

ARTICLE

A Dual-Neuron Memristor Hopfield Neural Network with Controllable Multiple Equilibrium Points: Dynamical Analysis, FPGA Implementation, and Image Encryption Application

Yanyu Zhu¹, Jie Jin^{2,*}, Lv Zhao^{2,3} and Fei Yu⁴

¹School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China

²School of Information Engineering, Changsha Medical University, Changsha, China

³School of Electrical and Information Engineering, Hunan Institute of Technology, Hengyang, China

⁴School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China

*Corresponding Author: Jie Jin. Email: jj67123@hnust.edu.cn

Received: 12 May 2026; Accepted: 27 July 2026; Published: 15 September 2026

ABSTRACT: To address the issues of multi-neuron architectures, high parameter redundancy, and complex hardware implementation in existing memristive Hopfield neural networks (MHNN) for image encryption, a simple structure dual-neuron memristive Hopfield neural network (DNMHNN) modulated by multifrequency square waves is proposed in this study. The proposed DNMHNN model consists of only two neurons and one memristor, and by introducing dual-frequency square-wave external excitation into the memristor, the dynamical behavior of the DNMHNN model can be flexibly regulated. The simulation results verify that the proposed DNMHNN model can generate stable chaotic behavior over a wide parameter range. Furthermore, the DNMHNN is also implemented on a Zynq-7000 FPGA platform, and the experimental results are highly consistent with the numerical simulations, demonstrating the feasibility and stability of the DNMHNN model in practical hardware applications. On this basis, a DNMHNN-based image encryption algorithm with pixel permutation, XOR substitution, and bidirectional ciphertext-feedback diffusion is designed using chaotic sequences generated by the DNMHNN model. Security analysis indicates that the DNMHNN-based image encryption algorithm exhibits strong performance in histogram distribution, correlation reduction, information entropy, NPCR, UACI, key sensitivity, and resistance to noise, occlusion, tampering, and differential attacks, which is further verified by quantitative robustness metrics including MSE, PSNR, SSIM, and recovered-image correlation. The results demonstrate that the proposed dual-neuron memristive DNMHNN maintains a simple structure while preserving rich chaotic dynamical properties. Therefore, it provides an effective chaotic key source for lightweight image encryption and edge secure computing.

KEYWORDS: Memristor; Hopfield neural network (HNN); chaotic dynamics; FPGA implementation; image encryption

1 Introduction

Artificial neural networks have demonstrated strong capabilities in learning, association, and parallel information processing, motivating the development of various neural-network architectures and their dynamical models [1]. Among the myriad of neural network architectures, the Hopfield neural network (HNN) stands as a quintessential exemplar bridging “neural computation—nonlinear dynamics—engineering applications”, owing to its energy function framework and associative memory capabilities [2,3]. Concurrently, memristors, nonlinear devices characterized by “state-dependent conductance/resistance” and

“historical memory” properties, have furnished a fundamental physical basis for fabricating neuromorphic circuits exhibiting synaptic plasticity and complex dynamics since the theoretical conception of memristive systems by Chua [4,5]. The subsequent realization of nanoscale memristive devices further validated the engineering feasibility of the memristive effect, thereby catalyzing extensive research into memristors for in-memory computing and neuromorphic hardware applications [6–8].

In recent years, the integration of memristors and other mem-elements, such as memdiodes and discrete memristive elements, into the synaptic coupling or self-feedback pathways of HNNs has demonstrably enhanced complex behaviors such as multistability, hidden attractors, multi-scroll/multi-wing structures, and initial condition modulation [9–11]. This has positioned memristive HNNs (MHNN) as significant dynamical sources capable of generating high-complexity chaotic sequences, including memdiode-driven HNNs for medical-volume encryption and discrete MHNNs for FPGA-oriented image security [12–15]. Notably, phenomena like “initial-offset/spatial-offset coexisting attractors” have introduced novel mechanisms for constructing numerous distinguishable attractor families within a single parameter set, thereby providing theoretical underpinnings for key space expansion, sequence diversity enhancement, and controllable complexity design [16–18]. Concurrently, recent studies have demonstrated the potential of MHNNs in nonlinear dynamics and information-security applications [19,20].

From an engineering implementation perspective, the successful high-speed, parallel, and reproducible experimental validation of chaotic/hyperchaotic neural networks on programmable platforms like FPGAs would significantly enhance their deployability in edge devices and embedded security computing [21–24]. On the application side, the inherent initial value sensitivity, ergodicity, and pseudo-randomness of chaotic systems and neural network output sequences make them widely adopted for permutation and diffusion key stream generation in image encryption. Numerous studies have demonstrated encryption schemes for color and medical images, including medical-image transmission, wireless-sensor-network image protection, and lightweight communication-oriented image encryption, utilizing Hopfield networks, memristive chaotic systems, and their variants [25–28]. The practicality of these approaches has been validated through metrics such as information entropy, correlation, NPCR/UACI, and robustness against noise and occlusion [29–33]. Furthermore, research has begun to directly address the feasibility of two-neuron HNNs in dynamics and image encryption, suggesting that a valuable trade-off space may exist between low-dimensional structures and the realization of complex dynamics vs. simplified hardware costs [34,35]. Comprehensive reviews on memristive chaotic image encryption have also highlighted that, under the guarantee of security, reducing system dimensionality, minimizing parameter count, and improving implementability are key directions for future deployable solutions [36].

Despite these efforts and recent progress in memdiode-driven CT encryption and discrete MHNN-based industrial image encryption, a significant structural gap persists in the current research landscape. A substantial body of cryptographic investigation into MHNNs has predominantly focused on architectures with three or more neurons. This focus aims to harness the “inherent complexity” arising from higher dimensionality, yet it concurrently introduces challenges such as parameter redundancy, increased implementation complexity, and augmented hardware resource overhead [36]. In contrast, more compact, two-neuron MHNNs, if capable of exhibiting stable and reproducible chaotic dynamics under external stimuli or parameter modulation (e.g., multi-frequency switching signals), and subsequently undergoing FPGA-level validation for high-security image encryption applications, could establish a more practically engineered balance among “dynamical complexity, key controllability, and hardware implementability”. This would, in turn, offer a more streamlined and deployable chaotic key generation and encryption framework for lightweight information security and edge intelligence applications [36]. Specifically, recent mem-element HNN encryption schemes have achieved strong security performance, but many of them rely on

three-neuron/four-dimensional structures, fractional-order models, DNA/S-box operations, or application-specific medical-volume encryption. Therefore, the motivation of this study is to investigate whether a dual-neuron, MHNN can simultaneously provide controllable multiple equilibrium points, reproducible chaotic dynamics, FPGA implementability, and competitive image-encryption performance.

On the basis of the preceding analysis, this work establishes a three-dimensional DNMHNN with a controllable number and distribution of equilibrium points. Specifically, the proposed model possesses multiple unstable equilibrium points, and its chaotic behaviors are systematically examined using several standard dynamical analysis methods. The resulting dynamics are further reproduced through FPGA implementation, with the experimental results showing good agreement with the numerical simulations. Moreover, an image encryption algorithm is designed based on the proposed DNMHNN, with its security and feasibility validated through comparative analyses involving key sensitivity, histogram characteristics, correlation coefficients, information entropy, and noise attacks.

The remainder of this paper is organized as follows. [Section 2](#) develops the DNMHNN model composed of one memristor and two neurons. [Section 3](#) presents a detailed investigation of the equilibrium points of the DNMHNN, analyzes and simulates its dynamic behaviors under different conditions. [Section 4](#) further confirms the chaotic dynamics of the DNMHNN through FPGA-based implementation. [Section 5](#) designs and validates a new chaotic image encryption scheme derived from the presented DNMHNN. [Section 6](#) concludes the paper.

2 The Proposed DNMHNN Model

This section develops a voltage-controlled memristor model as a non-smooth system driven by multi-frequency signals, analyzes its memory effects and $V - I$ hysteresis, and then integrates it into a DNMHNN to enhance adaptive synaptic interactions and dynamic pattern processing.

2.1 The Designed Voltage-Controlled Memristor

This subsection investigates a class of non-smooth memristive systems modulated by multi-frequency switching signals, whose dynamical behaviors are described by first-order nonlinear differential equations. The system, composed of external AC voltage excitation, internal state variables, and state-dependent nonlinear admittance, exhibits significant memory effects and current-voltage hysteresis characteristics. The relationship between voltage, output current, and internal state variables can be clearly observed through its mathematical expression, laying the foundation for further experimental design and theoretical analysis. Specifically, the mathematical expression for the memristor model is presented below.

$$\begin{cases} v(t) = A \sin(2\pi Ft) \\ \dot{\varphi}(t) = g(t) v(t) - d h_{M,N}(\varphi) \\ i(t) = [a + b h_{M,N}^2(\varphi)] v(t) \end{cases} \quad (1)$$

where the parameter v represents the input voltage, A and F denote the voltage amplitude and frequency, respectively. The system's output current satisfies a state-dependent Ohmic relationship with the input voltage. The parameter i represents the output current, while parameters a and b are memristor parameters, whose values are constants. The evolution of the internal state variable $\varphi(t)$ is given by the following first-order differential equation: where $d > 0$ is the feedback intensity coefficient, and $h_{M,N}(\varphi)$ is a piecewise-defined nonlinear function describing multi-threshold saturation and non-smooth feedback characteristics. Here, M and N are integer-valued branch-control parameters. Their admissible domain is defined as $(M, N) \in \mathcal{D} = \{M \geq 0, N < 0\} \cup \{M < 0, N \geq 0\}$. The complete definition of $h_{M,N}(\varphi)$ is given as

$$h_{M,N}(\varphi) = \begin{cases} \varphi - \operatorname{sgn}(\varphi), & N < 0, M = 0, \\ \varphi - \operatorname{sgn}(\varphi) - \sum_{j=1}^M [\operatorname{sgn}(\varphi + 2j) + \operatorname{sgn}(\varphi - 2j)], & N < 0, M \geq 1, \\ \varphi, & M < 0, N = 0, \\ \varphi - \sum_{i=1}^N [\operatorname{sgn}(\varphi + 2i - 1) + \operatorname{sgn}(\varphi - 2i + 1)], & M < 0, N \geq 1. \end{cases} \quad (2)$$

For the branch $N < 0, M \geq 0$, the switching thresholds are located at $\varphi = 0, \pm 2, \pm 4, \dots, \pm 2M$. For the branch $M < 0, N \geq 1$, the switching thresholds are located at $\varphi = \pm 1, \pm 3, \dots, \pm(2N - 1)$. Thus, M and N do not represent two independent summation limits in the same branch. Instead, their signs determine the threshold-distribution branch, while the non-negative parameter in that branch determines the number of switching thresholds. Physically, these thresholds correspond to multi-level conductance switching states of the memristor. Increasing the number of thresholds increases the number of piecewise regions in the internal-state space, thereby forming a multi-level piecewise structure within the state space and enabling richer equilibrium distributions and more complex nonlinear dynamics. This structure can effectively emulate the behavior of nonlinear devices possessing multi-saturation or multi-stable characteristics. To delineate the modulation effect of external control signals on system parameters, a time-varying modulation function is incorporated, constructed by superimposing two square waves of different frequencies.

$$g(t) = a_1 \operatorname{sgn}(\sin(\omega_1 t)) + a_2 \operatorname{sgn}(\sin(\omega_2 t)) \quad (3)$$

Here, a_1 and a_2 represent the modulation weights, and ω_1 and ω_2 denote the modulation angular frequencies, with $\operatorname{sgn}(\cdot)$ signifying the sign function. This modulation function renders the system a type of non-autonomous system characterized by non-smooth time-varying parameters. Consequently, the system's equivalent admittance is not only dependent on the instantaneous input voltage, but also intrinsically linked to internal state variables, thereby exhibiting typical memristive system characteristics.

By combining the above equations, the designed model can be identified as a type of non-smooth memristive system. As its internal state continuously changes in response to external excitation, the system displays characteristic hysteresis loops in the $V - I$ plane. These loops are obtained by imposing a periodic voltage on the memristor and simultaneously recording the corresponding output current i . In this work, the memristor parameters are chosen as $a = 0.85$, $b = 1.5$, and $d = 1.2$. While keeping the remaining parameters in Eq. (1) unchanged, an input voltage with amplitude $A = 1$ is applied, and the corresponding frequency-dependent $V - I$ trajectories are examined at $f = 2, 4, 8$ Hz. The differential equations are solved numerically by the ODE45 method, and the results are presented in Fig. 1a. The area enclosed by the pinched hysteresis loop gradually decreases as the frequency rises, which suggests that the memristor dynamics are closely affected by the frequency of the applied signal. Fig. 1b presents the amplitude-dependent $V - I$ curves obtained when the input frequency is fixed at $f = 5$ Hz and the voltage amplitudes are set to $A = 0.99, 0.8, 0.5$. It can be observed that, with increasing A , the “8”-shaped pinched hysteresis loop becomes increasingly evident, indicating a strong dependence of the memristor response on the signal amplitude. Moreover, both the morphology and enclosed area of the hysteresis curve vary markedly with changes in excitation frequency and amplitude: at low frequency or high amplitude, the system exhibits a significant memory effect, whereas under high-frequency excitation, the hysteresis loop progressively shrinks and tends toward nearly linear resistive behavior.

