 + 0.2k_3 & -6.5k_4 & 0 \\ 2k_1 & 0 & 2.9k_3 & -1 + 4k_4 & 0 \\ c_1 k_1 & 0 & 0 & 0 & d_1 \pi \sin(\pi w) \end{bmatrix}, \tag{6}$$

where $n = (a_1 + b_1 w)$, $k_1 = \mathrm{sech}^2(x)$, $k_2 = \mathrm{sech}^2(y)$, $k_3 = \mathrm{sech}^2(z)$, $k_4 = \mathrm{sech}^2(u)$, $k_5 = \mathrm{sech}^2(w)$. After substituting the parameters and initial conditions, the characteristic equation is given by

$$\begin{aligned} \det(A - \lambda I) = {} & \lambda^5 - (4.16 + \pi)\lambda^4 + (62.49 + 4.16\pi)\lambda^3 \\ & - (104.63 + 62.49\pi)\lambda^2 \\ & + (495.84 + 104.63\pi)\lambda - 495.84\pi. \end{aligned} \tag{7}$$

The final eigenvalues are $\lambda_1 = 3.1416$, $\lambda_{2,3} = 1.2640 \pm 6.8150\mathrm{i}$, $\lambda_{4,5} = 0.8175 \pm 3.1069\mathrm{i}$. It can be observed that the real parts of all eigenvalues are positive, indicating that the corresponding equilibrium point is unstable. The subsequent multi-scroll phenomena in the system arise from the periodic characteristics of the cosine memristor, which lead to the existence of infinitely many unstable equilibrium points. As a result, the system can spontaneously generate chaotic attractors.

4. Dynamic analysis

To analyze the dynamic characteristics of the proposed self-synaptic HNN, various dynamical tools are employed, including the Lyapunov exponent spectrum, bifurcation diagrams, attractor phase plots, and time-domain plots. Numerical simulations performed in Matlab reveal that the system exhibits a diverse range of distinct dynamical behaviors.

4.1. Dynamic analysis of the coupling strength k_1

In this subsection, we set the parameters as $a_1 = 1.5$, $b_1 = 0.02$, $c_1 = 1$, $d_1 = 1$, and $k_1 = 1.3$, with initial conditions $[1, 0, 0, 0.1, \pi/2]$. The resulting chaotic attractor phase diagram is shown in Fig. 5.

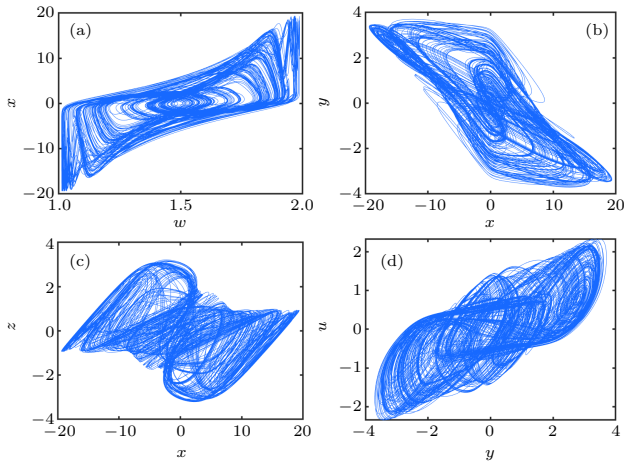


Fig. 5. Chaotic attractors of different planes. (a) w–x, (b) x–y, (c) x–z, and (d) y–u.

We perform a detailed dynamic behavior analysis using the coupling strength k_1 as the primary control parameter. The calculated Lyapunov exponent spectrum and the corresponding bifurcation diagram are presented in Fig. 6. Keeping the other system parameters unchanged, when k_1 varies within the range $[1.2, 3]$, the following observations can be made:

(i) In the range $[1.2, 2.38]$, the largest Lyapunov exponent is positive, while the remaining exponents are zero or negative, indicating that the system is in a chaotic state with sensitive dependence on initial conditions.

(ii) In the range $[2.38, 2.65]$, the largest Lyapunov exponent becomes zero, and the remaining exponents are negative, implying that the system exhibits stable periodic behavior with well-defined oscillations.

(iii) In the range $[2.65, 3]$, the largest Lyapunov exponent becomes positive again, while the remaining exponents remain zero or negative, confirming that the system returns to a chaotic state with complex dynamics and unpredictable trajectories.

The corresponding bifurcation diagram, shown in Fig. 6(b), is consistent with the Lyapunov exponent spectrum, providing strong evidence of the system's rich and varied dynamic behaviors over the specified range of k_1.

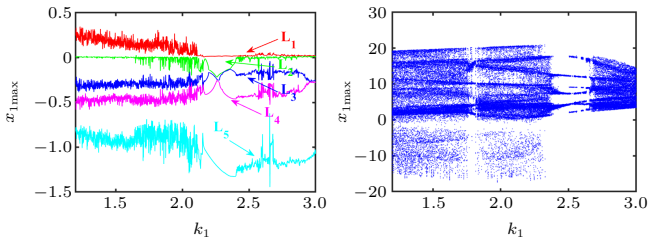


Fig. 6. The dynamics of parameter k_1. (a) Lyapunov exponent spectrum, (b) Bifurcation diagram.

4.2. Transient attractor

Transient chaos refers to a type of chaotic behavior that persists for a limited time before the system eventually transitions to a stable periodic state. This fascinating phenomenon typically arises from a boundary crisis in nonlinear dynamical systems. The conditions for transient chaos in nonlinear systems are extremely stringent; it occurs only when the boundary between an attractor and its basin of attraction completely vanishes. Consequently, transient chaos is regarded as a rare and highly intricate phenomenon in chaotic dynamics.

During the detailed numerical simulation with k_1 as the primary control parameter, multiple instances of transient chaos were observed, a phenomenon that is extremely rare in typical chaotic systems. With the other parameters held constant, the time-domain plots and attractor phase portraits corresponding to different values of k_1 are shown in Fig. 7. The figures clearly indicate that when $k_1 = 1.27$, a distinctive instance of transient chaos emerges. The trajectory initially exhibits strongly chaotic behavior, followed by a decaying periodic signal in the latter part. This remarkable phenomenon has not been previously reported in self-synaptic neural networks.

