IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 July 2022, accepted 3 September 2022, date of publication 6 September 2022, date of current version 15 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3204821

RESEARCH ARTICLE

Analysis and Optimization of Network Properties for Bionic Topology Hopfield Neural Network Using Gaussian-Distributed Small-World Rewiring Method

JUN SUN, (Member, IEEE), SARATHA SATHASIVAM, AND MAJID KHAN BIN MAJAHAR ALI
School of Mathematical Sciences, Universiti Sains Malaysia (USM), Gelugor, Penang 11800, Malaysia

Corresponding author: Saratha Sathasivam (saratha@usm.my)

This work was supported by the Research University Grant (RUI) 1001/PMATHS/8011131 by Universiti Sains Malaysia.

ABSTRACT The fully connected topology, which coordinates the connection of each neuron with all other neurons, remains the most commonly used structure in Hopfield-type neural networks. However, fully connected neurons may form a highly complex network, resulting in a high training cost and making the network biologically unrealistic. Biologists have observed a small-world topology with sparse connections in the actual brain cortex. The bionic small-world neural network structure has inspired various application scenarios. However, in previous studies, the long-range wirings in the small-world network have been found to cause network instability. In this study, we investigate the influence of neural network training on the small-world topology. The role of the path length and clustering coefficient of neurons is expounded in the neural network training process. We employ Watt and Strogatz's small-world model as the topology for the Hopfield neural network and conduct computer simulations. We observe that the random existence of neuron connections may cause unstable network energies and generate oscillations during the training process. A new method is proposed to mitigate the instability of small-world networks. The proposed method starts with a neuron as the pattern centroid along the radial, which arranges its wirings in compliance with the Gaussian distribution. The new method is tested on the MNIST handwritten digit dataset. The simulation confirms that the new small-world series has higher stability in terms of the learning accuracy and a higher convergence speed compared with Watt and Strogatz's small-world model.

INDEX TERMS Small world network, artificial neural networks, hopfield neural networks, k satisfiability, network theory (graphs), logic programming.

I. INTRODUCTION

John Hopfield introduced a neurobiology-based computational model in his research on content addressable memory in 1982 [1], and the Hopfield neural network (HNN) became one of the best bionic computational models of that time. The HNN showed various advantageous properties such as object recognition capabilities, categorization, and error correction. Despite the recent dominance of neural networks (e.g., convolutional neural networks) using different learning methods (e.g., gradient descent), classifiers (e.g., softmax), and hardware accelerations (e.g., CUDA), the Hopfield-type neural network is still one of the most effective computational models that can be trained similar to the real biological brain. The HNN is a spin dynamics system that coordinates the connection of each neuron with all the other neurons (without self-loops, Discrete HNN). Each neuron pair is connected by a synaptic weight, and each neuron performs a weighted summing on the states of other neurons. The neuron state is activated by the presetting threshold of the signum function

The associate editor coordinating the review of this manuscript and approving it for publication was Rodrigo S. Couto.

and the influences of the other neurons. The weight matrix of Hopfield neural networks is symmetric and has zeroes as diagonal elements. The Lyapunov function is employed to define its minimum network energy. Driven by the minimized energy, neurons flip the state towards the local minimum solution.