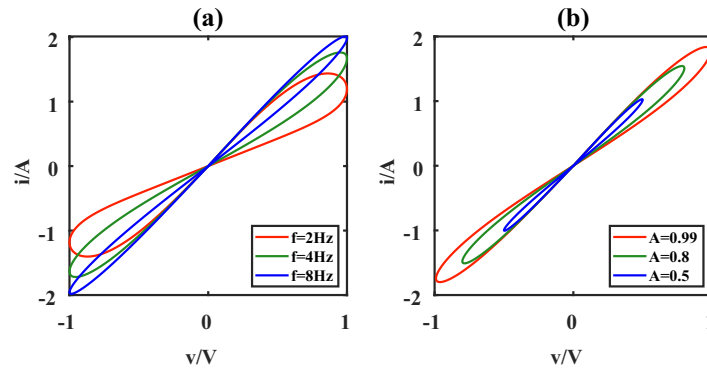


Figure 1: Memristor clamping hysteresis loops. (a) The trajectory of the pinched hysteresis loop with a constant input voltage amplitude of $A = 1$ at different frequencies f . (b) The trajectory of the pinched hysteresis loop with different input voltage amplitude values A at a fixed frequency $f = 5$ Hz.

Moreover, the $V - I$ characteristic of the memristor demonstrates evident nonlinear behavior, which implies that its resistance is indirectly regulated by the applied voltage. The emergence of the pinched hysteresis loop further reveals the memristor's memory capability, allowing it to preserve its resistance state even when the voltage becomes zero. These properties indicate that the model shows substantial adaptability and representativeness in characterizing complex nonlinear memristive devices and their parameter-adjustment behaviors, thereby emphasizing the potential importance of memristors in memory storage and neural network applications.

2.2 The Constructed DNMHNN Model

HNNs are a foundational type of neural network widely studied in network research for their unique ability to perform pattern recognition and support associative memory. Each HNN consists of interconnected neurons that collectively form a mathematical model capable of storing and recalling patterns efficiently. This baseline model demonstrates the efficacy of HNNs in various applications, from cognitive computing to optimization problems, making them a critical component in understanding and advancing neural network architectures. Generally, the mathematical formulation of a baseline HNN model is expressed as follows:

$$C_i \dot{x}_i = -\frac{x_i}{R_i} + \sum_{j=1}^n w_{ij} \tanh(x_j) + I_i, \quad (i, j \in N^*) \quad (4)$$

where x_i represents the membrane potential of neuron i , C_i its membrane capacitance, R_i its membrane resistance, w_{ij} the synaptic weight from neuron j to neuron i , $\tanh(x_j)$ the nonlinear activation of neuron j , I_i an external input current applied to neuron i , and N^* the set of all neurons in the network; this equation captures the temporal evolution of each neuron's membrane potential as a result of passive decay, interactions with other neurons via synaptic connections, and external stimuli, thereby defining the core dynamics that enable the network to perform associative memory and pattern recognition tasks.

A MHNN can be developed from the classical HNN to better describe complex neural interactions and nonlinear memory effects. In this dual-neuron model, the memristor is introduced as a key element for synaptic coupling, enabling bidirectional interaction between neurons and improving the network's ability to represent adaptive connections. Owing to the state-dependent properties of the memristor, the proposed structure exhibits rich dynamic behavior under different operating conditions. Meanwhile, the

activation function plays an essential role in determining the nonlinear response of each neuron, while the external stimulation current further affects the internal evolution of the system. Through this mechanism, the network enhances information processing capability and provides a useful framework for studying neural dynamics and memristive computation. Based on the aforementioned baseline HNN model and the described memristors, the proposed DNMHNN model is illustrated in Fig. 2.

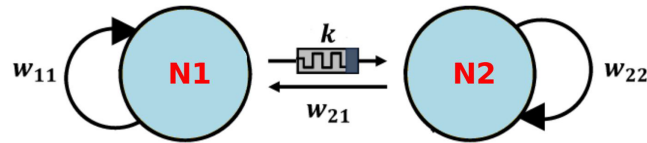


Figure 2: Topological structure of the DNMHNN model.

The proposed DNMHNN model has two neurons labeled as N1 and N2, each represented by a circular node. The self-connections w_{11} and w_{22} indicate the recurrent synaptic weights for N1 and N2, respectively, representing the influence of each neuron on itself. The inter-neuron connections are asymmetric: the $N_1 \rightarrow N_2$ pathway is governed by the fixed synaptic weight w_{21} , whereas the $N_2 \rightarrow N_1$ pathway is mediated by the memristive coupling term $kW(\varphi)$, enabling adaptive modulation only in the $N_2 \rightarrow N_1$ direction. This diagram illustrates the core architecture of the DNMHNN, highlighting both recurrent and memristor-based inter-neuron interactions that enable associative memory and dynamic pattern processing. Furthermore, to investigate the network's dynamic characteristics, an external stimulus current I is applied to memristor k . This external current I can perturb the memristor k , thereby influencing the overall dynamic behavior of the network. Consequently, the mathematical formulation of the DNMHNN model, as depicted in Fig. 2, is presented below.

$$\begin{cases} \dot{x}_1 = -x_1 + w_{11} \tanh(x_1) + kW(\varphi) \tanh(x_2), \\ \dot{x}_2 = -x_2 + w_{21} \tanh(x_1) + w_{22} \tanh(x_2), \\ \dot{\varphi} = I(t) \tanh(x_2) - d h_{MN}(\varphi). \end{cases} \quad (5)$$

To facilitate both analytical and numerical computations, the membrane capacitances of the neurons are standardized as $C_1 = C_2 = 1$, and the membrane resistances are similarly set as $R_1 = R_2 = 1$. The memristor parameters are specified as $a = 0.55$ and $b = 1.3$, accompanied by a memristive damping coefficient $d = 5.7$, which are essential for accurately capturing the nonlinear dynamic behavior inherent to the memristor. The inter-neuron coupling strength is assigned a value of $k = -1.63$, which determines the degree of interaction between the two neurons. The external stimulus current $I(t)$ is implemented using the same dual-frequency square-wave form as the modulation function $g(t)$ defined in Eq. (3), with driving amplitudes $a_1 = 3.5$ and $a_2 = 2.8$, and corresponding angular frequencies $\omega_1 = 2.1$ and $\omega_2 = 3.3$, enabling the network to respond to multiple temporal scales. Furthermore, to maintain the overall stability of the system and ensure proper functionality of the network dynamics, the synaptic weight coefficients are chosen as $w_{11} = 2.2$, $w_{21} = 3.5$, and $w_{22} = 1.5$, which regulate the recurrent and inter-neuron interactions within the DNMHNN architecture.

3 Dynamical Analysis of the DNMHNN Model

This section systematically investigates the DNMHNN model by identifying equilibrium points and analyzing their stability, generating phase-space diagrams to visualize dynamic behavior, and employing bifurcation diagrams along with Lyapunov exponents to examine chaotic and periodic transitions under

varying parameters, thereby providing a comprehensive understanding of the system's complex dynamics and parameter-dependent behavior.

3.1 Equilibrium Points and Stability Analysis of the DNMHNN Model

Equilibrium points and their stability play a pivotal role in understanding the chaotic states of dynamic systems, and stability issues have also been recently extended to generalized neural-network models such as Clifford-valued neural networks with time-varying delays [37]. The procedure for determining the equilibrium points of the DNMHNN model is outlined as follows:

Step 1: The equilibrium equations are formulated for the time-independent equilibrium points of the DNMHNN model. It should be clarified that the proposed DNMHNN model is non-autonomous due to the square-wave excitation $I(t)$. Therefore, the equilibrium points discussed in this section are understood as time-independent constant solutions of the original non-autonomous system, rather than solutions obtained by arbitrarily freezing the excitation at a specific time instant. In the proposed model, the external excitation enters the state equations only through the product $I(t) \tanh(x_2)$ in the third equation. For the equilibrium points obtained in this work, one has $x_2^* = 0$, which gives

$$I(t) \tanh(x_2^*) = 0, \quad \forall t. \quad (6)$$

Thus, the time-varying excitation term vanishes identically at these points, regardless of the instantaneous value of $I(t)$. Furthermore, substituting $x_2^* = 0$ into the second equilibrium equation yields $x_1^* = 0$, and the remaining equilibrium condition reduces to $h(\varphi^*) = 0$. Hence, the obtained points $(0, 0, \varphi^*)$ are valid time-independent equilibrium points of the original non-autonomous DNMHNN model, rather than artifacts caused by setting $I(t) = I(0) = 0$. Based on the above analysis, the equilibrium point search can be reduced to solving the following nonlinear algebraic equations:

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} -x_1 + w_{11} \tanh(x_1) + k_1 W(x_3) \tanh(x_2) \\ -x_2 + w_{21} \tanh(x_1) + w_{22} \tanh(x_2) \\ -d h(\varphi) \end{bmatrix} = 0 \quad (7)$$

Here,

$$W(\varphi) = a + b [h(\varphi)]^2 \quad (8)$$

where $h(\varphi)$ represents a piecewise non-smooth nonlinearity involving the superposition of multi-threshold sign functions. Due to the strong nonlinearity and multi-segment structure of F , equilibrium points often exhibit coexisting multiple solutions. Consequently, a local iterative method with a single initial value cannot guarantee the discovery of all equilibrium points, necessitating the adoption of a comprehensive initial value strategy.

Step 2: To enhance the coverage of multiple solutions, a set of initial points is constructed based on the principle of “perturbation in the neighborhood of critical points”. This is achieved by pre-selecting a group of critical center values along the φ direction (covering multiple threshold segments and their neighborhoods). Small random perturbations are then applied around each critical value to form a dense sampling. Simultaneously, random initial values are assigned within finite intervals in the (x_1, x_2) plane to achieve broad coverage of the state space. The resulting initial point matrix $\{x_0^{(m)}\}_{m=1}^N$ serves as an entry point for subsequent multiple local solutions, ensuring a high probability of “hitting” equilibrium points within different basins of attraction.

Step 3: An improved Newton-Raphson iteration is independently performed for each initial point $x_0^{(m)}$ to solve $F(x) = 0$. In the k -th iteration, the residual $F(x_k)$ and Jacobian matrix are first computed:

$$J(\mathbf{x}) = \begin{bmatrix} -1 + w_{11}T_1 & k_1 W(\varphi)T_2 & k_1 \tanh(x_2) W'(\varphi) \\ w_{21}T_1 & -1 + w_{22}T_2 & 0 \\ 0 & 0 & -d h'(\varphi) \end{bmatrix} \quad (9)$$

Here,

$$\begin{aligned} T_1 &= 1 - \tanh^2(x_1), & T_2 &= 1 - \tanh^2(x_2) \\ W(\varphi) &= a + b h(\varphi)^2, & W'(\varphi) &= 2b h(\varphi) h'(\varphi) \end{aligned} \quad (10)$$

If $\|F(x_k)\| < \varepsilon$ (where $\varepsilon = 10^{-12}$), the solution is convergent, and x_k is output as a candidate equilibrium point. Otherwise, the update direction is adaptively selected based on the invertibility of the Jacobian. When $J(\mathbf{x}_k)$ is nearly singular, the algorithm switches from the Newton step to a conservative residual-based update along $-F(\mathbf{x}_k)$ with step-size reduction. This fallback direction differs from the steepest-descent direction $J(\mathbf{x}_k)^T F(\mathbf{x}_k)$ of the merit function $\phi(\mathbf{x}) = \frac{1}{2} \|F(\mathbf{x})\|_2^2$. When $J(x_k)$ is well-conditioned, the standard Newton step is employed:

$$\Delta \mathbf{x}_k = -J(\mathbf{x}_k)^{-1} F(\mathbf{x}_k), \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \Delta \mathbf{x}_k. \quad (11)$$

To mitigate overshoot and divergence under strong nonlinearity, the algorithm incorporates a simplified line search (damping strategy). Initially, it attempts a full step update with $\alpha_k = 1$. If the norm of the residual at the new point increases ($\|F(x_{k+1})\| > \|F(x_k)\|$), the step size is halved proportionally until the residual improves or the step size reaches a predefined lower bound. This approach enhances global robustness while preserving the local quadratic convergence potential of Newton's method. It is important to note that, owing to the non-smoothness introduced by the sign functions, the smooth approximation $\text{sgn}(z) \approx \tanh(\alpha z)$ is employed only in the Newton iteration to improve numerical convergence. This approximation is not used in the final local stability analysis. After the candidate stationary states are obtained, they are substituted back into the original non-smooth system for verification. Since the considered stationary states are not located on the switching surfaces of the sign functions, the original system is locally smooth around them, and the derivatives of the sign terms are zero. Therefore, the local Jacobian is calculated from the original piecewise-smooth system rather than from the smoothed approximation.

Step 4: Validation and deduplication of results from all initial values. The program not only requires the iterative process to meet convergence criteria but also imposes a strict residual test (e.g., $\|F(x_*)\| < 10^{-10}$) to discard results with spurious convergence or numerical drift. Subsequently, a Euclidean distance threshold method is employed for clustering the candidate solution set. If a newly obtained x_* is within a given tolerance (e.g., 10^{-3}) of any point in the existing solution set, it is considered the same equilibrium point and discarded; otherwise, it is added to the set as a new, unique equilibrium point. This process ensures that the final output set of equilibrium points satisfies the equation's accuracy requirements while avoiding redundant entries due to multiple initial values converging to the same point. Additionally, the six resulting equilibrium points are presented in [Table 1](#).

Table 1: Equilibrium points and stability of the DNMHNN model for the representative case $M = 2, N = -3$.