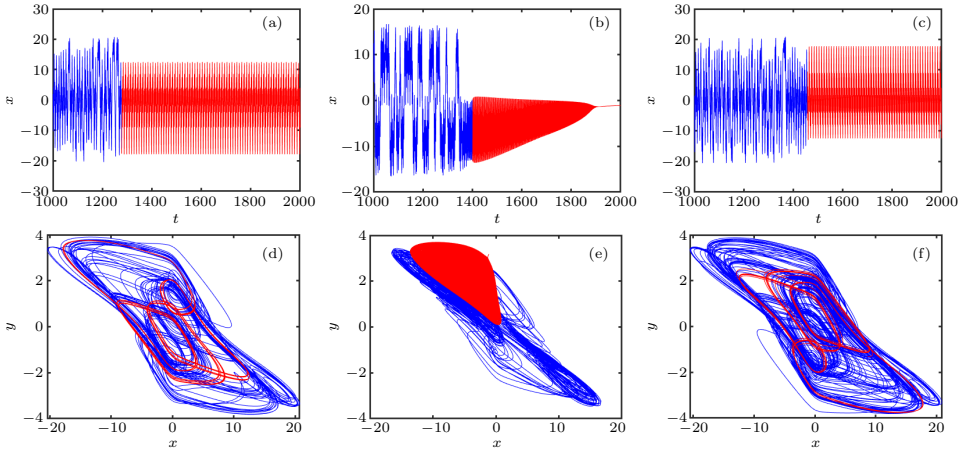


Fig. 7. Time-domain waveforms and attractor plots for different k_1 values. (a)–(c) Time-domain waveforms with $k_1 = 1.2, 1.27, 1.36$, respectively; (d)–(f) corresponding attractor plots for $k_1 = 1.2, 1.27, 1.36$, respectively.

4.3. Coexisting attractor

Coexisting attractors are defined as distinct attractors that arise in a chaotic system under identical parameter settings but with different initial conditions. Using the coupling strength k_1 as a dynamical parameter, with initial values $(1,0,0,0.1,\frac{\pi}{2})$ (red) and $(-1,0,0,0.1,\frac{\pi}{2})$ (blue), and keeping the parameter values $a_1 = 1.5$, $b_1 = 0.02$, $c_1 = 1$, and $d_1 = 1$ constant, the coexistence bifurcation diagram, as shown in Fig. 8, clearly demonstrates the coexistence behaviors. The corresponding values of k_1 and the coexistence dynamical phenomena are presented in Table 3 and Fig. 9. It can be observed that the system exhibits coexistence of chaos and transient chaos, periodic and periodic, periodic and chaotic, as well as chaotic and chaotic attractors, depending on the value of k_1. Hence, the proposed self-synaptic memristive neural network exhibits multistability, highlighting its superior chaotic characteristics. From the perspective of image encryption, attractor coexistence enhances sensitivity to initial conditions, thereby improving key sensitivity and rendering the encryption sequence highly unpredictable.

4.4. Multi-scroll attractor

In the system constructed in this study, multi-scroll attractors manifest as multiple non-overlapping chaotic scroll structures alternately distributed in the phase space. The system trajectories can transition aperiodically among these vortices, exhibiting greater complexity than single-scroll attractors. Numerical simulations indicate that system (4) can generate multi-scroll attractors with an arbitrary number of scrolls. Specifically, we fix the parameters $a_1 = 1.5$, $b_1 = 0.02$, $k_1 = 1.2$, and the initial condition as $(1,0,0,0.1,\frac{\pi}{2})$. The w–x plane phase portraits of chaotic attractors, shown in Fig. 10, indicate that n-scroll chaotic attractors can be induced by adjusting c_1 and d_1. Numerical simulations further demonstrate that varying these parameters allows the system to generate a

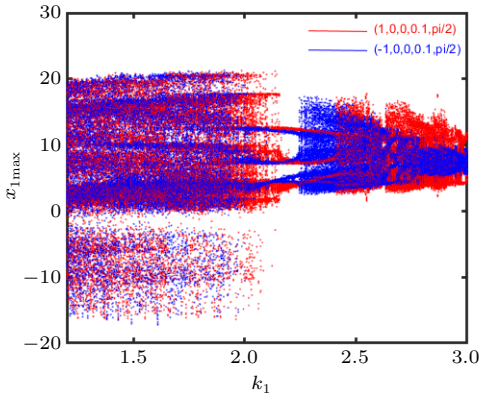


Fig. 8. Bifurcation diagram with different initial values.

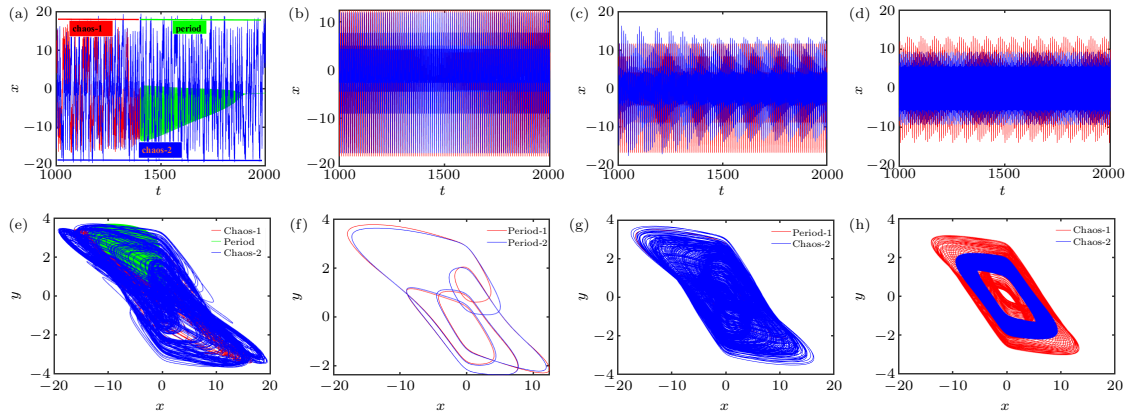


Fig. 9. Coexisting time domain waveform diagram and coexisting attractor phase diagram with different k_1 values. (a) $k_1 = 1.27$, (b) $k_1 = 2.2$, (c) $k_1 = 2.4$, (d) $k_1 = 2.8$, (e) $k_1 = 1.2$, (f) $k_1 = 2.2$, (g) $k_1 = 2.4$, (h) $k_1 = 2.8$.