Decades after its introduction, the HNN had been used and enhanced in various applications, such as object recognition [2], image restoration [3], combination optimization [4], and very-large-scale integrated arrays [5]. By mining the logic relations of a neural network, Abdullah proposed a new learning method for the DHNN in 1992 [6]. In 2011, Sathasivam and Abdullah expanded this method and formally named it the Wan Abdullah method [7]. The Wan Abdullah method first processes neurons in bipolar states (DHNN), and then, the Boolean relation between the pair of neurons is written as a clause in a conjunctive normal form (CNF). When all clauses are ''true,'' the CNF is satisfied. The cost function is established, and the inconsistencies in the clauses are mostly minimized. In 2016, Mansor *et al.* extended the Wan Abdullah method to 3-satisfiability (3-SAT) to optimize the pattern satisfiability problem [8]. In 2017, Kasihmuddin *et al.* proposed a hybrid method that employed the HNN and genetic algorithm to solve k-SAT problems [9]. In 2020, Sathasivam *et al.* [10] proposed a method that integrated random k-SAT with the HNN. Despite the HNN being enhanced with various reinforcements from the Wan Abdullah learning method in terms of computing efficiencies, learning accuracies, etc., fully connecting a large number of neurons remains a bottleneck for training large neural networks. In practice, if a 400 × 400 pixel image is used, a neural network would need 160,000 neurons to generate a weight matrix of a size of 1/2 × (160,000 – 1) ×160,000. This leads to an insurmountable problem of allocating considerable memory space for the weight matrix [11]. During the training process, enumeration of such a large weight matrix would also reduce the learning efficiency of the algorithm. Moreover, the fully connected topology is biologically unrealistic [12]. As shown in biological applications [13], the structure of the real brain is sparser than that of the fully connected topology, each neuron is connected with only a few other neurons, exhibiting more complexity than the fully connected and the fully random topologies [14]. In a recent biological study [15], the small-world topology was observed in human anatomical and functional brain networks. The small-world network is named based on the analogy with the small-world phenomenon, and it is a high-centrality network discovered by Duncan J. Watts and Steven H. Strogatz in 1998 [16]. The Watts and Strogatz (WS) small-world model starts with a regular ring lattice of N nodes, where each node connects with K nodes from a nearby region. Subsequently, based on parameter P ($0 \leq P \leq 1$), each node is rewired at one side (K/2 connections per side) to other randomly chosen nodes in the network, forming a high centrality network. After biological observations, the bionic sparse small-world network has attracted significant attention across many fields. Scientists have evaluated the influence of the topology on the network memory function and measured varieties of topologies in terms of storage performance and pattern retrieval [17], [18]. The small-world network with several shortcuts has been shown to have the same efficiency as a random network. Recently, researchers further confirmed the biological reality of small-world networks. Chen *et al.* [19] reported that the cerebellar functional connectome of an actual human is small-world organized. Rosen *et al.* [20] estimated the absolute number of axons linking cortical areas from a whole-cortex diffusion MRI connectome and observed that real human cortical areas are small-world connected. Pircher *et al.* [21] further compared the small-world network between artificial and biological-based neural networks and observed remarkable parallels between these two neural networks. Scientists have started exploring the specifics of the characteristics of small-world networks. Arvin *et al.* [22] explored the role of short and long-range connections in small-world networks. Their research revealed that short-range connections dominate the dynamics of the system, e.g., affect the volatility and stability of the network, and long-range connections drive the system state. Rüdiger *et al.* [23] also reported that the long-range connections of small-world networks may make the network unstable, supporting frequent supercritical mutations. Ercsey-Ravasz *et al.* [24], [25] uncover a rule that the probability that two neurons are connected declines exponentially as a function of the distance between them. This important principle is termed ''the exponential distance rule''. Takagi [26] studied energy constraints for modeling human brain connections. His results shown that the energy constraints play a crucial role in regulating brain structures. These studies have implied the random rewiring mechanism of the WS small world model may form a neuron connection distribution that does not conform to the bio-growth cost rule and the exponential distance rule. In addition, the current WS small-world model is yet to consider the specifics of neural network training, such as the consistency of network energy and the random neuron connections that may cause unstable network energy. Therefore, clarifying the influence of neural network training on the small-world topology and improving its stability are urgent.

The contributions of this study are as follows: 1) The impact of the small-world topology on the HNN training is investigated. The role of the path length and clustering coefficient of neurons in the neural network training process is elaborated. 2) The instability shortcoming of the random rewire mechanism in the WS small-world model is discussed. The random existence of neuron connections that may cause unstable network energies and generate oscillations during the training process is highlighted. 3) The Gaussian-distributed small-world wiring method is proposed to improve the stability of the small-world HNN. The novelty of the new rewiring method is that it organizes neuron connections in compliance with the Gaussian distribution. This reduces random connections from the distant area and makes the short-range connections dominate the main part of

network energy, improving the stability of the small-world network model.

The remainder of this paper is organized as follows: In Section II, we provide a brief overview of the DHNN. We introduce the fully connected topology, Hebbian learning method, and 2-Satisfiability (2-SAT) Wan Abdullah learning method. Then, we present the theory of the small-world network model and the network properties. In section III, we describe the small-world DHNN. We start by elaborating on the impact of the small-world topology on discrete neural network training. Then, we discuss the implementation of our computer simulation. Finally, we discuss our evaluation of small-world network characteristics and learning accuracies. In section IV, we explain the proposed Gaussian-distributed small-world wiring method. We introduce the instability problem of the small-world neural network and the Gaussian -distributed small-world wiring method. Next, we discuss the coincidence degree comparison test and logic energy validation test. Finally, we describe our digit recognition experiment. In section V, we provide the conclusions.