Number	Equilibrium Point	Eigenvalue	Stability
Equilibrium point 1	[0.0000, 0.0000, 5.0000]	0.8500 + 1.7364i, 0.8500 - 1.7364i, -5.7000 + 0.0000i	Unstable index-2 saddle point
Equilibrium point 2	[0.0000, 0.0000, 3.0000]	0.8500 + 1.7364i, 0.8500 - 1.7364i, -5.7000 + 0.0000i	Unstable index-2 saddle point
Equilibrium point 3	[0.0000, 0.0000, 1.0000]	0.8500 + 1.7364i, 0.8500 - 1.7364i, -5.7000 + 0.0000i	Unstable index-2 saddle point
Equilibrium point 4	[0.0000, 0.0000, -1.0000]	0.8500 + 1.7364i, 0.8500 - 1.7364i, -5.7000 + 0.0000i	Unstable index-2 saddle point
Equilibrium point 5	[0.0000, 0.0000, -3.0000]	0.8500 + 1.7364i, 0.8500 - 1.7364i, -5.7000 + 0.0000i	Unstable index-2 saddle point
Equilibrium point 6	[0.0000, 0.0000, -5.0000]	0.8500 + 1.7364i, 0.8500 - 1.7364i, -5.7000 + 0.0000i	Unstable index-2 saddle point

For the time-independent equilibrium points obtained in this work, one has $x_1^* = 0$, $x_2^* = 0$, and $h(\varphi^*) = 0$. Therefore, $T_1 = T_2 = 1$, $W(\varphi^*) = a$, and $W'(\varphi^*) = 0$. Since the equilibrium values of φ^* are located away from the switching thresholds of the sign functions, $h'(\varphi^*) = 1$ is used in the local linearization. Hence, the Jacobian matrix at each equilibrium point becomes

$$J(\mathbf{x}^*) = \begin{bmatrix} -1 + w_{11} & k_1 a & 0 \\ w_{21} & -1 + w_{22} & 0 \\ 0 & 0 & -d \end{bmatrix}. \quad (12)$$

Using the parameters of the proposed model, the eigenvalues are recalculated as $\lambda_{1,2} = 0.8500 \pm 1.7364i$, $\lambda_3 = -5.7000$. Since two eigenvalues have positive real parts and one eigenvalue has a negative real part, each equilibrium point has a two-dimensional unstable manifold and is therefore classified as an unstable index-2 saddle point.

To further clarify the controllability of the multiple equilibrium points, the relationship between the branch-control parameters M, N and the equilibrium-point distribution is analyzed explicitly. According to Eq. (2), M and N determine different threshold-distribution branches of the nonlinear function $h_{M,N}(\varphi)$. Since all the time-independent equilibrium points obtained in this work satisfy $x_1^* = 0$, $x_2^* = 0$, and $h_{M,N}(\varphi^*) = 0$, the equilibrium set of the DNMHNN can be expressed as $E_{M,N} = \{(0, 0, \varphi^*) \mid h_{M,N}(\varphi^*) = 0\}$. Therefore, the number and locations of equilibrium points are determined by the zeros of $h_{M,N}(\varphi)$.

For the branch $N < 0, M \geq 0$, the zeros of $h_{M,N}(\varphi)$ are located at $\varphi^* = \pm 1, \pm 3, \dots, \pm(2M + 1)$, and the number of equilibrium points is $N_E = 2M + 2$. This means that increasing M adds one pair of equilibrium points symmetrically along the φ -axis.

For the branch $M < 0, N = 0$, the nonlinear function reduces to $h_{M,N}(\varphi) = \varphi$, and thus the only equilibrium point is $E_{M,0} = (0, 0, 0)$.

For the branch $M < 0, N \geq 1$, the zeros of $h_{M,N}(\varphi)$ are located at $\varphi^* = 0, \pm 2, \pm 4, \dots, \pm 2N$, and the number of equilibrium points is $N_E = 2N + 1$. This indicates that increasing N adds one pair of equilibrium points at even locations while preserving the central equilibrium point at the origin.

The above analytical results show that the number and spatial locations of equilibrium points can be explicitly controlled by M and N . To verify this conclusion, the improved Newton-Raphson method described above was used to calculate the equilibrium points under different parameter settings. The results are summarized in Table 2.

Table 2: Variation of equilibrium points with M and N .

Branch	Parameter Setting	Number of Equilibrium Points	Equilibrium Locations
$M < 0, N = 0$	$(-2, 0)$	1	$(0, 0, 0)$
$M < 0, N = 1$	$(-2, 1)$	3	$(0, 0, 0), (0, 0, \pm 2)$
$M < 0, N = 2$	$(-2, 2)$	5	$(0, 0, 0), (0, 0, \pm 2), (0, 0, \pm 4)$
$M < 0, N = 3$	$(-2, 3)$	7	$(0, 0, 0), (0, 0, \pm 2), (0, 0, \pm 4), (0, 0, \pm 6)$
$M < 0, N = 6$	$(-2, 6)$	13	$(0, 0, 0), (0, 0, \pm 2), \dots, (0, 0, \pm 12)$
$M = 0, N < 0$	$(0, -3)$	2	$(0, 0, \pm 1)$
$M = 1, N < 0$	$(1, -3)$	4	$(0, 0, \pm 1), (0, 0, \pm 3)$
$M = 2, N < 0$	$(2, -3)$	6	$(0, 0, \pm 1), (0, 0, \pm 3), (0, 0, \pm 5)$
$M = 3, N < 0$	$(3, -3)$	8	$(0, 0, \pm 1), (0, 0, \pm 3), (0, 0, \pm 5), (0, 0, \pm 7)$
$M = 6, N < 0$	$(6, -3)$	14	$(0, 0, \pm 1), (0, 0, \pm 3), \dots, (0, 0, \pm 13)$

It can be seen from Table 2 that, in the branch $M < 0, N \geq 0$, the number of equilibrium points depends on N , while changing the negative value of M does not affect the equilibrium-point number. In contrast, in the branch $M \geq 0, N < 0$, the number of equilibrium points depends on M , while the negative value of N only identifies the corresponding branch. Therefore, M and N provide an explicit and controllable mechanism for adjusting both the number and locations of the equilibrium points. The three cases shown in Fig. 3, namely $(M, N) = (-2, 1), (2, -3)$, and $(3, -6)$, correspond to 3, 6, and 8 equilibrium points, respectively, which further confirms the controllable equilibrium-point distribution of the proposed DNMHNN.

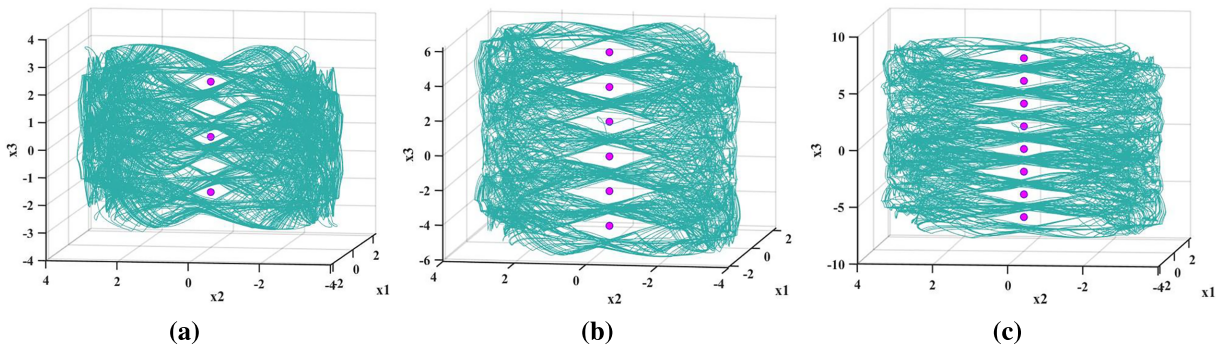


Figure 3: The spatial distribution of multiple equilibrium points of the system solved by the Newton-Raphson method under different conditions of M and N , specifically for (a) $M = -2, N = 1$; (b) $M = 2, N = -3$; (c) $M = 3, N = -6$.

Finally, after obtaining the set of equilibrium points in Table 1, the program performs a linearization stability analysis for each equilibrium point. Specifically, the Jacobian matrix $J(x_i)$ is computed at each equilibrium point x^i , and its eigenvalue spectrum $\{\lambda_j^i\}$ is determined. Local stability is assessed based on the sign of the real parts of the eigenvalues: if all $R\{\lambda_j^i\} < 0$, the equilibrium point is locally asymptotically stable; if there are p eigenvalues with positive real parts, the equilibrium point possesses a p -dimensional unstable manifold and can be further classified as a saddle point with an instability index of p . The spatial distribution of multiple equilibrium points for three distinct scenarios where M and N are varied under different conditions is depicted in Fig. 3.

3.2 Phase Diagram of the DNMHNN Mode

In this subsection, we conduct a dynamical analysis of the DNMHNN model using phase-space analysis. A MHNN can be developed from the classical HNN to better describe complex neural interactions and nonlinear memory effects. In this dual-neuron model, the memristor is introduced as a key element for synaptic coupling, enabling bidirectional interaction between neurons and improving the network's ability to represent adaptive connections. Owing to the state-dependent properties of the memristor, the proposed structure exhibits rich dynamic behavior under different operating conditions. Meanwhile, the activation function plays an essential role in determining the nonlinear response of each neuron, while the external stimulation current further affects the internal evolution of the system. Through this mechanism, the network enhances information processing capability and provides a useful framework for studying neural dynamics and memristive computation.

During the simulation, the memristor parameters were assigned as $a = 0.55$, $b = 1.3$, and $d = 5.7$, while the coupling strength between the memristors and neurons was set to $k = -1.63$, allowing for a detailed investigation of the DNMHNN model's dynamical phenomena. Additional system parameters were defined as $M = -2$, $N = 1$, $a_1 = 3.5$, $a_2 = 2.8$, $\omega_1 = 2.1$, $\omega_2 = 3.3$, $w_{11} = 2.2$, $w_{21} = 3.5$, and $w_{22} = 1.5$, ensuring a comprehensive representation of the network interactions and modulation effects. The simulation was conducted over a total duration of 1200 time units with a time step of 0.01, starting from the initial state $(x, y, z) = (0.1, 0.1, 0.1)$, which provides a consistent initial condition for observing system evolution. Based on these settings, the three-dimensional phase-space trajectory of (x, y, z) was generated, along with the phase diagram projected onto the (y, z) plane, and the time-domain evolution of the system over the interval (0 s, 1200 s), all of which are illustrated in Fig. 4 to visualize the DNMHNN's dynamic behavior.

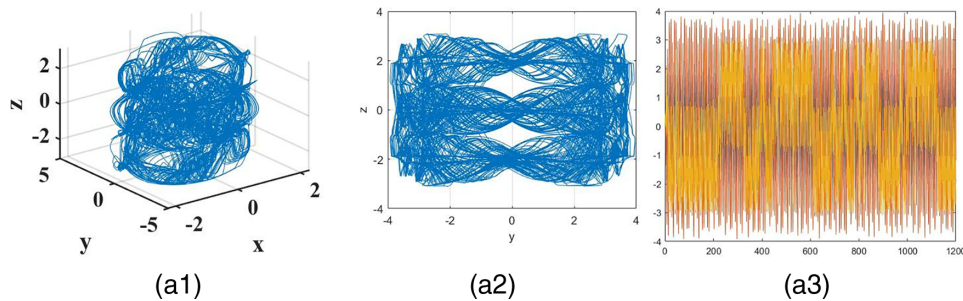


Figure 4: (Continued)

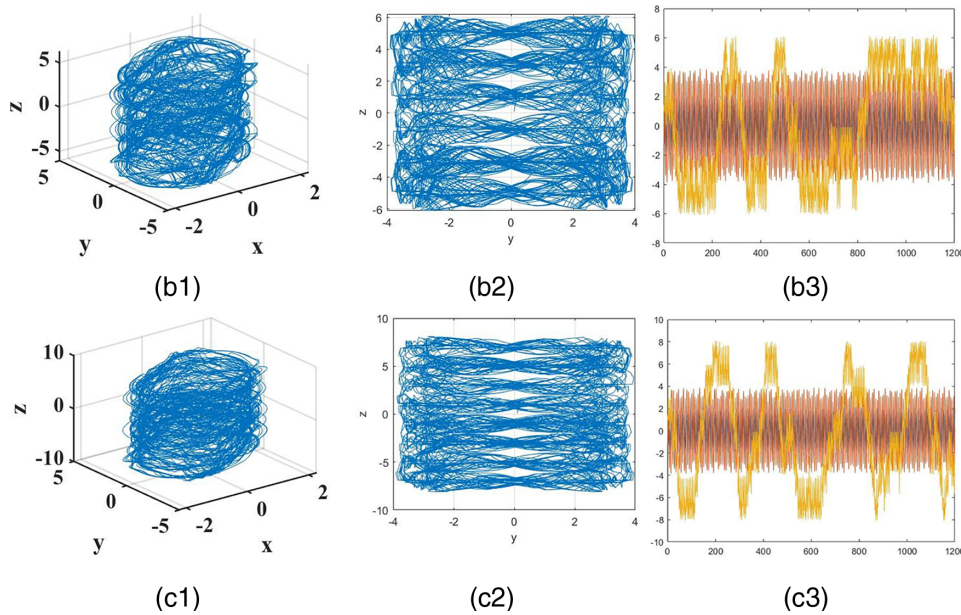


Figure 4: The 3D phase diagrams, planar phase diagrams, and time-domain plots for the DNMHNN model (5) under varying M and N conditions: (a1–a3) $M = -2$, $N = 1$; (b1–b3) $M = 2$, $N = -3$; (c1–c3) $M = 3$, $N = -6$.