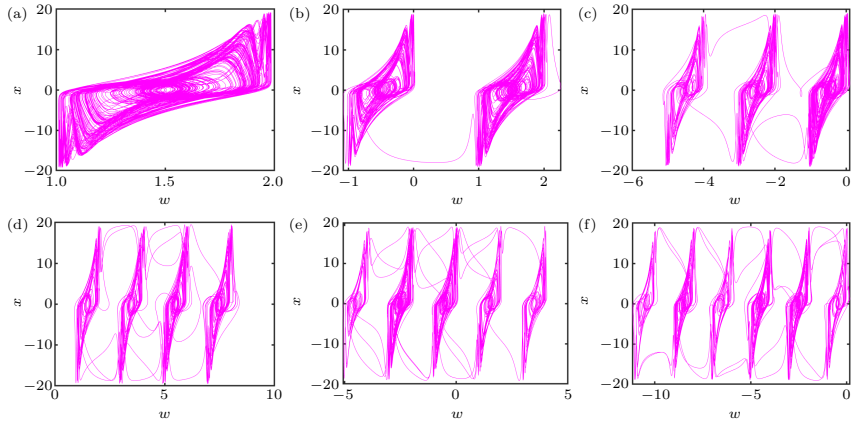


Fig. 10. Multi-scroll attractors in system (4) with 1–6 scrolls: (a) 1, (b) 2, (c) 3, (d) 4, (e) 5, (f) 6.

Table 3. Dynamical behavior and coexistence of attractors for different k_1 values.

k_1	Dynamical behavior induced by initial conditions		Type of
Value	$(1,0,0,0.1,\frac{\pi}{2})$	$(-1,0,0,0.1,\frac{\pi}{2})$	attractor
1.2	transient	chaos	chaos and transient chaos
1.27	transient chaos (periodic decay)	chaos	chaos and transient chaos
2.2	periodic	periodic	periodic and periodic
2.4	periodic	chaos	periodic and chaos
2.8	chaos	chaos	chaos and chaos

controllable number of scroll attractors. Figure 10 presents attractors ranging from 1 to 6 scrolls, highlighting the distinctive chaotic features of system (4).

5. FPGA implementation

Digital circuit technology has been continuously evolving, and in recent years, FPGA has been increasingly utilized in chaos research due to its high precision and strong parallel computation capabilities. These characteristics make FPGA an ideal platform for implementing chaotic digital circuits. In this study, we employ FPGA technology to implement the proposed self-synaptic neural network system. The FPGA-based digital circuit hardware block diagram is shown in Fig. 11. The development board used in this study is the ALINX AX7020, equipped with the xc7z100ffg900II chip. The DAC used is the AN9767 model, and the oscilloscope model is the TBS1102C. The software development platform is Vivado 2018.3, and the floating-point standard adopted is IEEE 754.



Fig. 11. Block diagram of the hardware experiment.

Based on the system dynamics, we employ the fourth-order Runge–Kutta (RK-4) algorithm for numerical integration and discretization. The time step is set to 0.001, as shown in Fig. 12. The simulation process involves discretizing the system and iteratively solving it using the RK-4 method, ensuring accurate numerical integration.

The clock module, Clock_n, operates with a clock cycle of 108×20 ns, providing a stable clock signal. Five register modules employ a differential module to perform iterative calculations based on the RK-4 method. All numerical computations and simulations are carried out on the Vivado 2018.3 simulation platform. The final output is a bitstream file generated by the software and programmed into the FPGA development board. The computed values are sent to the DAC module, which outputs dual-channel chaotic signals. These signals are visualized on an oscilloscope to display the chaotic attractor phase diagram, thereby verifying the feasibility of the proposed system.

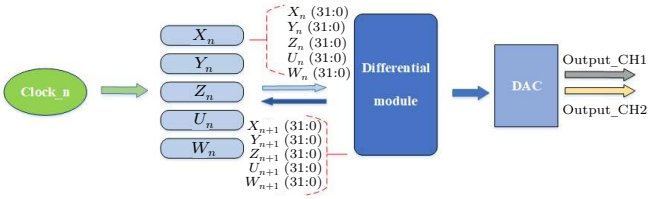


Fig. 12. Chaotic system implementation flow.

Through the above design process, we implemented the FPGA realization of the system outputs shown in Figs. 5 and 10, and the results are presented in Fig. 13. Figures 13(a)–13(c) show the phase diagrams of chaotic attractors in different planes, while Figs. 13(d)–13(i) present the evolution process of multi-scroll attractors. These results are consistent with the MATLAB simulation results, thereby verifying the feasibility and correctness of both the theoretical model and the FPGA implementation of the proposed self-synaptic neural network system.

In this study, the floating-point number format based on the IEEE 754 standard is employed. The Vivado 2018.3 simulation platform employs single-precision (32-bit) floating-point representation by default, whereas theoretical verification and MATLAB simulations utilize double-precision (64-bit) format. During the initial iterations, results from both 32-bit and 64-bit computations nearly coincide, and the local morphology of the attractors remains consistent, indicating a high degree of short-term agreement between the two representations.

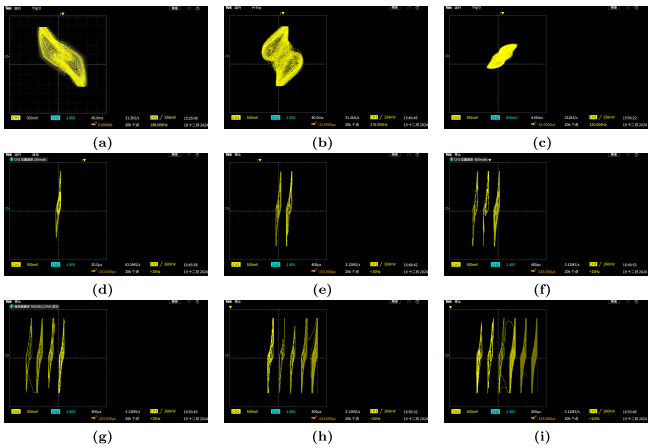


Fig. 13. Chaotic attractors in system (4) on FPGA. (a) x–y plane, (b) z–x plane, (c) y–u plane, (d) 1 scroll, (e) 2 scroll, (f) 3 scroll, (g) 4 scroll, (h) 5 scroll, (i) 6 scroll.

Table 4. Resource utilization of FPGA chip.

Resource	Utilization	Available	Utilization (%)
LUT	53,243	277,400	19.19
LUTRAM	2,423	108,200	2.24
FF	68,341	554,800	12.32
DSP	228	2,020	11.29
IO	35	362	9.67
BUFG	2	32	6.25
PLL	1	8	12.50

This phenomenon is also carefully validated through FPGA-based experiments: the FPGA implementation employs a 32-bit floating-point format, and the chaotic signals output by the DAC generate detailed attractor plots on the oscilloscope (Fig. 13) that are in excellent agreement with those obtained from MATLAB simulations (Figs. 5 and 10). This clearly demonstrates that the system exhibits a certain degree of robustness against floating-point errors, thereby effectively meeting the strict requirements for practical engineering implementations. The comprehensive resource consumption data of the FPGA implementation in this study are summarized in Table 4.