II. BACKGROUND

A. DHNN

The DHNN is a special HNN capable of processing binary data. The DHNN remains in compliance with the spin dynamic system; however, the state of each neuron is bipolar and is usually denoted by $\{0,1\}$ or $\{-1,1\}$. Each neuron is fully connected without the self-loop, and each neuron pair is connected by synaptic weight. Figure 1 illustrates a fully connected DHNN with n neurons. $X_1, X_2, \ldots, X_n$ represents the output of each neural node, and $T_1, T_2 \ldots, T_n$ represents the predefined threshold. Each neuron performs a weighted sum on the states of other neurons, denoted by $V_i = \sum_{j=1}^{n} w_{i,j} \cdot x_j$, where $i \neq j$, $i \in \{1,n\}$. The state of neural node i can be calculated by its activation function, which typically is a signum function. S_i represents the state of neural node i,

$$S_i = \begin{cases} 1 \; where \; V_i - T_i > 0 \\ -1 \; where \; V_i - T_i < 0. \end{cases}$$

Therefore, the state of neura l node i is determined by threshold T_i and the influences received from the states of other nodes. The weight matrix of the DHNN, which is symmetric and zero diagonal, has $w_{i,j} = w_{j,i}$, $w_{i,i} = w_{j,j} = 0$. The state updating rule is maintained by $S_i(t+1) = signum[V_i(t) - T_i]]$, where t represents the time of process for the neural node i. The Lyapunov energy function is $E(t) = -\frac{1}{2}\sum_j \sum_i w_{i,j} S_i(t) S_j(t) - \sum_i T_i S_i(t)$, where $i \neq j$. The energy function reduces with dynamics monotonically. Subsequently, motivated by the minimized energy, neural nodes flip state towards the local or global minimum solution.

In the training process, the weight between neurons i and j can be calculated by (1). In equation (1), s denotes the pattern number. This weight updation method is usually called the Hebbian method.

$$w_{i,j} = \sum_{s=1}^{m} \left(2V_i^s - 1\right)\left(2V_j^s - 1\right) \tag{1}$$

In the Wan Abdullah method, energy is built upon the satisfiability of the clause composed of neurons. α represents the Boolean relations between neurons i and j, and the CNF is written as (2). To identify inconsistencies between the clauses, (2) is negated by applying the De Morgan law and is written as (3). Subsequently, the cost function can be written as (4).

$$\begin{aligned} \alpha = {} & \left(S_i \vee S_j\right) \wedge \left(S_i \vee \neg S_j\right) \wedge \left(\neg S_i \vee S_j\right) \\ & \wedge \left(\neg S_i \vee \neg S_j\right) \wedge \left(T_i \vee T_j\right) \end{aligned} \tag{2}$$

$$\begin{aligned} \neg\alpha = {} & \left(\neg S_i \wedge \neg S_j\right) \vee \left(\neg S_i \wedge S_j\right) \vee \left(S_i \wedge \neg S_j\right) \\ & \vee \left(S_i \wedge S_j\right) \vee \left(\neg T_i \wedge \neg T_j\right) \end{aligned} \tag{3}$$

$$\begin{aligned} E_\alpha = {} & \frac{1}{2}(1 - S_i)\frac{1}{2}(1 - S_j) + \frac{1}{2}(1 - S_i)\frac{1}{2}(1 + S_j) \\ & + \frac{1}{2}(1 + S_i)\frac{1}{2}(1 - S_j) + \frac{1}{2}(1 + S_i)\frac{1}{2}(1 + S_j) \\ & + \frac{1}{2}(1 - T_i)\frac{1}{2}(1 - T_j) \end{aligned} \tag{4}$$

The synaptic weight can be computed in Table 1. In equation (2), when all the clauses (composed of two literals) are satisfied, α is called 2-satisfied. When extended to the entire network, each weight of the network can be computed by iterating all the pairs of neurons according to equations (3), (4), and Table 1.

In this study, we implemented both the Hebbian method and the 2-SAT Wan Abdullah learning method to explore the impact of small-world topology on neural network training in our computer simulations. We compare these two methods under the small-world neural network regarding learning accuracy and convergence speed. However, higher 3-SAT and Max-kSAT are not integrated because the small-world rewiring mechanism does not guarantee connections between any three neurons.

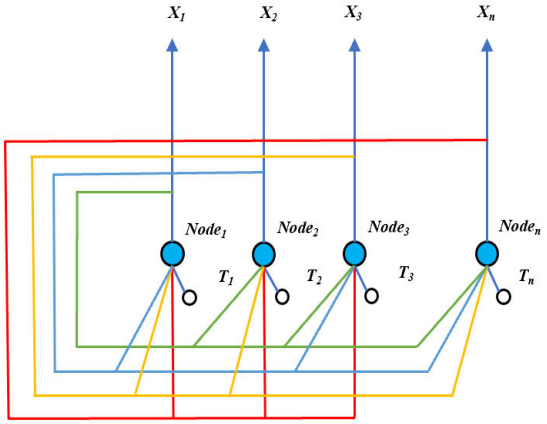

FIGURE 1. Fully connected Hopfield neural network.