3.3 Dynamics Analysis of Bifurcation Diagrams and Lyapunov Exponents of the DNMHNN Model

Bifurcation diagrams and Lyapunov exponents serve as essential tools in the analysis of dynamical systems, providing insight into the system's dynamic characteristics and revealing its evolution patterns over time. By applying these methods, researchers can investigate how variations in parameters influence the system state, and how such changes affect overall system behavior. This approach allows for a detailed examination of complex dynamics, making bifurcation diagrams and Lyapunov exponents invaluable in understanding and analyzing dynamical systems.

Lyapunov exponents provide a quantitative measure of how trajectories within a system diverge or converge over time. Specifically, the largest Lyapunov exponent serves as a key indicator: if it is positive, the system is considered to be in a chaotic state, demonstrating sensitivity to initial conditions and significant trajectory divergence. Conversely, if the largest exponent is negative or zero, the system may exhibit a periodic state or remain in a non-chaotic condition, with trajectories converging or maintaining overall system stability. These exponents thus offer a numerical foundation for distinguishing between chaotic and stable behaviors within a system.

Bifurcation diagram analysis provides a visual method for observing system transitions, illustrating how neuronal state values change as parameters vary. Within certain parameter ranges, if the state values form discrete points arranged in a closed loop, the system is identified as being in a periodic state. In contrast, when the neuronal state values appear as dense, continuous points filling an interval, the system exhibits chaotic behavior. This approach effectively captures the evolution from simple to complex dynamics and clearly distinguishes between periodic and chaotic states.

Fig. 5 presents the Lyapunov exponents of the DNMHNN model (5) over time t under the specific parameter configuration $a = 0.55$, $b = 1.3$, $d = 5.7$, $w_{11} = 2.2$, $w_{21} = 3.5$, $w_{22} = 2.5$, $a_1 = 3.5$, $a_2 = 2.8$, $\omega_1 = 2.1$, $\omega_2 = 3.3$, $M = -2$, and $N = 1$. The computed Lyapunov exponents are $LE_1 = 1.73547$, $LE_2 = -0.00147544$, and $LE_3 = -0.482913$. The positive value of LE_1 indicates exponential divergence of nearby trajectories along

one direction and confirms the presence of chaotic dynamics. Although LE_2 is numerically negative, its magnitude is extremely small, on the order of 10^{-4} , and is therefore effectively indistinguishable from zero within the numerical accuracy of the finite-time Lyapunov exponent calculation. Consequently, LE_2 should be interpreted as an effectively neutral or marginal direction rather than as a genuinely contracting stable direction. Such a near-zero exponent is consistent with the neutral direction commonly associated with the temporal evolution of continuous-time trajectories. In contrast, the distinctly negative value $LE_3 = -0.482913$ represents a genuinely contracting direction. Therefore, the Lyapunov spectrum consists of one expanding direction, one effectively neutral direction, and one contracting direction, which provides a more precise characterization of the chaotic dynamics of the DNMHNN model.

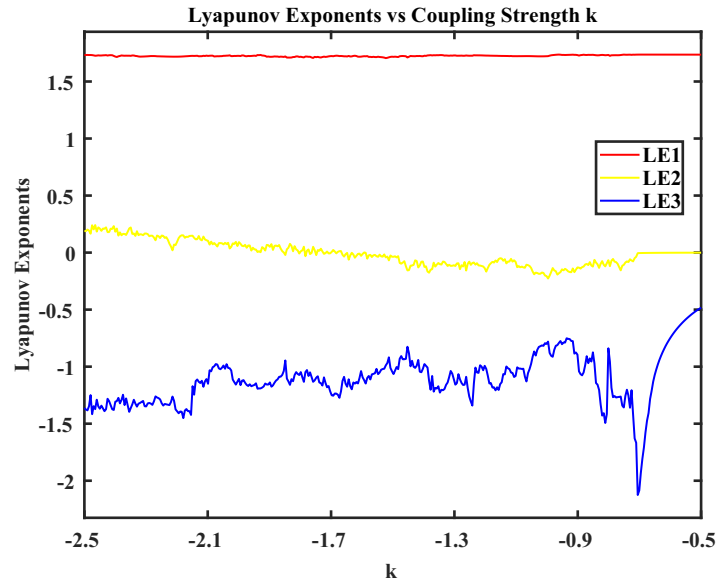


Figure 5: The Lyapunov exponents of model (5) vs. coupling strength k .

To ensure the reproducibility of the Lyapunov exponent calculation, the numerical procedure used in this work is described in detail as follows. The Lyapunov exponents of the DNMHNN model were computed using the standard Benettin method with periodic QR reorthonormalization. For each selected value of the coupling strength k , the original state equations and the corresponding variational equations were integrated simultaneously. The tangent matrix was initialized as the identity matrix, $Q_0 = I_3$, and was periodically reorthonormalized by QR decomposition during the integration process. At each reorthonormalization step, the diagonal elements of the upper triangular matrix R were recorded, and the Lyapunov exponents were obtained from the time-averaged logarithmic growth rates of these diagonal elements, namely

$$LE_i = \frac{1}{T - T_{tr}} \sum_{j=1}^{N_q} \ln |R_{ii}^{(j)}|, \quad i = 1, 2, 3, \quad (13)$$

where T is the total simulation time, T_{tr} is the transient time removed from the calculation, and N_q is the number of QR reorthonormalization steps after the transient interval.

The system was numerically integrated using the stiff solver `ode15s`, since the memristive nonlinear function and the square-wave excitation introduce nonsmooth and potentially stiff dynamics. The total simulation time was set to $T = 1000$. To eliminate the influence of the initial transient process, the first $T_{tr} = 200$ time units were discarded, and the Lyapunov exponents were accumulated only over the remaining interval

$[T_{tr}, T] = [200, 1000]$. The reorthonormalization interval was set to $\Delta t_{QR} = 1$, so that QR decomposition was performed at each sampling interval. The initial condition was fixed as $(x_1(0), x_2(0), x_3(0)) = (0.1, 0.1, 0.1)$ for all parameter values.

The Jacobian matrix required by the variational equations was evaluated along the numerical trajectory. For the smooth terms, the derivatives were calculated directly or equivalently approximated by finite differences. For the nonsmooth memristive terms containing the sign function, the derivative was evaluated piecewise away from the discontinuity points. The singular contributions exactly at the switching thresholds were not included in the state Jacobian, because these points form a set of measure zero in the time integration.

The non-autonomous square-wave excitation was treated as a prescribed time-dependent input. It is expressed as

$$I(t) = a_1 \operatorname{sgn}(\sin(\omega_1 t)) + a_2 \operatorname{sgn}(\sin(\omega_2 t)). \quad (14)$$

Since this excitation is independent of the state variables, it enters the variational equations only through its instantaneous value in the vector field and does not introduce additional state-dependent Jacobian terms. To handle the discontinuities of the square-wave excitation accurately, the integration interval was divided at the switching instants $t = \frac{m\pi}{\omega_1}, t = \frac{n\pi}{\omega_2}, m, n \in \mathbb{Z}$. The numerical solver was restarted at these switching instants, while the state variables and tangent vectors were kept continuous. This piecewise integration strategy avoids numerical interpolation across discontinuities and improves the reproducibility of the Lyapunov exponent spectra.

To investigate the stability and chaotic characteristics of the DNMHNN model, we analyzed its Lyapunov exponents and bifurcation diagrams while varying the memristor coupling strength within the range $-2.5 < k < -0.5$. This approach allowed us to examine how changes in the coupling parameter influence the system behavior, revealing transitions between stable and chaotic dynamics. The results, including both the bifurcation diagram and Lyapunov exponents for the DNMHNN model (5), are presented in Fig. 6, providing a comprehensive analysis of the network's dynamic behavior under parameter variation.

Fig. 6 shows that the DNMHNN model exhibits parameter-dependent transitions between periodic and chaotic dynamics as the coupling parameter k varies. The bifurcation diagram contains both chaotic intervals and periodic windows, indicating that the system does not follow a single monotonic transition from chaos to periodicity. In accordance with dynamical systems theory, successive period-doubling bifurcations provide a route from periodic motion to chaos, whereas reverse period-doubling, or period-halving bifurcations, provide a route from chaotic or high-period dynamics back to lower-period oscillations. Therefore, the observed bifurcation structure is more appropriately described as an alternation among periodic states, period-doubling transitions, chaotic regimes, and periodic windows. The corresponding Lyapunov exponent spectra support this interpretation, with positive values of the largest Lyapunov exponent identifying chaotic parameter intervals and values close to zero indicating periodic regimes. The associated Lyapunov exponent spectra, displayed in the same figure with blue, red, and yellow curves representing LE_1 , LE_2 , and LE_3 , respectively, provide a quantitative verification of the bifurcation diagram. Together, the bifurcation diagram and Lyapunov exponent spectra provide mutually consistent evidence for the parameter-dependent transitions between periodic and chaotic dynamics in the DNMHNN model.

The analysis of model (5) reveals not only the presence of periodic bifurcations but also the emergence of complex dynamic behaviors within the system. By systematically examining how variations in key parameters affect the network, the study provides a thorough understanding of the DNMHNN's system behavior. These insights support a detailed bifurcation analysis and dynamic analysis, highlighting the

interplay between different parameters and the resulting system dynamics, and offering a comprehensive perspective on the network's overall performance and stability.

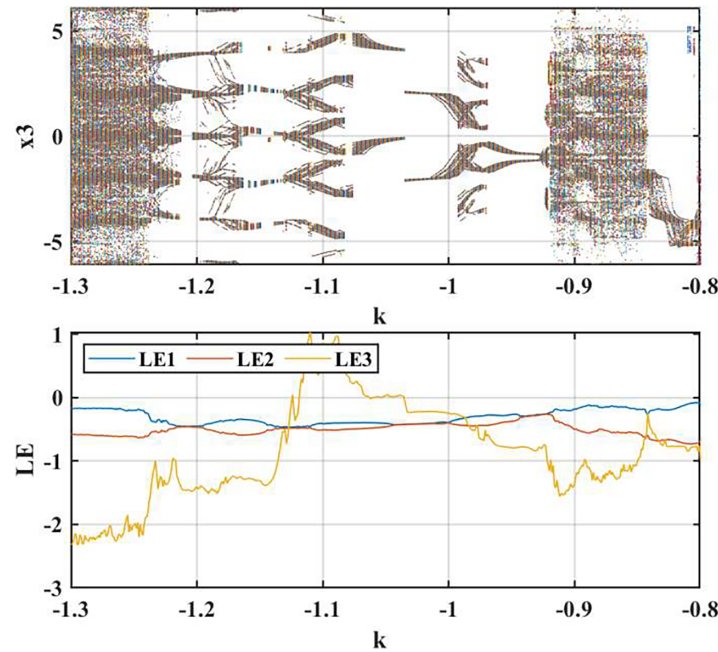


Figure 6: The bifurcation diagram (top) and Lyapunov exponents (LEs) (bottom) of the proposed DNMHNN model (5).

4 FPGA Implementation of the DNMHNN Model

With the rapid advancement of integrated circuits, modern FPGAs featuring high memory bandwidth, fast multiply-accumulate units, and versatile I/O peripherals support high-speed parallel processing. These capabilities make it possible to satisfy the intensive computation requirements, allowing the proposed DNMHNN model to be efficiently implemented on an FPGA platform.

The AX7020 development board, based on the Xilinx Zynq-7000 SoC, integrates a dual-core ARM Cortex-A9 processor with FPGA programmable logic, providing a versatile platform for implementing complex neural network models such as the DNMHNN. Equipped with rich peripheral interfaces, including DDR3 memory, QSPI Flash, DAC modules, and HDMI output, the board facilitates high-speed data acquisition, processing, and visualization. Complementing the FPGA, a dual-channel 14-bit DAC module (AN9767) enables high-speed digital-to-analog conversion, supporting precise output signal generation for real-time observation and measurement. The entire design and deployment process is managed using Xilinx Vivado, which allows for efficient synthesis, implementation, and debugging of FPGA logic, facilitating real-time parallel processing, accurate timing control, and seamless hardware-software co-design. Together, this combination of AX7020, AN9767, and Vivado ensures high-speed, accurate, and reliable execution of memristive neural network models.

In the experimental setup, a 50 MHz clock provided by the PS system, corresponding to a clock period of 20 ns, was employed. Two memory blocks were reserved for storing the digitized signals after each iteration. Once the iterations were completed, the stored digital signals were converted to analog using the AN9767 chip, with an output voltage range of $-5V$ to $+5V$. The resulting analog signals were then directly fed to an oscilloscope through the BNC output interface for real-time observation.

The FPGA implementation experiments of DNMHNN model (5) are presented in Fig. 7. Fig. 7a1–d1 illustrates the FPGA hardware configurations corresponding to different dynamical trajectories of the DNMHNN model, while Fig. 7a2 shows the dynamic trajectory of the DNMHNN model on the X–Y orthogonal plane. Fig. 7b2–d2 displays the dynamical trajectories under the conditions of $M = -2$, $N = 1$ and $M = 2$, $N = -3$ and $M = 3$, $N = -6$, respectively. To further provide quantitative support for the FPGA feasibility, the main implementation performance is summarized in Table 3.