6. Image encryption

6.1. Encryption process

With the rapid development of informatization technologies, including the Internet of Things,[68] large-scale data center networks,[69,70] and advanced deep learning techniques,[71,72] information security has attracted increasing attention worldwide. Chaotic systems are particularly suitable for encryption due to their extreme sensitivity to initial conditions and the inherently non-periodic nature of their motion trajectories.[73–76] By employing the ECB mode encryption method, this paper proposes and implements an efficient image encryption algorithm based on the HNN chaotic sequence. The overall encryption process is illustrated in Fig. 14. The specific implementation steps are as follows:

$$\begin{cases} X = \mathrm{mod}\left(\mathrm{floor}\left((X_0+100)^*10^{10}\right), M^*N\right)+1, \\ Y = \mathrm{mod}\left(\mathrm{floor}\left((Y_0+100)^*10^{10}\right), 255\right)+1, \\ Z = \mathrm{mod}\left(\mathrm{floor}\left((Z_0+100)^*10^{10}\right), 255\right)+1, \\ U = \mathrm{mod}\left(\mathrm{floor}\left((U_0+100)^*10^{10}\right), 255\right)+1, \\ W = \mathrm{mod}\left(\mathrm{floor}\left((Z_0+100)^*10^{10}\right), 255\right)+1. \end{cases} \tag{8}$$

Step 1: Select a grayscale image A and record its dimensions as $M \times N$.

Step 2: According to the encryption requirements, the initial values $x_0, y_0, z_0, w_0, u_0, a_1, b_1, c_1, d_1, k_1$ of the HNN are set as the encryption key, and discretized to generate the chaotic sequences X_0, Y_0, Z_0, U_0, W_0.

Step 3: Compute the chaotic sequences X, Y, Z, U, W using Eq. (8).

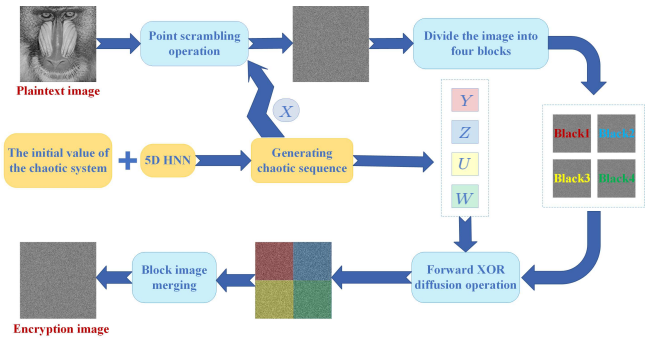


Fig. 14. Image encryption flow chart.

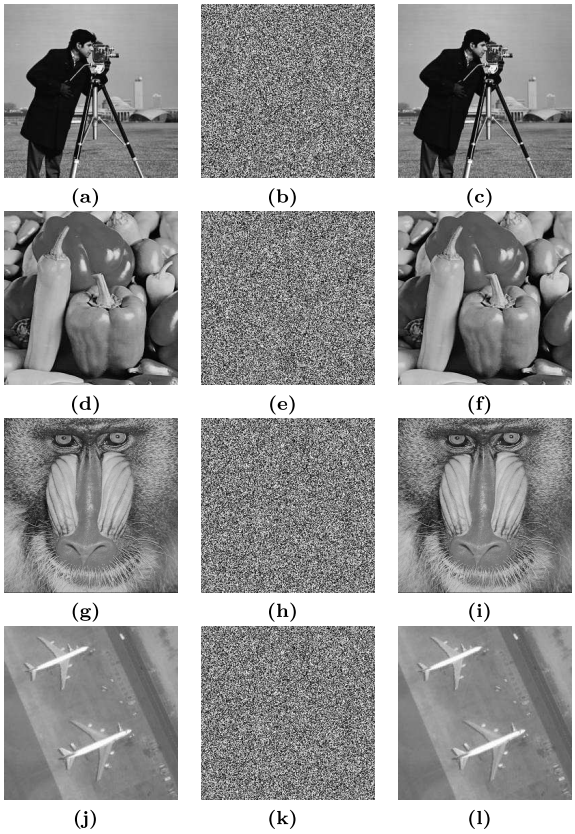


Fig. 15. Image encryption and decryption effect: (a), (d), (g), (j) is the plaintext image, (b), (e), (h), (k) is the encrypted image, and (c), (f), (i), (l) is the decrypted image.

Step 4: The chaotic sequence X is used to perform pixel scrambling on the grayscale image. The pixels of image A are rearranged according to Eq. (9). Subsequently, the image A is unfolded into a one-dimensional vector and divided into four equal-length blocks: block1, block2, block3, and block4.

$$A(i) \Leftrightarrow A(X(i)). \tag{9}$$

Step 5: Under the control of the encryption key, the four blocks — block1, block2, block3, and block4 — are each subjected to a forward XOR diffusion operation with different chaotic sequences Y, Z, U, W according to Eq. (10). Here, P is the one-dimensional vector of the plaintext image, and C and S are the cipher vectors. The encrypted output is the ciphertext image. Since this is a symmetric encryption scheme,

decryption is performed as the inverse of the encryption process.

$$C_i = C_{i-1} \oplus S_i \oplus P_i. \tag{10}$$

Four representative images were selected for testing: *Camerman.tif* (256×256, Fig. 15(a)), *Peppers.tif* (256×256, Fig. 15(d)), the remote sensing image *airplane.jpg* (256×256, Fig. 15(j)), and *Baboon.tiff* (512×512, Fig. 15(g)). The experiments were conducted on a platform consisting of Windows 11, an Intel Core i7-14700HX processor, 8 GB RAM, and MATLAB R2023b. As shown in Fig. 15, the proposed encryption scheme effectively performs both image encryption and decryption.

6.2. Histogram and variance analysis

The pixel distribution of an image can be analyzed through its histogram. Therefore, the more uniform the histogram of the encrypted image, the more difficult it becomes for an attacker to infer statistical information about the plaintext pixels. As shown in Fig. 16, the histogram curve of the plaintext image exhibits significant variations in height distribution, while the histogram of the encrypted image shows a more uniform distribution. These results indicate that the proposed encryption scheme can effectively resist plaintext attacks.