TABLE 1. Wan Abdullah method for 2 satisfiability.

	$\frac{1}{2}(1-S_i)\frac{1}{2}(1-S_j)$	$\frac{1}{2}(1-S_i)\frac{1}{2}(1+S_j)$	$\frac{1}{2}(1+S_i)\frac{1}{2}(1-S_j)$	$\frac{1}{2}(1+S_i)\frac{1}{2}(1+S_j)$	$\frac{1}{2}(1-T_i)\frac{1}{2}(1-T_j)$
	$=\frac{1}{4}(1-S_j-S_i+S_iS_j)$	$=\frac{1}{4}(1+S_j-S_i-S_iS_j)$	$=\frac{1}{4}(1-S_j+S_i-S_iS_j)$	$=\frac{1}{4}(1+S_j+S_i+S_iS_j)$	$=\frac{1}{4}(1-T_j-T_i+T_iT_j)$
	$-\frac{1}{2}W_{i,j}S_iS_j-W_jS_j-W_iS_i-\frac{1}{2}W_{T_iT_j}T_iT_j-W_{T_i}T_i-W_{T_j}T_j$				
w_i	$\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	0
w_j	$\frac{1}{4}$	$-\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	0
$w_{i,j}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	0
W_{T_i}	0	0	0	0	$\frac{1}{4}$
W_{T_j}	0	0	0	0	$\frac{1}{4}$
W_{T_i,T_j}	0	0	0	0	$-\frac{1}{2}$

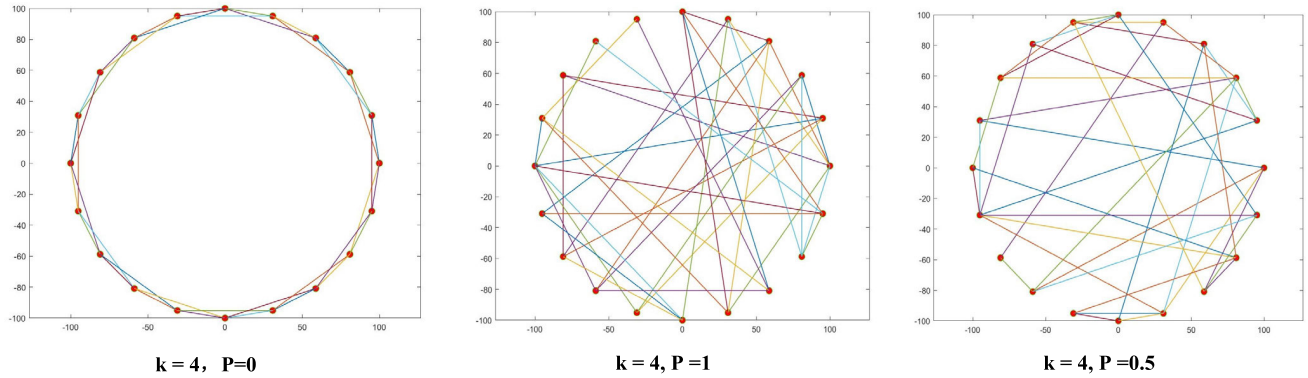

FIGURE 2. Small world network and topologies formed by different rewired probabilities.

B. SMALL-WORLD NETWORK MODEL

The small-world network, named after the analogy with the small-world phenomenon, is a high-centrality network discovered by Watts and Strogatz in 1998 [16]. The small-world network is a network model with a high clustering coefficient and low average path length. The nodes in the small-world network are mostly not neighbors of each other. However, a node pair can access another pair by just a few steps (neighbors). Therefore, the small-world network is defined as a network [16] where $L \infty log\, N$, where L denotes the average path length of the network, and N denotes the number of nodes in the network. $L \infty log\, N$ means L increases proportionally to $\log N$. Meanwhile, given a node i that might connect with k nodes in the network and denoting e as the actual connections of node i, the maximum connection quantity can be written as $\mathrm{Max}(e) = 1/2\, k\, (k-1)$, representing the node i being fully connected with the other nodes. The clustering coefficient of node i is defined as $\frac{e}{\frac{1}{2}k(k-1)}$. The entire network clustering coefficient can be computed by averaging the total clustering coefficient of all nodes.

In 1998, WS proposed a method to form the small-world network [16]. The method proposed two stages to form the small-world network: initializing and rewiring. With n nodes in the network initializing stage, each node is connected with k nodes from the near region to begin forming a high-centralization ring structure. Each node N_i is divided into two sides (left and right) in the rewiring stage, with k/2 connections per side. P is the rewiring probability. Connections are taken from the right side and rewired to a randomly selected node by P, and no self-loop is stipulated. Adjusting P can yield a small-world network between the regular network ($P = 0$) and the random network ($P = 1$). Figure 2 shows three topologies formed by different rewiring probabilities from a 20 node network. The case $k = 4$, $P = 0$, forms a regular network, while $k = 4$, $P = 1$ generates a random network. The small-world network may be formed when P is in the interval [0,1]. This illustrates a small-world network formed under $k = 4$, $P = 0.5$. By adjusting the P, the small-world network can be formed between the regular and random networks.