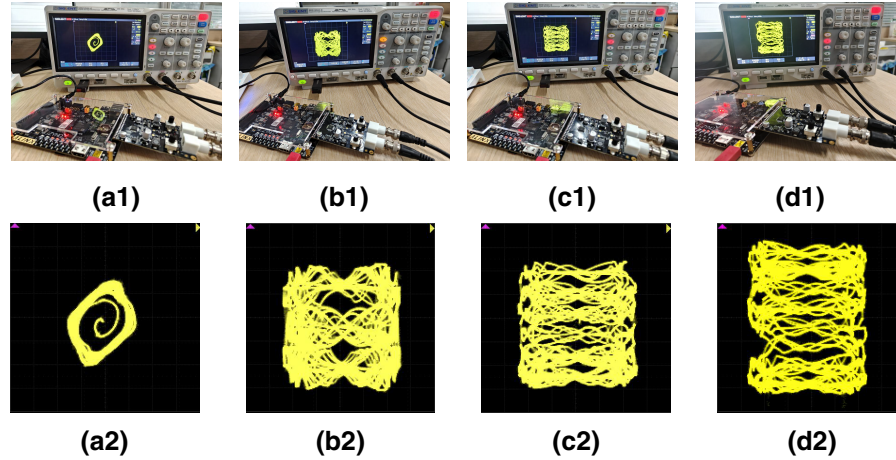


Figure 7: FPGA implementation and oscilloscope observations of the DNMHNN model. (a1–d1) FPGA experimental setups corresponding to the dynamical trajectories shown in (a2–d2), respectively. (a2) Phase trajectory of the DNMHNN model on the X–Y plane. (b2–d2) Oscilloscope-observed phase trajectories for $(M, N) = (-2, 1)$, $(2, -3)$, and $(3, -6)$, respectively.

Table 3: FPGA implementation performance.

Metric	Result
Resource Utilization	LUT: 28.46%, FF: 12.37%, BRAM: 9.29%, DSP: 22.73%
Timing Closure	>0 at 50 MHz
Power Consumption	0.384 W
Throughput	0.186 M Samples/s
Numerical Error	MAE $\approx 10^{-5}$

As shown in Table 3, the FPGA implementation satisfies timing closure under the original 50 MHz clock setting, while the resource utilization and power consumption remain acceptable for the Zynq-7000 platform. Moreover, the small numerical error between the numerical and FPGA outputs indicates that the hardware implementation can reproduce the DNMHNN dynamics. Therefore, the FPGA results are supported not only by qualitative waveform agreement, but also by quantitative resource, timing, power, throughput, and error-performance evaluations. By mapping the complicated mathematical operations of the chaotic system into FPGA-based hardware logic, this study not only confirms the practicality of the proposed design in physical realization, but also establishes a feasible technical foundation for engineering applications such as chaotic signal generation and secure communication modules.

5 The DNMHNN-Based Image Encryption Scheme

Due to the rich dynamic and memory properties of the above proposed DNMHNN model, a DNMHNN-based image encryption algorithm is designed in this section.

5.1 The Designed DNMHNN-Based Image Encryption Algorithm

Based on the above DNMHNN model, a DNMHNN-based image encryption algorithm is designed and presented in Fig. 8. The encryption system comprises four functional blocks: (1) chaotic sequence generator; (2) permutation module; (3) substitution module; (4) bidirectional ciphertext-feedback diffusion module. The methodology for both encryption and decryption is presented in the following description.

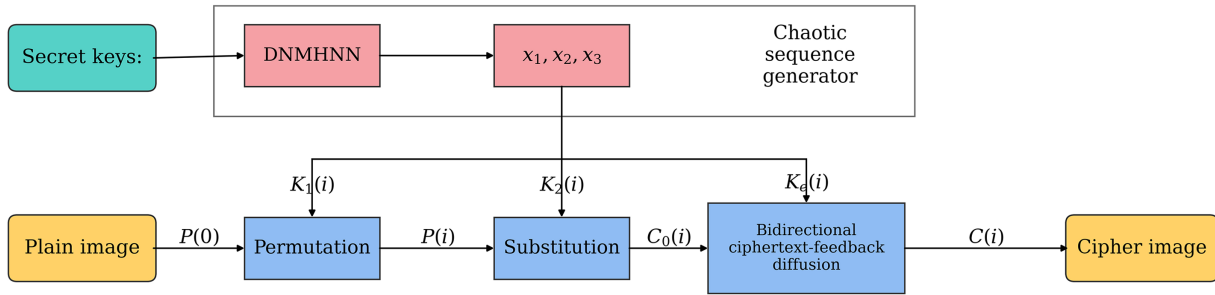


Figure 8: Flowchart of the designed DNMHNN-based image encryption algorithm.

Step 1: Chaotic sequence generator. Pseudorandom sequence generation for the image encryption process begins by reading a color plaintext image P_0 in RGB format to determine its fundamental attributes: height (H), width (W), and number of channels (C). A key composed of 13 parameters ($a, b, d, k, M, N, a_1, a_2, \omega_1, \omega_2, w_{11}, w_{21},$ and w_{22}) is then applied to the DNMHNN model to produce three chaotic sequences (x_1, x_2, x_3) of length $H \times W$. Among them, M and N are integer branch-control parameters, while the remaining 11 parameters are real-valued parameters. Since real-valued parameters cannot be regarded as having infinite precision in digital implementation, the effective key space is estimated under finite precision. Assuming that the effective precision of each real-valued parameter is 10^{-12} , the key space is still much larger than 2^{128} , which is sufficient to resist brute-force attacks. To further enhance sequence randomness and security, these sequences undergo additional N iterations following an initial N_0 pre-iterations phase. Finally, the three chaotic sequences are mathematically transformed into two pseudorandom sequences, $K_1(i)$ and $K_2(i)$, ensuring high complexity and unpredictability for secure image encryption.

$$K_1(i) = \text{mod} \left(\left\lfloor \frac{(x_1(i) + x_2(i) + x_3(i) + 1) \cdot 10^6}{(x_1(i) + x_2(i) + x_3(i)) \cdot 10^3} \right\rfloor, \text{height} \right) \quad (15)$$

$$K_2(i) = \text{mod} \left(\left\lfloor \frac{x_1(i) + 2x_2(i) + 3x_3(i)}{3} \cdot 10^{10} \right\rfloor, 256 \right) \quad (16)$$

It should be noted that K_1 is not directly used as the pixel-position index, but as a sorting key for generating the permutation vector. The actual permutation vector is obtained as

$$P = \text{argsort}(K_1). \quad (17)$$

Although the modulo operation with respect to the image height may lead to repeated values in K_1 , the sorting operation returns the corresponding index sequence rather than the repeated values themselves. For

equal values in K_1 , the original pixel index is used as a deterministic tie-breaking rule, which is equivalent to sorting the pairs $(K_1(i), i)$. Therefore, each pixel index appears exactly once in P , and the permutation is a one-to-one mapping. During decryption, the inverse permutation P^{-1} is constructed from the same P , ensuring that the permutation process is reversible.

Step 2: Permutation module. This module is designed to create permutation boxes for image scrambling. The chaotic key sequence $K_1(i)$ is first sorted in ascending order to generate an index sequence. Using this index sequence, the plaintext image $P(0)$ is globally rearranged to produce the permuted sequence $P(i)$, effectively shuffling the pixels of the original image according to $K_1(i)$. The permutation method is detailed in Eq. (18).

$$P_1(i) = P(\text{index}(K_1(i))) \quad (18)$$

Step 3: Substitution module. This module performs pixel substitution on the permuted image. The sequence $K_2(i)$ is used to encrypt the permuted image through an XOR operation, denoted by $\oplus$, as follows. The substitution method is detailed in Eq. (19). The DNMHNN iterates until all pixels are processed. After the XOR substitution operation, an intermediate cipher image C_0 is obtained. To further improve the avalanche effect and plaintext sensitivity, C_0 is subsequently processed by a bidirectional ciphertext-feedback diffusion module to generate the final cipher image.

$$C(i) = P(i) \oplus K_2(i) \quad (19)$$

Step 4: Bidirectional ciphertext-feedback diffusion module. To improve the diffusion capability of the proposed encryption scheme, a bidirectional ciphertext-feedback diffusion stage is added after the permutation-XOR substitution operation. Let C_0 denote the intermediate cipher image obtained from Eq. (19). The image C_0 is first converted into a one-dimensional sequence $V = \{V(i)\}_{i=1}^{N_e}$, where $N_e = H \times W \times C$. The auxiliary diffusion key stream $K_e(i)$ is generated from $K_2(i)$ and the pixel position index as

$$K_e(i) = \text{mod}(K_2(i) + 37(i-1) + 101 \text{mod}((i-1)^2, 257), 256). \quad (20)$$

The forward diffusion process is defined as

$$F(1) = V(1) \oplus K_e(1), \quad (21)$$

$$F(i) = \text{mod}(V(i) + K_e(i) + F(i-1), 256) \oplus K_e(i), \quad i = 2, 3, \dots, N_e. \quad (22)$$

Then, the backward diffusion process is performed as

$$C(N_e) = F(N_e) \oplus K_e(N_e), \quad (23)$$

$$C(i) = \text{mod}(F(i) + K_e(i) + C(i+1), 256) \oplus K_e(i), \quad i = N_e - 1, \dots, 1. \quad (24)$$

Finally, the one-dimensional sequence C is reshaped into an RGB image to obtain the final cipher image. During decryption, the inverse backward diffusion and inverse forward diffusion operations are performed first, and then the original inverse XOR substitution and inverse permutation operations are applied to recover the plaintext image.

5.2 Security Analysis of the DNMHNN-Based Encryption Algorithm

5.2.1 Key Space Analysis

In the experiment, the DNMHNN model was operated with the parameter values $a = 0.55$, $b = 1.3$, $d = 5.7$, $k = -1.63$, $M = 2$, $N = -3$, $a_1 = 3.5$, $a_2 = 2.8$, $\omega_1 = 2.1$, $\omega_2 = 3.3$, $w_{11} = 2.2$, $w_{21} = 3.5$, and $w_{22} = 1.5$. The initial transient was removed by setting $N_0 = 10,000$, and the effective chaotic sequence was extracted beginning from the 10,001st iteration of the DNMHNN system. The iteration step size was fixed at 0.01, with the total evolution time taken as 3000. Meanwhile, a 512×512 RGB image was adopted as the test object for the encryption procedure, and an image was selected to evaluate the encryption performance through a histogram comparison before and after encryption. As shown in Fig. 9, Fig. 9a gives the original image, Fig. 9b presents its histogram, Fig. 9c shows the encrypted result, and Fig. 9d corresponds to the histogram of the encrypted image, where the red, green, and blue components represent the R, G, and B channels of the color image, respectively.

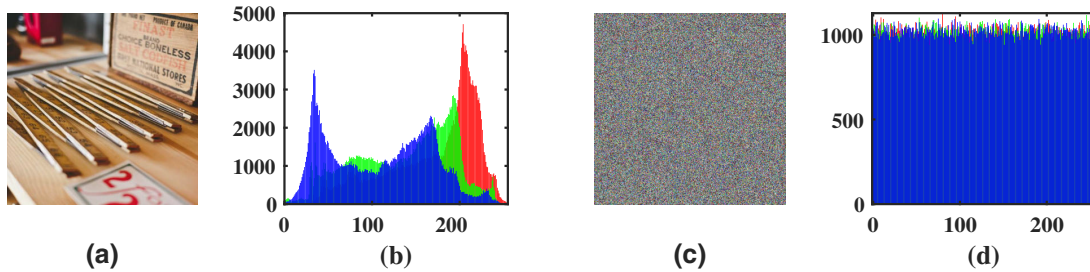


Figure 9: Encryption results and histogram analysis. (a) Plaintext image. (b) Histogram of the plaintext image. (c) Cipher image. (d) Histogram of the cipher image. The red, green, and blue curves in (b,d) represent the R, G, and B channels, respectively.

As shown in Fig. 9, the histogram of the encrypted image exhibits a near-uniform distribution, with the intensity levels in each color channel occurring with approximately equal frequencies, which demonstrates that the encryption process achieves effective pixel scrambling. Such a distribution significantly weakens the usefulness of statistical analysis and reduces the feasibility of a histogram attack by an adversary. Therefore, the encrypted image shows enhanced security, since the histogram reveals little exploitable information about the original image or the underlying encryption process.

Histogram variance is used to quantitatively evaluate the uniformity of the image histogram. For a well-encrypted image, the occurrence frequencies of different intensity levels should be approximately equal, resulting in a small histogram variance. Therefore, a smaller histogram variance indicates a more uniform cipher-image histogram and stronger resistance to histogram-based statistical attacks.

In our experiments, the histogram variance was calculated for each RGB channel before and after encryption, and the corresponding results are summarized in Table 4. As shown in Table 4, the histogram variances of the R, G, and B channels decrease from 961,108.72, 455,944.37, and 507,213.75 in the plaintext image to 859.73, 1104.91, and 1002.39 in the cipher image, respectively. This substantial reduction indicates that the occurrence frequencies of the different intensity levels become much more uniform after encryption. The quantitative results are consistent with the near-uniform cipher-image histogram shown in Fig. 9. Therefore, the proposed encryption algorithm effectively suppresses the statistical characteristics of the plaintext image and improves its resistance to histogram-based attacks.