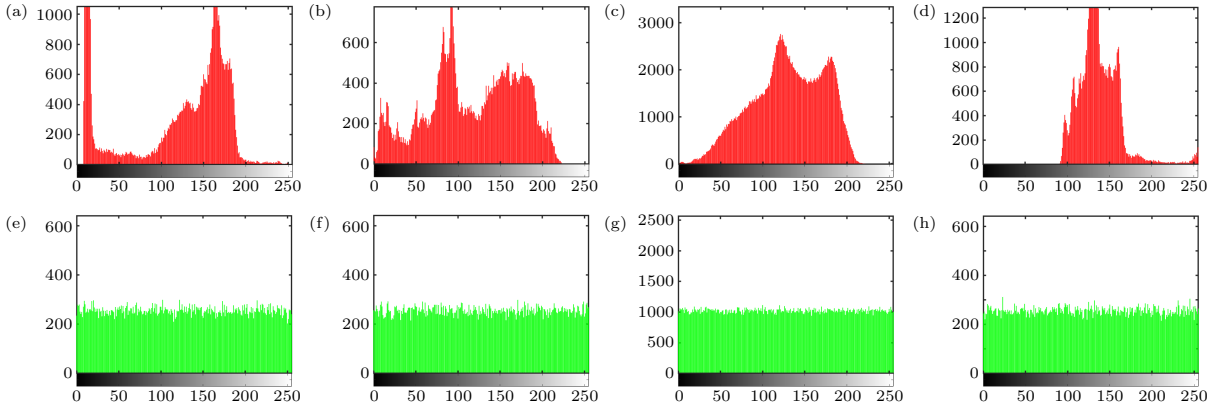


Fig. 16. Histograms before and after encryption: Cameraman before encryption (a) and after encryption (e), Peppers before encryption (b) and after encryption (f), Baboon before encryption (c) and after encryption (g), Airplane before encryption (d) and after encryption (h).

Meanwhile, the Chi-square test is employed to further evaluate whether the pixel value distribution of the encrypted image approximates an ideal random distribution. Table 5 presents the histogram variances of the plaintext and ciphertext images shown in Fig. 16. It is evident that the variance of the plaintext images is significantly larger, whereas the variance of the ciphertext images generated by the encryption algorithm in Fig. 14 is much smaller, indicating that the proposed algorithm achieves secure and reliable encryption performance.

Table 5. Histogram variance comparison between plaintext and encrypted images (Chi-square analysis).

Image type	Cameraman	Peppers	Baboon	Airplane
Plaintext image	1.10973×10^5	3.07745×10^4	1.87357×10^5	2.00033×10^5
Encryption image	281.0547	289.9141	269.7500	231.8359

6.3. Correlation analysis

Correlation analysis of adjacent pixels after image encryption can evaluate the effectiveness of the encryption algorithm in protecting the image's local features. High correlation in pixel distribution could allow an attacker to obtain information about the plaintext image. Therefore, to enhance the security and encryption performance, the correlation coefficient should be close to 0, indicating low correlation between adjacent pixels in the encrypted image. For the plaintext and ciphertext images, N pairs of adjacent pixels are selected, and their grayscale values are recorded as (u_i, v_i), where $i = 1, 2, \ldots, N$. The correlation coefficient between vectors $u = \{u_1, u_2, \ldots, u_N\}$ and $v = \{v_1, v_2, \ldots, v_N\}$ is computed as follows:

$$\begin{aligned} r_{sy} &= \frac{\operatorname{cov}(u,v)}{\sqrt{D(u)}\sqrt{D(v)}}, \\ \operatorname{cov}(u,v) &= \frac{1}{N}\sum_{j=1}^{z}\left(u_j - E(u)\right)\left(u_j - E(v)\right), \\ D(u) &= \frac{1}{N}\sum_{j=1}^{z}\left(u_j - E(u)\right)^2, \quad E(u) = \frac{1}{N}\sum_{j=1}^{z} u_j. \end{aligned} \tag{11}$$

As shown in Table 6, adjacent pixels of plaintext images exhibit high redundancy and strong correlation in various directions. In contrast, the correlation coefficients of encrypted images are close to 0, indicating that the encryption scheme effectively eliminates correlations in the original image. By applying Eq. (11) and using MATLAB, the adjacent pixel values of the "Camerman" plaintext and encrypted images in the horizontal, vertical, positive diagonal, and anti-diagonal directions were obtained. As shown in Fig. 17, the pixel distribution of the plaintext image is concentrated along the diagonals,

while after encryption, the pixel distribution of the encrypted image becomes nearly uniform across the entire plane.

6.4. Time efficiency analysis

Time complexity reflects how the computational cost of the encryption algorithm scales with increasing data size. Owing to the relatively simple encryption process of the ECB mode and its inherent ability to process multiple data blocks in parallel, the encryption time is comparatively short, making this scheme particularly suitable for large-scale image data. Table 7 presents the encryption efficiency of this scheme for images of various sizes, highlighting its practical suitability.

Table 6. Pixel correlation coefficients of plaintext and encrypted images in four directions.

Direction	Image type	Cameraman	Peppers	Baboon	Airplane
Horizontal	plaintext image	0.9603	0.9604	0.7536	0.9726
	encrypted image	−0.0111	0.0056	0.0410	−0.0328
Vertical	plaintext image	0.9347	0.9494	0.8696	0.9772
	encrypted image	−0.0075	−0.0140	−0.0214	−0.0171
Positive	plaintext image	0.8945	0.9216	0.7229	0.9558
diagonal	encrypted image	0.0222	−0.0132	0.0077	−0.0009
Sidelong	plaintext image	0.9151	0.9441	0.7318	0.9627
diagonal	encrypted image	0.0026	−0.0595	−0.0105	0.0070

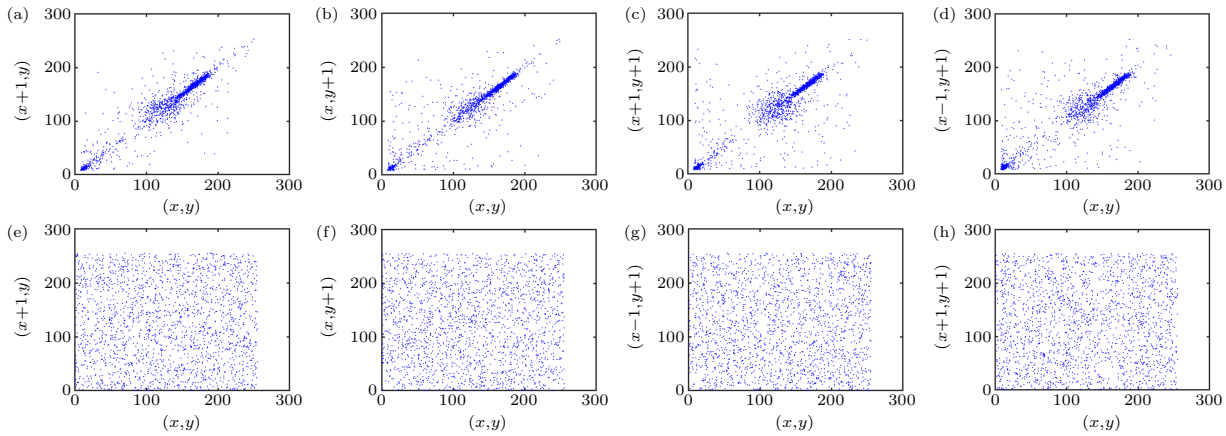


Fig. 17. Correlation distribution diagram: (a)–(d) show the correlation distribution of the plaintext image "Cameraman", and (e)–(h) show the correlation distribution of the corresponding ciphertext image.