The WS small-world model elaborates on the small-world phenomenon in the real world, such as the neural network of the worm Caenorhabditis elegans, the power grid, and the collaboration graph of film actors. In complex network theory, the small-world network belongs to a type of random network. However, the small-world network has unique bionic advantages that differ from other complex networks. In this study, we employed the WS small-world model as the topology for HNN. As the small-world network is the only complex network observed in real biological situations, we did not compare it with other complex networks regarding performance. However, the clustering coefficient and average length, important network properties in complex network theory, were considered.

C. SMALL WORLD NETWORK PROPERTIES

1) CLUSTERING COEFFICIENT

The clustering coefficient is the probability that specific neurons in a network are likely to cluster together [16]. In the local network, for node i, which has k neighboring nodes connected, the local cluster coefficient, C_i, of neuron i can be written as (5). Numerator e denotes the number of wired connections between neighbors, and denominator $\frac{1}{2}k\,(k-1)$ indicates the maximum possible (fully connected) connection number.

$$C_i = \frac{e}{\frac{1}{2}k\,(k-1)} \quad k \neq 1 \tag{5}$$

Extended to the network with n nodes. The summing of clustering coefficient C_i can be averaged to rewrite C_n as

$$C_n = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{6}$$

2) AVERAGE PATH LENGTH

The average path length is defined as the average distance between each node pair in the network [16]. Suppose $D_{i,j}$ denotes the distance of the shortest path between neural node i and j in the network, then L represents the average path length that can be computed by (7), where $\binom{n}{2}$ denotes the number of all possible pairs of neural nodes in the n nodes network.

$$L = \binom{n}{2}^{-1} \sum_{i \neq j} D_{i,j} \tag{7}$$

III. SMALL-WORLD DHNN

A. IMPACTION OF SMALL-WORLD TOPOLOGY ON DHNN TRAINING

We observed two major changes from the WS small-world model that significantly impact DHNN training. The first is the changes in the network characteristics. The sparse neuron connections increase the average path length and reduce the clustering coefficient of the network. The degradation of the network characteristics may eventually cause insufficient training neurons, decrease the training efficiency, and even generate oscillations. The second is the changes in network energy. The small-world network energy may be divided into two parts: the regular lattice part is formed by $K/2$ nodes from the near region, and randomly chosen nodes form the random rewire part of the network. The network neuron flips the state driven by these two energy parts. The regular lattice part is the same as that formed by the same region of the network. But the random rewired part is unstable. It may cause the neuron to flip to the wrong state. In this section, we compare the training process of the fully connected structure with that of the small-world structure, and then elaborate on the impact on network training in terms of network characteristics and energy.

Figure 3 illustrates two neural network training processes. The figure on the top is the fully connected structured neural network, and the bottom is the small-world structured neural network. The yellow node represents the neuron in training. The fully connected structure organizes the yellow node to connect with all other nodes. The yellow node connects to only a few blue nodes in the small-world structure. The nodes in the green frame represent the regular lattice nodes, and the rest of the nodes represent the random rewired nodes of the small-world network. During the training process, the yellow node traverses from left-top to right-bottom of the network. Under the fully connected structure, each weight between the yellow and blue nodes needs to be updated. By comparison, the yellow node connects fewer blue nodes under the small-world structure, thus requiring fewer weights to be updated. The Hebbian method and the Wan Abdullah method remain applicable as the learning method for the small-world neural network, and the signum function decides the neuron state. We noted no significant difference in the training mechanism between the fully connected and the small-world neural network. The changes in network characteristics and energy are two essential factors that impact neural network training.

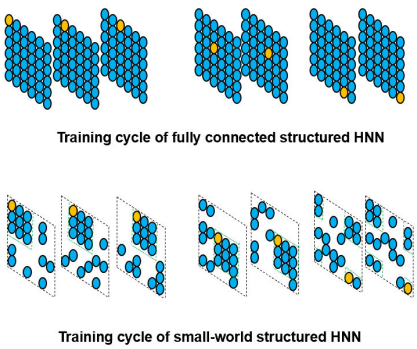

FIGURE 3. Small-world topology vs full connected topology on Hopfield neural network training process.