Table 4: Histogram variance of the plaintext and cipher images.

Image	R	G	B
Plain image	961,108.72	455,944.37	507,213.75
Cipher image	859.73	1104.91	1002.39

In addition to histogram variance, MD, ID, and UHD are further calculated to evaluate the encryption/decryption quality from the viewpoint of plaintext-cipher deviation and cipher-histogram uniformity. MD measures the deviation between the plaintext image and the cipher image, ID reflects the irregularity of the deviation distribution, and UHD measures the deviation between the cipher-image histogram and the ideal uniform histogram. As shown in Table 5, the average MD, ID, and UHD values are 261,617.5000, 150,554.0000, and 0.025927, respectively. The large MD value indicates that the cipher image is significantly different from the plaintext image, while the small UHD value shows that the cipher-image histogram is close to a uniform distribution. These results further verify the good encryption quality and statistical attack resistance of the proposed encryption algorithm.

Table 5: Histogram variance.

Channel	MD	ID	UHD
R	261,596.5000	137,534.0000	0.025764
G	261,645.0000	160,400.0000	0.025864
B	261,611.0000	153,728.0000	0.026154
Average	261,617.5000	150,554.0000	0.025927

5.2.2 Correlation Analysis

In a plaintext image, adjacent pixels generally exhibit strong pixel correlation in the horizontal direction, vertical direction, and diagonal direction because neighboring regions usually contain similar structural and intensity information. Such inherent statistical dependence provides potential clues about the image content and may be exploited by adversaries to perform cryptanalytic attacks or infer the encryption key. For this reason, an effective and secure encryption scheme should thoroughly destroy these correlations during the encryption process, so that the adjacent pixels in the encrypted image become nearly independent of one another. By significantly reducing the pixel correlation in all principal directions, the encrypted image can better resist statistical analysis and offer stronger protection for confidential visual information.

A total of 10,000 neighboring pixel pairs were randomly sampled, and their correlation distributions along the horizontal, vertical, and diagonal directions were depicted in Fig. 10. In Fig. 10, the red, green, and blue correspond to the R, G, and B channels, respectively.

As shown in Fig. 10, the adjacent pixels in the plaintext images demonstrate strong correlations across all three channels and directions, whereas pixels in the encrypted image are randomly distributed across different channels and directions. This result demonstrates that the encryption algorithm effectively eliminates the inter-pixel correlation inherent in the original image. Consequently, adversaries encounter much greater difficulty in performing image recovery or extracting useful information, which further improves the overall security of the encryption scheme.

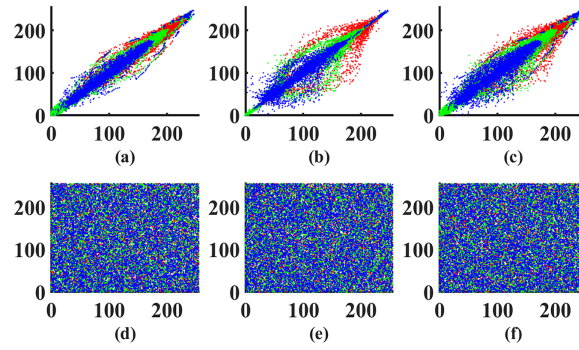


Figure 10: Image correlation analysis. (a–c) denote the horizontal, vertical, and diagonal correlations of the plaintext images, respectively. (d–f) represent the horizontal, vertical, and diagonal correlations of the encrypted images, respectively.

To further quantitatively analyze the correlation between the plaintext and encrypted images, the dependence among adjacent pixels is quantitatively examined by using Pearson's correlation coefficient and the distance correlation coefficient. The analysis is carried out for the original image and the encrypted image in the horizontal direction, vertical direction, and diagonal direction. In general, the adjacent pixels of the original image exhibit strong linear dependence, so their correlation values are usually close to 1, whereas the corresponding values of the encrypted image should be close to 0, indicating that the encryption process effectively weakens the statistical relationship between neighboring pixels. The formula for calculating the Pearson correlation coefficient is as follows:

$$r_{xy} = \frac{\text{Cov}(x, y)}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (25)$$

The computed Pearson correlation coefficients for the aforementioned image are summarized in Table 6. As can be observed from the data in Table 6, the correlation coefficients of the plaintext image remain extremely close to 1, indicating a strong statistical dependence between adjacent pixels in the original image. In contrast, the corresponding coefficients of the encrypted image are all very close to 0, which suggests that the encryption process effectively destroys the inherent linear correlation among neighboring pixels. This clear difference between the plaintext and encrypted results further demonstrates that the proposed encryption algorithm possesses a strong capability to resist correlation-based statistical analysis, thereby confirming its high level of security.

Table 6: Pearson correlation coefficients of the plain image and cipher image.

Image	Channels	Horizontal	Vertical	Diagonal
Plain image	R	0.9902	0.9780	0.9753
	G	0.9910	0.9801	0.9769
	B	0.9864	0.9755	0.9689
Cipher image	R	−0.0164	0.0011	−0.0146
	G	0.0030	0.0011	−0.0173
	B	−0.0032	0.0044	−0.0098

5.2.3 Information Entropy Analysis

Information entropy is a fundamental concept in information theory used to measure the uncertainty, randomness, or average information content of a data source. It was originally proposed by Claude Shannon and is widely applied in communication systems, signal processing, data compression, and cryptography. In the context of image encryption, information entropy is commonly used to evaluate the randomness of pixel distributions in an encrypted image. A higher entropy value indicates that the pixel values are distributed more uniformly and randomly, which generally implies stronger resistance against statistical attacks. For a discrete random variable with possible outcomes x_i and corresponding probabilities $p(x_i)$, the information entropy is defined as

$$H(X) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i). \quad (26)$$

For an 8-bit RGB image, each color channel contains 256 possible intensity levels, and the ideal information entropy of each channel is 8 bits per channel when all intensity levels have equal probability. Therefore, when the entropy of an encrypted image approaches the theoretical ideal value, the encryption scheme is usually considered to exhibit good security performance. The calculated information entropy of the aforementioned image is presented in Table 7. As can be seen from Table 7, the entropy value of the encrypted image is highly close to the theoretical optimal maximum of 8, which indicates that the encryption algorithm effectively increases the randomness of the image and correspondingly lowers its predictability.

Table 7: Information entropy of the RGB plaintext and cipher images.

Image	R (bits/pixel)	G (bits/pixel)	B (bits/pixel)	Mean (bits/pixel)
Plain image	7.4048	7.6487	7.6212	7.5582
Cipher image	7.9994	7.9992	7.9993	7.9993

5.2.4 Robustness Analysis and Quantitative Evaluation

Robustness analysis refers to the evaluation of whether a system, model, or algorithm can maintain stable performance and reliable functionality when subjected to disturbances, uncertainties, noise, parameter variations, or partial data loss. In the context of image encryption, robustness analysis is commonly used to examine whether the encrypted or decrypted image can still preserve acceptable visual quality and information integrity under attacks such as noise contamination, cropping, data loss, or transmission errors. A robust encryption scheme should not only provide high security under ideal conditions, but also retain a certain level of resistance and recovery capability in non-ideal environments. Therefore, robustness analysis serves as an important criterion for assessing the practical applicability and stability of an encryption algorithm.

To further evaluate the robustness of the proposed DNMHNN-based image encryption algorithm, quantitative metrics are introduced in addition to visual inspection. In this work, the decrypted image obtained from the attacked cipher image is compared with the original plaintext image. Four commonly used image quality indicators are adopted: mean square error (MSE), peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and recovered-image correlation coefficient. MSE measures the pixel-level error between the original and recovered images, PSNR evaluates the reconstruction quality in logarithmic scale, SSIM evaluates the structural similarity between two images, and the recovered-image

correlation coefficient measures their linear similarity. Lower MSE and higher PSNR, SSIM, and correlation values indicate better recovery quality and stronger robustness.

For the original plaintext image P and the recovered image R obtained after the cipher image is subjected to an attack, the MSE and PSNR are defined as follows:

$$\text{MSE} = \frac{1}{H \times W \times C} \sum_{i=1}^H \sum_{j=1}^W \sum_{k=1}^C [P(i, j, k) - R(i, j, k)]^2, \quad (27)$$

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right). \quad (28)$$

The recovered-image correlation coefficient is calculated as follows:

$$\text{Corr}(P, R) = \frac{\sum_{i=1}^H \sum_{j=1}^W \sum_{k=1}^C (P(i, j, k) - \bar{P})(R(i, j, k) - \bar{R})}{\sqrt{\sum_{i=1}^H \sum_{j=1}^W \sum_{k=1}^C (P(i, j, k) - \bar{P})^2} \sqrt{\sum_{i=1}^H \sum_{j=1}^W \sum_{k=1}^C (R(i, j, k) - \bar{R})^2}}. \quad (29)$$

In addition, SSIM is adopted to evaluate the structural similarity between P and R . These metrics provide a more comprehensive robustness evaluation than visual observation alone.

Fig. 11 displays the decryption results after occluding 1/25, 1/16, and 1/4 of the encrypted images. Fig. 12 illustrates the decryption outcomes after adding varying intensities of noise to the encrypted images. Fig. 13 presents the decryption results after randomly tampering with 10%, 30%, and 50% of the pixel values in the encrypted images. In addition to the visual results, the corresponding quantitative robustness results are listed in Tables 8–10. As shown in these tables, the recovery quality decreases markedly as the attack intensity increases. Measurable similarity is retained under some partial-damage conditions, particularly under occlusion; however, severe noise and tampering result in substantial reconstruction degradation. Therefore, the recovery performance of the proposed scheme is dependent on the attack type and intensity.

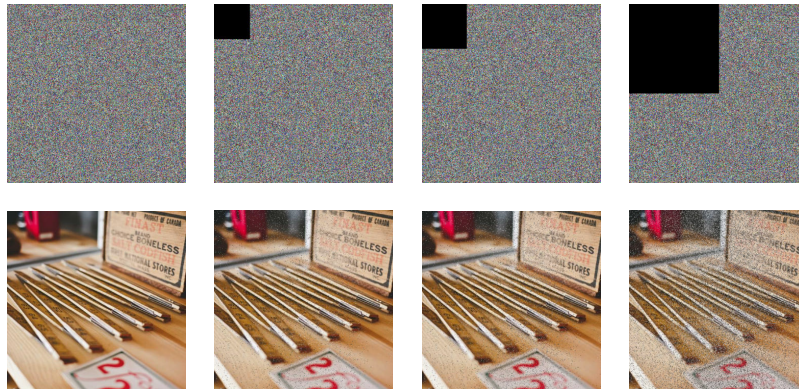


Figure 11: Decryption results after masking 1/25, 1/16, and 1/4 of the encrypted image.

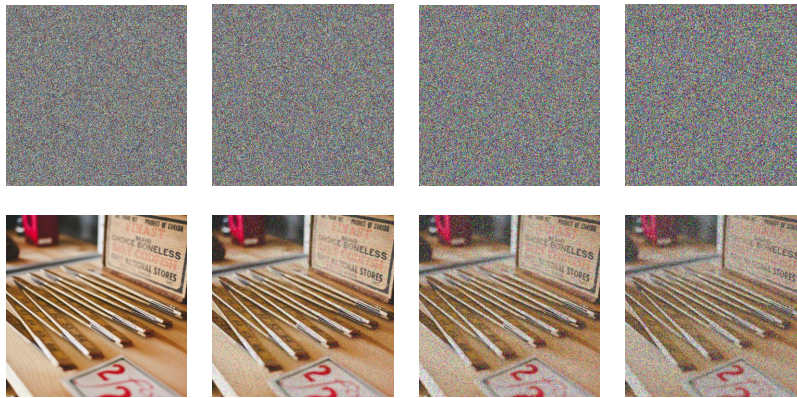


Figure 12: Decryption results after adding noise of varying intensities to the encrypted image.

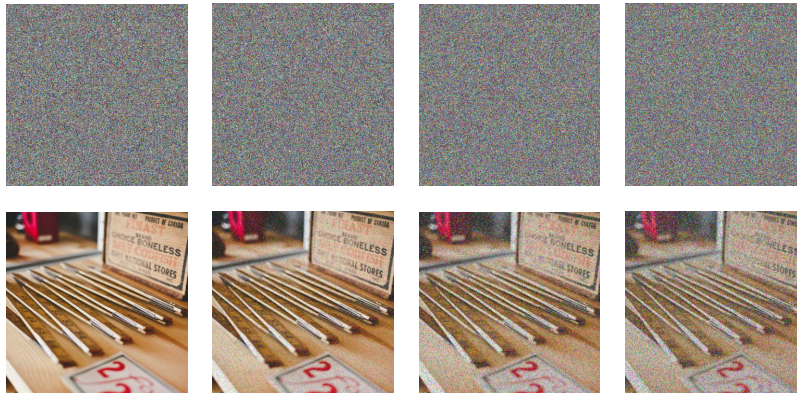


Figure 13: Decryption results after randomly tampering with 10%, 30%, and 50% of the pixel values in the encrypted image.

Table 8: Quantitative robustness results under occlusion attacks.

Attack Level	MSE	PSNR (dB)	SSIM	Correlation
1/25	374.4	22.397	0.67575	0.95021
1/16	592.39	20.405	0.571	0.92178
1/4	2333.7	14.45	0.22571	0.70697

Table 9: Quantitative robustness results under noise attacks.