Table 7. Comparison of encryption and decryption times.

Image/Reference	Decryption time (s)	Encryption time (s)
Camerman (proposed)	0.046748	0.005017
Peppers	0.046856	0.004681
Bamboo	0.068195	0.004952
Airplane	0.004516	0.003526
(Camerman)[66]	0.26560	0.07690
(Camerman)[67]	0.04440	0.04750

6.5. Differential attack analysis

To evaluate the sensitivity of the encryption system to key variations, quantitative methods are employed to compare the original image encrypted with the original key and the image encrypted with a slightly modified key. A larger difference between the two encrypted images indicates higher key sensitivity of the encryption system. The NPCR and UACI are computed using the following formulas:

$$\mathrm{NPCR}(P_1,P_2) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|\mathrm{Sign}\left(P_1(i,j)-P_2(i,j)\right)\right| \times 100\%, \tag{12}$$

$$\mathrm{UACI}(P_1,P_2) = \frac{1}{MN}\sum_{i}^{x}\sum_{j}^{z}\frac{|P_1(i,j)-P_2(i,j)|}{255}\times 100\%. \tag{13}$$

As shown in Table 8, when individual pixel values were sequentially modified by 10^{-12}, the resulting NPCR and UACI values closely approach their theoretical ideal values. Therefore, the encryption algorithm proposed in this study can effectively resist differential attacks.

Table 8. NPCR and UACI for different images.

Index		Camerman	Peppers	Bamboo	Airplane
x	NPCR	99.6399	99.5575	99.6273	99.585
	UACI	33.5371	33.6016	33.4718	33.5932
y	NPCR	99.6216	99.6231	99.6021	99.5804
	UACI	33.3788	33.6635	33.4935	33.6108
z	NPCR	99.6246	99.6170	99.5983	99.6063
	UACI	33.4983	33.3633	33.4775	33.6621
u	NPCR	99.6262	99.5361	99.6120	99.6399
	UACI	33.3008	33.4309	33.3860	33.5165
w	NPCR	99.5743	99.6399	99.6037	99.6094
	UACI	33.2865	33.4095	33.4244	33.6029

6.6. Robustness analysis under noise and data loss

Since image transmission is often imperfect, encrypted images may be affected by various types of attacks, resulting in data loss. Figure 18 illustrates the distribution of an encrypted image subjected to four types of attacks on the same plaintext image. Despite the presence of noise and cropping, the encrypted image maintains a high degree of visual integrity.

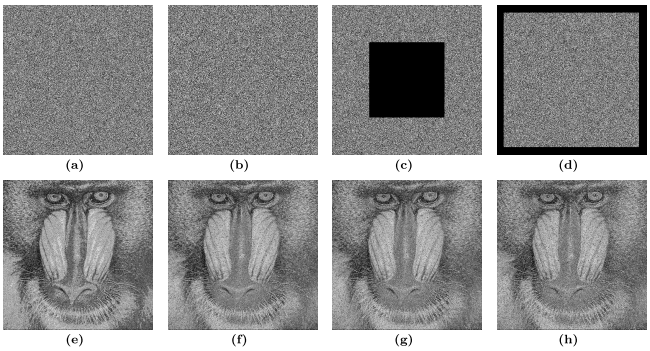


Fig. 18. Test results of data loss and robustness analysis: (a) Gaussian noise density 0.001, (b) salt-and-pepper noise density 0.1, (c) center cropping rate 1/2, (d) edge cropping rate 1/10. (e)–(h) The corresponding decrypted images.

6.7. Global entropy and local entropy analysis

Higher entropy in an encrypted image indicates greater uncertainty and complexity, making it more difficult for attackers to compromise the encryption via known-plaintext or statistical attacks. This reflects the increased security of the encryption algorithm.

Information entropy comprises both global entropy and local entropy, which differ in evaluation scales. Global entropy is calculated as a single value over all pixels in the entire image, reflecting the overall randomness of the encrypted image and satisfying Shannon's confusion criteria. Building on this, the image is divided into non-overlapping blocks of 44×44 pixels, and the local entropy is obtained by calculating the average entropy of these blocks. Local entropy assesses the statistical uniformity of localized regions, providing a more precise measure than global entropy. It serves as a complementary metric to global entropy, ensuring unpredictability at the local level.

The encrypted images were evaluated using both global and local entropy tests, and the results were compared with those of other algorithms. The experimental outcomes are presented in Tables 9 and 10. Consequently, the encrypted images produced by the proposed algorithm exhibit a high degree of uncertainty, providing robust resistance against entropy-based attacks.

Table 9. Comparison of global entropy values for encrypted image (Bamboo).

Image type	Cameraman	Peppers	Bamboo	Airplane	[77]	[78]	[79]
Encrypted image	7.9969	7.9967	7.9993	7.9971	7.9866	7.9992	7.9992

Table 10. Comparison of local entropy values for encrypted image (Camerman).

Image type	Camerman	Peppers	Bamboo	Airplane	[80]
Encrypted image	7.9007	7.9015	7.9024	7.9024	7.8093

Due to the parallel processing characteristics of the ECB mode and the synergistic effect with the rapid iteration capability of chaotic systems, the proposed chaotic image encryption algorithm based on ECB mode demonstrates significant advantages in encryption efficiency. Experimental results show that encrypting a 256×256 grayscale "Cameraman" image takes only 46.75 ms with the proposed method, compared to 265.6 ms reported in Ref. [66], achieving a 5.68-fold speed-up and an 82.4% improvement in efficiency. This substantial enhancement makes the algorithm particularly suitable for applications such as real-time video transmission and secure communication in autonomous driving systems. In terms of security, the algorithm exhibits strong performance and fault tolerance across multiple evaluation metrics. It significantly reduces the pixel correlation in the ciphertext and increases the information entropy, thereby enhancing resistance against various cryptographic attacks. These results suggest that the proposed method holds great potential for practical engineering applications.