The impact of the network characteristics is mainly reflected in the energy converge efficiency of neural network training. The network characteristics are described by two factors: the average path length and the clustering coefficient of the network. In Figure 4, we compare the average path length of the fully connected topology with that of the small-world topology HNN. In the fully connected structure, each

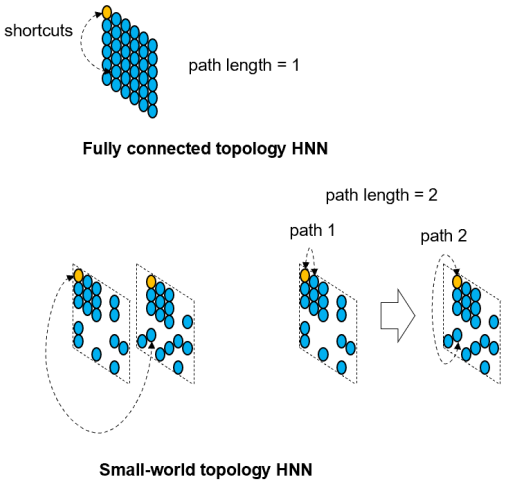

FIGURE 4. Topology impaction on Hopfield neural network training: average path length.

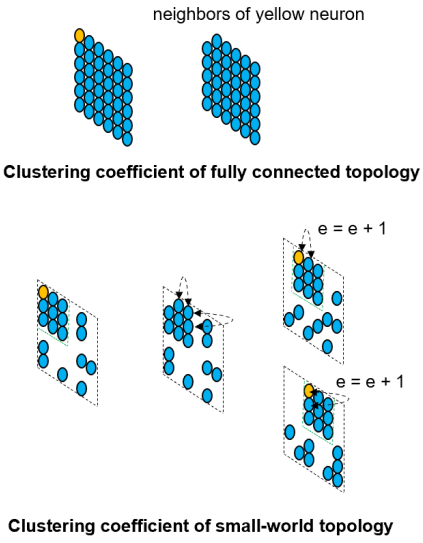

FIGURE 5. Topology impaction on Hopfield neural network training: clustering coefficient.

node pair has a shortcut, which means any node can be accessed in just one step. Hence, the path length of the fully connected structure equals one. After one round of training, each neuron considers the effects of all the other neurons in the network. However, most neuron pairs may not have a shortcut in the small-world structure. The regular lattice ring assures that each neuron is accessible, and the random rewire decreases the average path length. Figure 4 illustrates an example of a neuron pair whose path length is two. During network training, each neuron only considers a few connected neurons in the network. Therefore, the neuron in the small-world structure requires more training cycles to accumulate the full effects of all the neurons. The average path length usually is proportional to the number of training iterations. The longer the average path length, the more training iterations are required. In addition, the clustering coefficient impacts the network training from another angle.

In Figure 5, the clustering coefficient of the fully connected structure equals one because all neighbors are connected, where $e = \frac{1}{2}k(k-1)$. The clustering coefficient of the small-world structure is smaller than that of the fully connected structure; $C_i = \frac{e}{\frac{1}{2}k(k-1)}$ is the clustering coefficient of the yellow neuron, k denotes the number of neighbors (means the neurons that are connected with the yellow neuron), and e represents the actual connection number between its neighbors. The clustering coefficient describes how the neighbors of a neuron are clustered together. The smaller the clustering coefficient, the fewer neighbors are connected, and the lower the energy transfer efficacy. Therefore, more training is required.

The composition of the small-world neural network energy differs from the fully connected network. In Figure 6, a digit image is selected as the data pattern to display the network energy. The small-world topology is formed by the WS method. We evaluated and compared the energy distribution for the small-world and the fully connected topologies. Under the fully connected topology, the energy distribution appears uniform. A neuron may absorb the energy from each neuron in the network. The signum function decides the neuron states by summing the energy. By contrast, the energy distribution of the small-world network is highly centralized. The neurons compose the ring lattice occupying the central part of the energy. The energy of the rewiring part is distributed randomly and scattered in the network. During the training process, the neuron absorbs energy from both the lattice ring and the randomly wired neurons. Unconnected neurons are indirectly affected by the connected neurons. In addition, the energy of the random wire part is composed of randomly chosen neurons; hence, it is not guaranteed to be consistent with the remaining parts of the fully connected network.

B. IMPLEMENTATION OF SMALL WORLD DISCRETE HOPFIELD NEURAL NETWORK

1) THE SMALL-WORLD DHNN ALGORITHM

To better clarify the small-world DHNN, we constructed the WS small-world model over the DHNN. We divided the small-world HNN algorithm into two stages. In the initializing stage, a user is required to input the parameters k and rewire probability P. Then, the weight matrix for the HNN is initialized by the connection information of the topology. In the learning stage, both the Hebbian and 2SAT Wan Abdullah learning methods are integrated, and a variable is defined to switch between the learning methods. Either the Hebbian or 2SAT Wan Abdullah method may update the weight matrix. The signum function is applied to update the neuron state till all neurons remain at a stable state, and then the stop condition is met. Then, the output pattern can be produced. Based on the assumption that the input is a pattern in binary format, the algorithm of the small-world HNN is as follows.