Noise Density	MSE	PSNR (dB)	SSIM	Correlation
0.1	2410.7	14.309	0.18754	0.6973
0.3	5524.7	10.708	0.07033	0.35792
0.5	7219.8	9.5455	0.038653	0.19182

Table 10: Quantitative robustness results under tampering attacks.

Tampering Density	MSE	PSNR (dB)	SSIM	Correlation
10%	2404.6	14.32	0.18832	0.69786
30%	5517.1	10.714	0.070394	0.35875
50%	7217.7	9.5468	0.038841	0.19164

5.2.5 Differential Attack and Key Sensitivity Analyses

In image encryption, a secure cryptosystem should be highly sensitive to slight changes in the plaintext and secret key. To evaluate the resistance of the proposed scheme against differential attacks, the number of pixels change rate (NPCR) and the unified average changing intensity (UACI) are adopted. NPCR measures the percentage of different pixels between two cipher images, while UACI measures the average intensity difference between them.

For two cipher images C_1 and C_2 with size $H \times W \times C$, the difference matrix $D(i, j, k)$ is defined as

$$D(i, j, k) = \begin{cases} 0, & C_1(i, j, k) = C_2(i, j, k), \\ 1, & C_1(i, j, k) \neq C_2(i, j, k). \end{cases} \quad (30)$$

Then, NPCR and UACI are calculated as

$$\text{NPCR} = \frac{\sum_{i=1}^H \sum_{j=1}^W \sum_{k=1}^C D(i, j, k)}{H \times W \times C} \times 100\%, \quad (31)$$

$$\text{UACI} = \frac{1}{H \times W \times C} \sum_{i=1}^H \sum_{j=1}^W \sum_{k=1}^C \frac{|C_1(i, j, k) - C_2(i, j, k)|}{255} \times 100\%. \quad (32)$$

To visually illustrate the effect of the added bidirectional ciphertext-feedback diffusion stage, the encryption and decryption results are presented in Fig. 14. The original permutation-XOR algorithm can produce a noise-like cipher image, while the final cipher image after the external diffusion stage exhibits stronger visual randomness. When only one pixel of the plaintext image is changed, the corresponding final cipher image becomes completely different from the original final cipher image. In addition, when the perturbed key is used for decryption, the recovered image is noise-like and contains no meaningful visual information. These visual results are consistent with the NPCR, UACI, and key sensitivity results reported in Tables 11 and 12.

For an 8-bit random image, the ideal reference values of NPCR and UACI are approximately 99.6094% and 33.4635%, respectively. Values close to these references indicate strong diffusion performance and high resistance to differential attacks.

First, plaintext sensitivity was tested. Only one pixel of the plaintext image was modified by increasing its value by 1, while the secret key remained unchanged. The original plaintext image P and the slightly modified image P' were encrypted using the same key K , producing two final cipher images $C = E_{\text{ext}}(P, K)$ and $C' = E_{\text{ext}}(P', K)$, respectively. The NPCR and UACI values between C and C' are shown in Table 11. The results indicate that a one-pixel change in the plaintext image causes a significant change in the final cipher image, demonstrating that the revised encryption scheme has strong plaintext sensitivity.

Second, encryption key sensitivity was tested. The original key K was generated using the initial condition $(0.1, 0.1, 0.1)$, while the perturbed key K_δ was generated by changing only the first initial value to $0.1 + 10^{-14}$. The same plaintext image was encrypted using K and K_δ , producing $C = E_{\text{ext}}(P, K)$ and $C_\delta = E_{\text{ext}}(P, K_\delta)$, respectively. The NPCR and UACI values between C and C_δ are listed in Table 12.

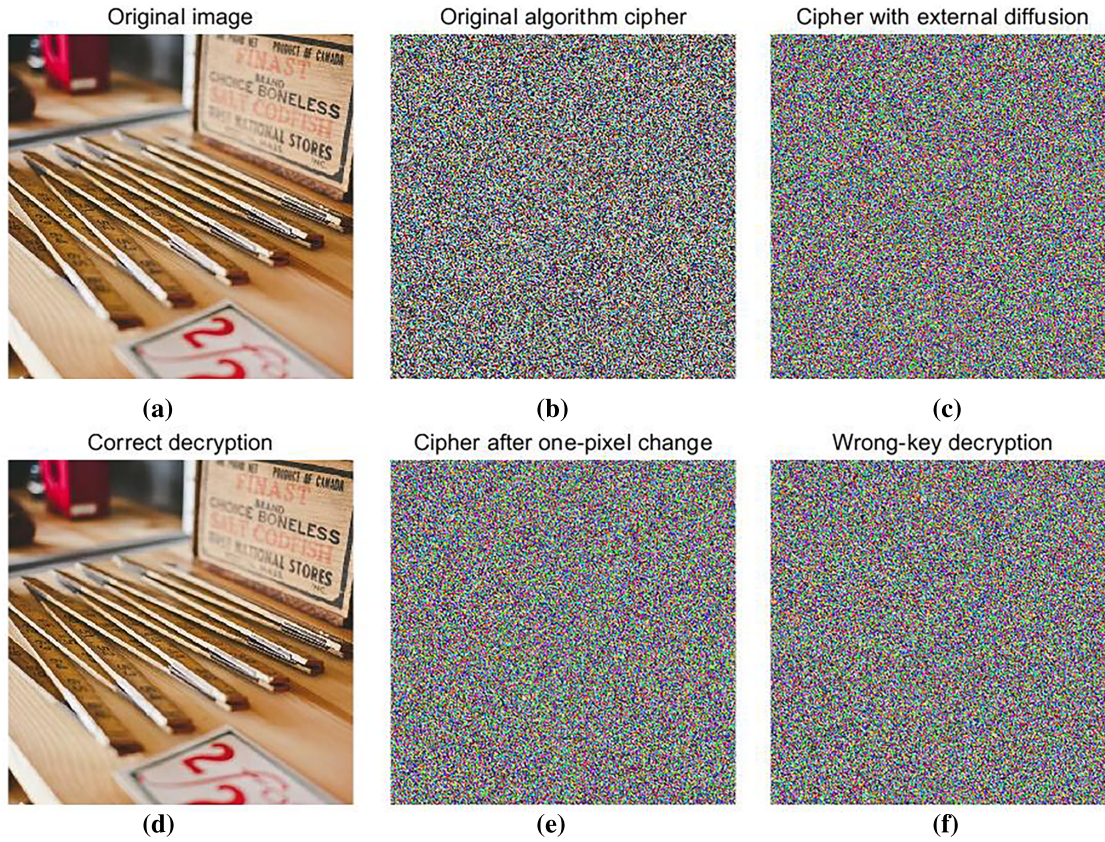


Figure 14: Visual results of the external diffusion and key sensitivity tests. (a) Original image. (b) Cipher image generated by the original permutation-XOR algorithm. (c) Final cipher image after bidirectional ciphertext-feedback diffusion. (d) Correctly decrypted image. (e) Final cipher image after one-pixel plaintext change. (f) Wrong-key decrypted image.

Table 11: NPCR and UACI results under one-pixel plaintext change.

Channel	NPCR (%)	UACI (%)
R	99.8417%	33.3111%
G	99.6201%	33.5196%
B	99.6326%	33.5496%
Average	99.6981%	33.4601%

Table 12: NPCR and UACI results for key sensitivity analysis.

Test Item	Channel	NPCR (%)	UACI (%)
C vs. C_δ	R	99.6243%	33.4721%
C vs. C_δ	G	99.6059%	33.4232%
C vs. C_δ	B	99.5937%	33.4941%
C vs. C_δ	Average	99.6080%	33.4631%
P vs. P_{wrong}	R	99.6151%	32.0728%
P vs. P_{wrong}	G	99.5979%	29.9497%
P vs. P_{wrong}	B	99.6117%	30.8806%
P vs. P_{wrong}	Average	99.6082%	30.9677%

Third, decryption key sensitivity was evaluated. The correct cipher image C was decrypted using the perturbed key K_δ , producing the wrong-key decrypted image P_{wrong} . The image P_{wrong} was then compared with the original plaintext image P . The corresponding NPCR and UACI values are listed in Table 12, while the MSE and PSNR values for wrong-key decryption are reported in Table 13. The wrong-key decrypted image contains no meaningful visual information, indicating that even an extremely small deviation in the secret key prevents successful decryption.

Table 13: MSE and PSNR results for wrong-key decryption.

Test Item	Channel	MSE	PSNR (dB)
P vs. P_{wrong}	Average	9303.1469	8.4445

The above results verify that the revised DNMHNN-based image encryption algorithm is highly sensitive to both plaintext and secret key changes. A tiny modification in the plaintext or a 10^{-14} -level perturbation in the initial condition of the chaotic system leads to a completely different final cipher image. Moreover, the correct plaintext cannot be recovered with an incorrect key. Therefore, the proposed encryption scheme has strong resistance to differential attacks and key sensitivity attacks.

5.2.6 Comparison with Existing Image-Encryption Schemes

To further evaluate the performance of the proposed DNMHNN-based encryption scheme, its principal security indicators are compared with those of several recently reported image-encryption algorithms. Table 14 presents a comparison in terms of information entropy, NPCR, and UACI. For an 8-bit image, the ideal values of information entropy, NPCR, and UACI are 8, 99.6094%, and 33.4635%, respectively. The proposed scheme achieves an average information entropy of 7.9993, an NPCR of 99.6981%, and a UACI of 33.4601%. The absolute deviation from the ideal entropy value is only 0.0007 bits/pixel, while the deviation from the ideal UACI value is 0.0034 percentage points. These results indicate that the proposed scheme provides a nearly uniform cipher-pixel distribution and a strong avalanche effect.

As shown in Table 14, the NPCR and UACI values of the proposed scheme are comparable to those reported for recent hyperchaos-, dynamic-DNA-, and fractional-chaos-based encryption schemes. In particular, the proposed UACI value is very close to the theoretical reference of 33.4635%. Although the proposed NPCR is slightly higher than the theoretical expectation, it remains within the range commonly reported for

secure image-encryption algorithms. Therefore, a one-pixel modification in the plaintext can propagate to almost all pixels in the final cipher image.

Table 14: Comparison of differential-attack and statistical-security metrics with existing image-encryption schemes.

Scheme	Test Configuration	Entropy	NPCR (%)	UACI (%)
Proposed DNMHNN	512 × 512 RGB image	7.9993	99.6981	33.4601
Zhang et al. [31]	256 × 256 Cameraman	7.9979	99.6111	33.4643
Qiu et al. [38]	256 × 256 Cameraman	N/R	99.6078	33.5237
Afify et al. [39]	256 × 256 Cameraman	N/R	99.6400	33.3900
Chang et al. [40]	256 × 256 Cameraman	N/R	99.6017	33.3855
Ideal value	8-bit random image	8.0000	99.6094	33.4635

Note: N/R denotes that the corresponding metric was not reported in the cited study. Since different schemes use different test images, resolutions, and color modes, the comparison is indicative rather than absolutely controlled.

Table 15 further compares the reconstruction quality under ciphertext cropping or occlusion attacks. Because PSNR and SSIM are affected by image content, image resolution, color-channel processing, cropping position, and the diffusion structure, only results obtained under approximately the same ciphertext-loss ratio are included. The PSNR of the proposed scheme is 14.4500 dB when one quarter of the cipher image is occluded, which is comparable to the values of 14.4552–14.6848 dB reported by the comparison schemes. Moreover, the proposed scheme provides an SSIM of 0.22571 and a recovered-image correlation coefficient of 0.70697, confirming that recognizable structural information can still be recovered after 25% ciphertext loss.

Table 15: Comparison of reconstruction quality under ciphertext cropping or occlusion attacks.

Scheme	Image and Size	Loss Ratio	PSNR (dB)	SSIM
Proposed DNMHNN	512 × 512 RGB test image	1/4	14.4500	0.22571
Zhang et al. [31]	Cameraman, 256 × 256	1/4	14.4552	N/R
Kumar and Sharma [41]	Cameraman, 512 × 512	1/4	14.6359	N/R
Zhang et al. [42]	Goldhill, 512 × 512	1/4	14.6848	N/R

Note: N/R denotes that SSIM was not reported under the corresponding cropping condition. Differences in image content, resolution, color mode, and cropping pattern should be considered when interpreting the comparison.

Most of the compared studies report PSNR but do not provide SSIM under the same cropping condition. Therefore, the unavailable SSIM entries are denoted by “N/R” rather than being inferred from PSNR. This distinction is important because PSNR measures pixel-level error, whereas SSIM evaluates structural similarity, and the two indicators are not interchangeable. Overall, the comparative results show that the proposed scheme provides competitive robustness while additionally offering quantitative structural-similarity and recovered-correlation evaluations.

6 Conclusion

In this paper, a novel three-dimensional dual-neuron memristive Hopfield neural network (DNMHNN) with a controllable number and distribution of equilibrium points has been constructed by incorporating a nonlinear memristor into the synaptic coupling of a classical Hopfield neural network. The proposed model exhibits rich nonlinear characteristics and complex dynamical behaviors due to the state-dependent properties of the memristor and the multi-threshold non-smooth function. Through equilibrium analysis

and stability investigation, it is demonstrated that the system possesses multiple coexisting equilibrium points, all of which are unstable saddle points, thereby providing a theoretical basis for the emergence of complex dynamics.