6.8. Key space analysis

The system key is composed of initial conditions and parameter values, each offering a resolution of approximately 10^{15}. As a result, the total key space size can reach $10^{15} \approx 2^{498.29}$, indicating strong resistance against brute-force attacks. Typically, an attacker would need to search only half of the key space to successfully decrypt an image. According to Table 7, the average decryption time for a 256×256 grayscale image is approximately 0.004849 seconds. Given a key space of $2^{498.29}$, it would take an attacker approximately 7.688×10^{139} years to exhaust half of the key space. Evidently, for higher-resolution or larger images, the decryption complexity increases significantly, requiring even more time for brute-force attempts. Therefore, the proposed image encryption scheme demonstrates excellent performance in terms of key space size and computational security, effectively resisting both current and long-term brute-force attacks.

7. Conclusion

In this paper, a novel cosine memristor model is introduced to replace the self-synapse of a single neuron, resulting in the construction of a five-dimensional HNN system with infinitely many equilibrium points. Through dynamic analysis, it is found that the system not only exhibits multiple rare transient attractors but also generates an arbitrary number of multi-scroll attractors. Additionally, various types of chaos and periodicity coexist within the system. The attractors and multi-scroll attractors are further verified on the FPGA platform.

The HNN system with rich dynamical phenomena provides high-complexity and high-randomness sequences for image encryption. Several security analyses show that the encrypted images have information entropy values close to

the ideal, demonstrating excellent key sensitivity and anti-interference capabilities. The ECB block encryption mode, which processes multiple data blocks simultaneously, ensures high encryption efficiency for large-scale image processing. In conclusion, this paper presents a self-synapse neural network system that exhibits richer dynamic behavior compared to most known synaptic neural systems.

Nevertheless, this study has certain limitations. The research primarily relies on numerical simulations, with insufficient experimental validation. Future work should focus on experimentally verifying the effectiveness and reliability of the model. Potential research directions include extending the proposed encryption algorithm to applications in data protection and information security, as well as further investigating the properties of attractors and their roles in complex systems. This study aims to advance the field and explore more innovative applications.

Acknowledgments

This project was supported by the Guiding Science and Technology Plan Project of Changsha City under Grant kzd2501129, by the Natural Science Foundation of Hunan Province (Grant No. 2025JJ50368), the Scientific Research Fund of Hunan Provincial Education Department (Grant No. 24A0248), and the National Natural Science Foundation of China (Grant No. 62273141).

References

[1] Zhang Z, He Y, Mai W, Luo Y, Li X, Cheng Y, Huang X and Lin R 2025 *IEEE Trans. Neural Netw. Learn. Syst.* **36** 8166
[2] Deng Q, Wang C and Yang G 2025 *Int. J. Bifurcat. Chaos* **35** 2550053
[3] Xiang Q, Gong H and Hua C 2025 *J. Supercomput.* **81** 159
[4] Yu F, Wang X, Guo R, Ying Z, He Y and Zou Q 2025 *Chin. Phys. B* **34** 120501
[5] Mou J, Cao H, Zhou N and Cao Y 2024 *IEEE Trans. Cybern.* **54** 7333
[6] Zhu J, Jin J, Chen C, Wu L, Lu M and Ouyang A 2025 *IEEE Trans. Emerg. Top. Comput. Intell.* **9** 176
[7] Yu F, He S, Yao W, Cai S and Xu Q 2025 *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **44** 4564
[8] Korn H and Faure P 2003 *Comptes Rendus Biol.* **326** 787
[9] Tan X, Xiang C, Cao J, Xu W, Wen G and Rutkowski L 2021 *IEEE Trans. Cybern.* **52** 8246
[10] Calitoiu D, Oommen B J and Nussbaum D 2007 *IEEE Trans. Syst. Man Cybern. B Cybern.* **37** 692
[11] Deng Q, Wang C, Sun Y, Cong X, Lin H and Deng Z 2025 *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **44** 4701
[12] Peng B, Yan S, Cheng D, Yu D, Liu Z, Yakovlev V V, Yuan L and Chen X 2023 *Chin. Phys. Lett.* **40** 034201
[13] Yuan F, Qi Y, Yu X and Deng Y 2024 *Chaos Solitons Fractals* **188** 115478
[14] Wang C, Li Y and Deng Q 2025 *Chaos Solitons Fractals* **193** 116053
[15] Bao H, Chen Z, Ma J, Xu Q and Bao B 2024 *IEEE Trans. Ind. Electron.* **71** 16398
[16] Wan Q, Yang Q, Liu T, Chen C and Shen K 2024 *Chaos Solitons Fractals* **189** 115584
[17] Deng Q, Wang C, Sun Y and Yang G 2025 *Nonlinear Dyn.* **113** 17161
[18] Yu F, Gracia Y M, Guo R, Ying Z, Xu J, Yao W, Jin J and Lin H 2025 *Axioms* **14** 638
[19] Bao H, Su Y, Hua Z, Chen M, Xu Q and Bao B 2024 *IEEE Trans. Circuits Syst. I Regul. Pap.* **71** 4145
[20] Bao B, Tang H, Su Y, Bao H, Chen M and Xu Q 2024 *IEEE Trans. Circuits Syst. I Regul. Pap.* **71** 5907
[21] Bao H, Wang R, Tang H, Chen M and Bao B 2025 *IEEE Internet Things J.* **12** 20902
[22] Yao W, Fang J, Yu F, Xiong L, Tang L, Zhang J and Sun Y 2024 *Chaos* **34** 073149
[23] Lin H, Wang C and Tan Y 2020 *Nonlinear Dyn.* **99** 2369
[24] Wang P, Chen M, Xie Y, Pan C, Watanabe K, Taniguchi T, Cheng B, Liang S J and Miao F 2023 *Chin. Phys. Lett.* **40** 117201
[25] Luo D, Wang C, Deng Q and Sun Y 2025 *Nonlinear Dyn.* **113** 5811
[26] Pham V T, Jafari S, Vaidyanathan S, Volos C and Wang X 2016 *Sci. China Technol. Sci.* **59** 358
[27] Dudkowski D, Jafari S, Kapitaniak T, Kuznetsov N V, Leonov G A and Prasad A 2016 *Phys. Rep.* **637** 1
[28] He S, Yu F, Guo R, Zheng F, Tang T, Jie J and Wang C 2025 *Fractal Fract.* **9** 561
[29] Lin H, Deng X, Yu F and Sun Y 2024 *IEEE Internet Things J.* **11** 29878
[30] Lin H, Wang C, Yu F, Hong Q, Xu C and Sun Y 2023 *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **42** 4948
[31] Lin H, Deng X, Yu F and Sun Y 2025 *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **44** 304
[32] Bao B, Tang H, Bao H, Hua Z, Xu Q and Chen M 2025 *IEEE Trans. Ind. Electron.* **72** 4105
[33] Chen C, Min F, Zhang Y and Bao H 2023 *Chaos Solitons Fractals* **167** 113068
[34] Peng X, Li C, Zeng Y and Li C L 2024 *IEEE Trans. Circuits Syst. I Regul. Pap.* **71** 3242
[35] Lai Q, Wan Z, Zhang H and Chen G 2023 *IEEE Trans. Neural Netw. Learn. Syst.* **34** 7824
[36] Yu F, Su D, He S, Wu Y, Zhang S and Yin H 2025 *Chin. Phys. B* **34** 050502
[37] Yu F, He Ting, He S, Tan B, Shi C and Lin H 2025 *Int. J. Bifurcat. Chaos* **35** 2550146
[38] Zhang S, He D, Li Y, Lu D and Li C 2025 *IEEE Trans. Autom. Sci. Eng.* **22** 17828
[39] Deng Q, Wang C, Sun Y and Yang G 2025 *IEEE Trans. Circuits Syst. I Regul. Pap.* **72** 7993
[40] Deng Q, Wang C, Sun Y and Yang G 2025 *IEEE Trans. Cybern.* **55** 3926
[41] Yu F, Wu C, Xu S, Yao W, Xu C, Cai S and Wang C 2025 *Signal Image Video Process.* **19** 77
[42] Hong Q, Jiang H, Xiao P, Du S and Li T 2025 *IEEE Trans. Comput.* **74** 996
[43] Xiao P, Fang J, Wei Z, Dong Y, Du S, Wen S and Hong Q 2025 *IEEE Trans. Autom. Sci. Eng.* **22** 15163
[44] Yang F, Ma J and Wu F 2024 *Chaos Solitons Fractals* **187** 115361
[45] Wang C, Li Y, Yang G and Deng Q 2025 *Math.* **13** 1600
[46] Feng W, Zhang K, Zhang J, Zhao X, Chen Y, Cai B, Zhu Z, Wen H and Ye C 2025 *Fractal Fract.* **9** 426
[47] Li F, Qin W, Xi M, Bai L and Bao B 2025 *Neural Netw.* **183** 107049
[48] Mou J, Ma T, Banerjee S and Zhang Y 2024 *IEEE Trans. Circuits Syst. I: Reg. Pap.* **71** 1771
[49] Xu Q, Chen X, Wu H, Iu H H C, Parastesh F and Wang N 2024 *IEEE Trans. Circuits Syst. II: Express Briefs* **71** 4551
[50] Sun J, Li C, Wang Z and Wang Y 2023 *IEEE Trans. Ind. Informat.* **20** 3778
[51] Lin H, Wang C, Xu C, Zhang X and Iu H H 2022 *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **42** 942
[52] Zhang S, Wang C, Zhang H and Lin H 2024 *Chaos Solitons Fractals* **186** 115191
[53] Yu F, Kong X, Yao W, Zhang J, Cai S, Lin H and Jin J 2024 *Chaos Solitons Fractals* **179** 114440
[54] Zhang S, Chen C, Zhang Y, Cai J, Wang X and Zeng Z 2025 *IEEE Trans. Syst. Man Cybern.: Syst.* **55** 735
[55] Li J, Wang C and Deng Q 2024 *Nonlinear Dyn.* **112** 14463
[56] Yu F, Lin Y, Yao W, Cai S, Lin H and Li Y 2025 *Neural Netw.* **182** 106904
[57] Zhao H, Ma X and Bi Q 2024 *Int. J. Non-Linear Mech.* **159** 104592
[58] Vignesh D, Ma J and Banerjee S 2024 *Neurocomputing* **564** 126961
[59] Bao H, Ren P, Xu K, Yang L, Zhou H, Li J, Li Y and Miao X 2024 *IEEE Trans. Circuits Syst. I: Reg. Pap.* **71** 3708
[60] Manchein C, Santana L, da Silva R M and Beims M W 2022 *Chaos* **32**