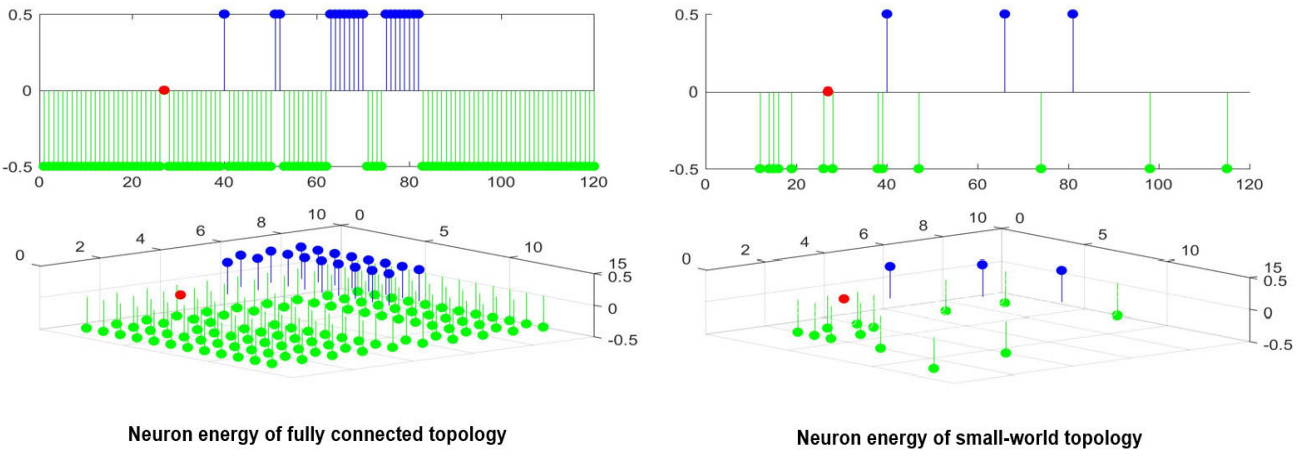

FIGURE 6. Energy comparison: small-world neural network vs fully connected neural network.

TABLE 2. Parameters of small-world Hopfield neural network.

Parameters		
Parameter name	**Data type**	**Descriptions**
num_neurons	Integer	neuron numbers of the network
k_neurons	Integer	number of connected nearest neurons
p_rewire	Double	rewire probability
weights	Double	2d matrix to store weights
links	Integer	2d matrix to store links
patterns	Scalar	matrix to store data for train patterns
cur_patterns	Scalar	Matric for output pattern
num_patterns	Integer	Number of patterns
threshold	Double	Threshold T of neural work
iterations	Long	Iterations of training cycle or epoch

Initializing stage:

Step 1: Initialize the input pattern, and specify the size of the pattern n and the data of the neuron states.

Step 2: Initialize the small-world parameters, specify parameter k, rewire probability P, and allocate the weight matrix with $n \times k$ size.

Step 3: Initialize the small-world network, and connect each neural node with its k nearest neighbor nodes.

Step 4: Rewire each neural node, divide $k/2$ connections for each neural node as "left" and the remaining $k/2$ connections as "right." Hold the connections of the left side of the neural node, and rewire the remaining connections from the right side to other nodes selected randomly by probability P.

Learning stage:

Step 5: Start iterating neural nodes in the network, and read the neuron state from the pattern.

Step 6: If the Hebbian method is specified, compute the weight for neural node i, j by equation $w_{i,j} = \sum_{s=1}^{m} \left(2V_i^s - 1\right)\left(2V_j^s - 1\right)$. Sum the weight calculated for each pattern.

Step 7: If the 2-SAT Wan Abdullah method is specified, check the weight value from equation (5). The Wan Abdullah method for 2-SAT by the neuron state represents variables of the clause.

Step 8: Update the state for neural nodes by $S_i = \begin{cases} 1 \; where \; V_i - T_i > 0 \\ 0 \; where \; V_i - T_i < 0 \end{cases}$, where T_i is the presetting threshold, and V_i can be computed by $V_i = \sum_{j=1}^{n} w_{i,j} \cdot x_j$. x_j refers to the output of neural node j.