Comprehensive dynamical analyses, including phase-space trajectories, bifurcation diagrams, and Lyapunov exponent spectra, have been conducted to explore the system behavior under different parameter conditions. The results reveal that the DNMHNN model exhibits chaotic dynamics, period-doubling bifurcations, and transitions between ordered and disordered states. In particular, the coexistence of positive and negative Lyapunov exponents confirms the presence of mixed dynamic characteristics, highlighting the intrinsic complexity and sensitivity of the system to initial conditions and parameter variations.

Furthermore, the feasibility of hardware implementation has been verified by deploying the DNMHNN model on an FPGA platform. The experimental results obtained from the hardware system are highly consistent with the numerical simulations, which validates the correctness of the model design and demonstrates its practical applicability. This implementation also establishes a reliable framework for real-time chaotic signal generation and lays a solid foundation for engineering applications in secure communication and embedded systems.

Based on the proposed DNMHNN model, a novel image encryption scheme has been designed by exploiting the chaotic sequences generated from the system. The encryption algorithm integrates permutation, XOR substitution, and bidirectional ciphertext-feedback diffusion operations to achieve effective confusion and diffusion. Extensive security analyses, including histogram distribution, correlation coefficients, information entropy, NPCR, UACI, key sensitivity, and robustness tests, indicate that the encrypted images exhibit near-uniform histograms, low pixel correlations, entropy values close to the theoretical maximum of 8, strong plaintext and key sensitivity, and strong resistance against noise, occlusion, tampering, and differential attacks. The additional quantitative results based on MSE, PSNR, SSIM, and recovered-image correlation characterize the attack-dependent recovery performance, showing partial structural recovery under limited ciphertext damage but substantial degradation under severe noise or tampering.

Furthermore, comparisons with recent image-encryption schemes show that the proposed algorithm achieves competitive information entropy, NPCR, UACI, and PSNR values. Under 25% ciphertext loss, its reconstruction PSNR is comparable to those of recently reported schemes, while the additionally reported SSIM and recovered-image correlation provide complementary evidence of structural recovery. These results indicate that the proposed scheme achieves a favorable balance among encryption security, attack robustness, and structural simplicity.

In conclusion, the proposed DNMHNN model not only provides a compact yet highly complex dynamical system with strong chaotic properties, but also demonstrates excellent performance in FPGA implementation and image encryption applications. The results suggest that the combination of memristive neural networks and chaotic dynamics offers a promising approach for developing efficient, secure, and hardware-friendly encryption systems, especially for lightweight and real-time information security applications.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the National Natural Science Foundation of China (No. 62273141) and Hainan Provincial Natural Science Foundation of China (No. 626MS0245).

Author Contributions: The authors confirm their contribution to the paper as follows: Conceptualization: Yanyu Zhu and Jie Jin; Methodology: Jie Jin; Software: Yanyu Zhu, Lv Zhao and Fei Yu; Validation: Yanyu Zhu, Jie Jin, Lv Zhao and Fei Yu; Formal analysis: Yanyu Zhu; Investigation: Yanyu Zhu, Jie Jin, Lv Zhao and Fei Yu; Resources: Jie Jin; Data

curation: Yanyu Zhu, Jie Jin, Lv Zhao and Fei Yu; Writing—original draft preparation: Yanyu Zhu; Writing—review and editing: Yanyu Zhu, Jie Jin, Lv Zhao and Fei Yu; Visualization: Yanyu Zhu and Jie Jin; Supervision: Jie Jin and Fei Yu; Project administration: Jie Jin. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request for academic verification purposes.

Ethics Approval: Not applicable. The data collection process was non-invasive and involved no medical or clinical procedures.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. He C, Jiang M, Shan K, Yang SH, Li Z, Wang S, et al. A hardware-adaptive learning algorithm for superlinear-capacity associative memory on memristor crossbars. *Nat Commun*. 2026;17:3096. doi:10.1038/s41467-026-69958-0.
2. Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci U S A*. 1982;79(8):2554–8. doi:10.1073/pnas.79.8.2554.
3. Hopfield JJ. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc Natl Acad Sci U S A*. 1984;81(10):3088–92. doi:10.1073/pnas.81.10.3088.
4. Chua LO. Memristor-the missing circuit element. *IEEE Trans Circuit Theory*. 1971;18(5):507–19. doi:10.1109/tct.1971.1083337.
5. Chua LO, Kang SM. Memristive devices and systems. *Proc IEEE*. 1976;64(2):209–23. doi:10.1109/proc.1976.10092.
6. Xia Q, Yang JJ. Memristive crossbar arrays for brain-inspired computing. *Nat Mater*. 2019;18(4):309–23. doi:10.1038/s41563-019-0291-x.
7. Li Z, Jin J, Zhang D, Chen C. Design and realization of pulse-controlled multi-memristor Hopfield neural networks and their applications in information encryption. *Multimed Syst*. 2025;31(6):427. doi:10.1007/s00530-025-02012-z.
8. Ielmini D, Wong HSP. In-memory computing with resistive switching devices. *Nat Electron*. 2018;1(6):333–43. doi:10.1038/s41928-018-0092-2.
9. Hu Z, Zhao Z. Memristive Hopfield neural network with hidden multiple attractors and its application in color image encryption. *Mathematics*. 2025;13(24):3972. doi:10.3390/math13243972.
10. Sağbaş M, Minaei S. Dynamic memdiode-driven Hopfield neural network for 3D CT encryption in IoMT. *Chaos Solitons Fractals*. 2026;208:118253. doi:10.1016/j.chaos.2026.118253.
11. Zhang F, Ge J, Jiang C, Bao H, Fan J, Sun K. Multi-scroll attractor generation in a discrete memristive Hopfield neural network: FPGA implementation and industrial image encryption. *Phys A Stat Mech Appl*. 2026;691(10):131501. doi:10.1016/j.physa.2026.131501.
12. Li R, Dong E, Tong J, Wang Z. A novel multiscroll memristive Hopfield neural network. *Int J Bifurc Chaos*. 2022;32(9):2250130. doi:10.1142/s0218127422501309.
13. Leng X, Wang X, Zeng Z. Memristive Hopfield neural network with multiple controllable nonlinear offset behaviors and its medical encryption application. *Chaos Solitons Fractals*. 2024;183:114944. doi:10.1016/j.chaos.2024.114944.
14. Deng Q, Wang C, Lin H. Memristive Hopfield neural network dynamics with heterogeneous activation functions and its application. *Chaos Solitons Fractals*. 2024;178:114387. doi:10.1016/j.chaos.2023.114387.
15. Lin H, Wang C, Yu F, Hong Q, Xu C, Sun Y. A triple-memristor Hopfield neural network with space multistructure attractors and space initial-offset behaviors. *IEEE Trans Comput-Aided Des Integr Circuit Syst*. 2023;42(12):4948–58. doi:10.1109/tcad.2023.3287760.
16. Liang S, Jia L, Yinhu Q. Initial offset boosting dynamics in a memristive hopfield neural network and its application in image encryption. *Chin J Comput Phys*. 2023;40(1):106–16. (In Chinese). doi:10.1007/s11571-026-10449-9.
17. Li F, Qin W, Xi M, Bai L, Bao B. Plane coexistence behaviors for Hopfield neural network with two-memristor-interconnected neurons. *Neural Netw*. 2025;183:107049. doi:10.1016/j.neunet.2024.107049.

18. Li Z, Jin J, Zhang D, Chen C. An attractor-controllable memristive Hopfield neural network and its application on voice encryption. *Integration*. 2025;105:102500. doi:10.1016/j.vlsi.2025.102500.
19. Deng X, Ding S, Lin H, Jiang L, Sun H, Jin J. Privacy-preserving online medical image exchange via hyperchaotic memristive neural networks and DNA encoding. *Neurocomputing*. 2025;653:131132. doi:10.1016/j.neucom.2025.131132.
20. Gao H, Xu H. Dynamics research of the Hopfield neural network based on hyperbolic tangent memristor with absolute value. *Micromachines*. 2025;16(2):228. doi:10.3390/mi16020228.
21. Wang H, Liu K, Liu Y, Wang J, Sun M, Tang K, et al. FPGA-implemented memristor-based transient chaotic neural network for AES edge encryption. In: *Proceedings of the 2023 International Conference on Electronics, Computers and Communication Technology*; 2023 Nov 17–19; Guilin China. p. 241–4.
22. Wang C, Luo D, Deng Q, Yang G. Dynamics analysis and FPGA implementation of discrete memristive cellular neural network with heterogeneous activation functions. *Chaos Solitons Fractals*. 2024;187:115471. doi:10.1016/j.chaos.2024.115471.
23. Mokbel AAM, Yu F, Gracia YM, Tan B, Lin H, Iu HH. Exploring dynamics and hardware implementation of an enhanced 5D hyperchaotic memristive system inspired by sprott-C system. *Complex Syst Model Simul*. 2025;5(1):34–45. doi:10.23919/csms.2024.0024.
24. Benkouider K, Mahdal M, Vaidyanathan S, Tlelo-Cuautle E, Martínez BO, Kaçar S, et al. FPGA implementation and voice encryption application of a new hyperchaotic system with high complexity. *Sci Rep*. 2026;16(1):4485. doi:10.1038/s41598-025-34605-z.
25. Yao W, Gao K, Zhang Z, Cui L, Zhang J. An image encryption algorithm based on a 3D chaotic Hopfield neural network and random row-column permutation. *Front Phys*. 2023;11:1162887. doi:10.3389/fphy.2023.1162887.
26. Liu G, Xu G, Wang C, Xu G. Hyper-chaos and CNN-based image encryption scheme for wireless communication transmission. *Comput Mater Contin*. 2025;84(3):4851–68. doi:10.32604/cmc.2025.066331.
27. Amdouni R, Madani M, Ali Hajjaji M, Bourennane EB, Atri M. Secure medical image transmission using chaotic encryption and blockchain-based integrity verification. *Comput Mater Contin*. 2025;84(3):5527–53. doi:10.32604/cmc.2025.065356.
28. Bhatti DS, Malik AW, Choi H, Kim KI. Sine-polynomial chaotic map (SPCM): a decent cryptographic solution for image encryption in wireless sensor networks. *Comput Mater Contin*. 2025;85(1):2157–77. doi:10.32604/cmc.2025.068360.
29. Sánchez BB, Rodríguez-Sela F, Alcarria R, Robles T. Hardware-enabled key generation in Industry 4.0 cryptosystems through analog hyperchaotic signals. *Comput Mater Contin*. 2025;83(2):1821–53. doi:10.32604/cmc.2025.059012.
30. Ulutas H. A novel memristor-based hyperchaotic hybrid encryption system with DNA for image encryption on the Jetson TX2. *Sci Rep*. 2025;15(1):35745. doi:10.1038/s41598-025-21604-3.
31. Zhang X, Chai Y, Xiang S, Li S. Research on image encryption with multi-level keys based on a six-dimensional memristive chaotic system. *Entropy*. 2025;27(11):1152. doi:10.3390/e27111152.
32. Liu S, Ding Q. A new color image encryption algorithm based on the memristor hyperchaos system and Rubik's cube theory. *Eur Phys J Plus*. 2024;139(9):820. doi:10.1140/epjp/s13360-024-05599-w.
33. Kumar BS, Revathi R. A comprehensive review on image encryption techniques using memristor based chaotic system for multimedia application. *IETE J Res*. 2024;70(11):8160–83. doi:10.1080/03772063.2024.2373899.
34. He S, Yu F, Guo R, Zhang F, Yao W, Cai S, et al. Burst-firing and extreme multistability in a dual-neuron fractional-order memristive HNN with hardware implementation. *Chaos Interdiscip J Nonlinear Sci*. 2025;35(11):113111. doi:10.1063/5.0294439.
35. Guo A, Fan C. A novel memristive Hopfield neural network with grid attractor and its application in image encryption. *Phys A Stat Mech Appl*. 2025;679:131005. doi:10.1016/j.physa.2025.131005.
36. Rahman ZSA, Jasim BH, Al-Yasir YIA, Abd-Alhameed RA. High-security image encryption based on a novel simple fractional-order memristive chaotic system with a single unstable equilibrium point. *Electronics*. 2021;10(24):3130. doi:10.3390/electronics10243130.

37. Manoj N, Sriraman R, Gurusamy R, Shang Y. Further results on global stability of Clifford-valued neural networks subject to time-varying delays. *Neurocomputing*. 2025;651:130886. doi:10.1016/j.neucom.2025.130886.
38. Qiu D, Zhang T, Liu J, Liu S, He P. An innovative image encryption algorithm based on the DNAS_box and hyperchaos. *Entropy*. 2025;27(3):239. doi:10.3390/e27030239.
39. Afify YM, Sharkawy NH, Gad W, Badr N. A new dynamic DNA-coding model for gray-scale image encryption. *Complex Intell Syst*. 2024;10(1):745–61. doi:10.1007/s40747-023-01187-0.
40. Chang H, Wang E, Liu J. Research on image encryption based on fractional seed chaos generator and fractal theory. *Fractal Fract*. 2023;7(3):221. doi:10.3390/fractalfract7030221.
41. Kumar S, Sharma D. A chaotic based image encryption scheme using elliptic curve cryptography and genetic algorithm. *Artif Intell Rev*. 2024;57(4):87. doi:10.1007/s10462-024-10719-0.
42. Zhang P, Jie J, Liu Z, Dong K. Chaotic image encryption system as a proactive scheme for image transmission in FSO high-altitude platform. *Photonics*. 2025;12(7):635. doi:10.3390/photonics12070635.