[61] Li F, Chen Z, Zhang Y, Bai L and Bao B 2024 *AEU-Int. J. Electron. Commun.* **174** 155037
[62] Han X, Bi X, Sun B, Ren L and Xiong L 2022 *Front. Phys.* **10** 911144
[63] Dong K, Xu K, Wang L, Zhang C, Jin F and Song J 2023 *Nonlinear Dyn.* **111** 22013
[64] Sun J, Li C, Wang Y and Wang Z 2024 *Cogn. Neurodyn.* **18** 2975
[65] Alexan W, Elkandoz M, Mashaly M, Azab E and Aboshousha A 2023 *IEEE Access* **11** 11541
[66] Li L 2024 *Expert Syst. Appl.* **252** 124316
[67] Wang C and Song L 2023 *Inf. Sci.* **642** 119166
[68] Sevin A 2025 *Math.* **13** 734
[69] Hu J, Zhi R and Wang J 2025 *IEEE Trans. Ind. Informat.* textbf21 6495
[70] Hu J, Rao S, Zhu M, Huang J, Wang J and Wang J 2025 *IEEE Trans. Ind. Informat.* **21** 2799
[71] Alabadi M, Habbal A and Guizani M 2024 *IEEE Internet Things J.* **11** 20271
[72] Zhang J, Yang J, Liu Z and Wang J 2025 *Neurocomputing* **616** 128908
[73] Ning Y, Jin J, Li Z, Chen C and Ouyang A 2025 *Tsinghua Sci. Technol.*
[74] Feng W, Zhang J, Chen Y, Qin Z, Zhang Y, Ahmad M and Woźniak M 2024 *Expert Syst. Appl.* **246** 123190
[75] Jin J, Wu M, Ouyang A, Li K and Chen C 2025 *IEEE Internet Things J.* **12** 14297
[76] Gao S, Zhang Z, Iu H H C, Ding S, Mou J, Erkan U, Toktas A, Li Q, Wang C and Cao Y 2025 *IEEE Internet Things J.* **12** 18115
[77] Zhang W, Wang S, Han W, Yu H and Zhu Z 2020 *Entropy* **22** 73
[78] Patro K A K and Acharya B 2020 *First International Conference On Power, Control And Computing Technologies (ICPC2T)* pp. 411–416
[79] Panwar A, Biban G, Chugh R, Tassaddiq A and Alharbi R 2024 *Heliyon* **10** e31618
[80] Fang P, Liu H, Wu C and Liu M 2022 *Multimed. Tools Appl.* **81** 21811