Step 9: Check the stop condition where all neurons of the network meet $S(t) = S(t-1)$, then stop the training iteration. Otherwise return to Step 5. Continue the training iterations till the stop condition is met.

Step 10: Obtain the weight matrix. Compute the state for all neurons of the network by signum function. Yield the pattern for output.

2) IMPLEMENTATION

The small-world-based HNN algorithm was implemented using C++ for Microsoft visual studio 2017 on a machine with an i5 CPU, 16 GB memory, and the Windows 10 operating system. In our embodiment, the neural network's size (number of neural nodes) is defined as an integer variable. The rewire probability P and the number of the nearest neuron k are parameterized to form the small-world network. A multi-dimensional array is assigned to store the weight matrix and the connection in the formation. The weight matrix is allocated on a continuous memory address to boost

the searching and iteration speed. In the initializing stage, the weight matrix is initialized according to the network size and the value of parameter k. Then, each node is connected with k neurons from the nearest region. While rewiring neurons, the random number generator (RNG) service is applied to generate the rewire probability P and the random address of neural nodes for selection. Each neuron rewires $k/2$ connections using the probability P. The input pattern is stored in a scalar matrix in the learning stage, and the pattern data is assigned to the neuron at the corresponding address. The Hebbian and Wan Abdullah methods are alternatives for updating the neuron weights. The state of the neuron is determined by the signum function $S_i = \begin{cases} 1 \text{ where } V_i - T_i > 0 \\ 0 \text{ where } V_i - T_i < 0 \end{cases}$, T_i denotes the neuron's threshold, and V_i is updated by summing up the neuron states. As an intermediate result, these states are saved in a separate matrix. The training process is maintained till the stop condition is met. In our embodiment, two criteria are considered to stop training. The first criterion is that all neurons remain stable and meet the condition S(t)=S(t-1). Meanwhile, a threshold (epoch) for limiting the maximum learning cycle is integrated to prevent the program from overflowing. In the retrieval stage, the output pattern can be computed by applying the signum function to the learned weight matrix.

C. MEASUREMENTS AND TRAINING EXPERIMENTAL RESULTS

To further explore the competency of the small-world network as the topology of HNN, we measured its performance in two aspects. First, we measured the network characteristics and compared them with the fully connected network. Second, to comprehensively measure the learning competence of the small-world neural network, we tested the learning accuracy under different combinations of k and P.

The first test was run on small-world structures of 100, 500, and 1000 nodes, mainly considering the impact of the rewiring on the network clustering coefficient and the average path length. By making k constant, P was increased gradually. The network clustering coefficient and average path length were then measured. In Figure 8, the horizontal coordinates represent the rewiring probability, and the vertical coordinates represent $L(P)/L(0)$ and $C(P)/C(0)$, which denote the ratios of the average path length and clustering coefficient when rewiring probability equals P and 0. From Figure 8, we observed that while the rewiring probability remains small, a slight increase leads to a substantial drop in the average path length; meanwhile, the impact on the clustering coefficient appears very small. Therefore, such network characteristics may enable the small-world network to reduce its average path length by only slightly rewiring and further shrink the gap with the fully connected network. This also suggests that the sparse small-world may achieve a performance comparable to the fully connected network. Its competency as a neural network topology is confirmed regarding network characteristics.

In the second test, we measure the learning accuracy of the small-world neural networks formed by different combinations of the parameter k and rewiring probability P. We defined learning accuracy as the similarity between the retrieved and original input patterns. We assumed that k equals the size of a network with n neurons and that the rewiring probability P equals 0. Then, k is gradually reduced to $k = 4$, and for each k, P is distributed from 0 to 1 at 0.05 intervals. Figure 9 shows the similarity distributions with different combinations of k and P. We observed that in most cases where k is large ($k > 20$), the similarity equals 1, which means that the retrieved pattern is entirely consistent with the original pattern. Points at which the similarity is less than 1 mostly appear in the area where k is small ($k < 16$). Figure 9 shows the trend in learning accuracy under different combinations of k and P for the small-world network characteristics. Starting with $P = 0$, when k tends toward n, it means the network is closer to the fully connected structure and tends toward obtaining the same learning results as the fully connected neural network. The fully connected neural network is obtained till $k = n$. Conversely, while k is gradually decreased by only slight rewiring, we may obtain a remarkably high learning accuracy (learning accuracy $= 1$). When k drops to a small value ($k < 20$), the learning accuracy also drops. This trend is also observed throughout our tests on the MNIST dataset.

However, we also realize that one combination of k and P may form different structures because the current random rewiring mechanism may wire a neuron to any neuron in the network. Therefore, the random drop in the learning accuracy may be attributed to the unguaranteed composition of the inconsistent energy compared with that of the fully connected structure.

IV. GAUSSIAN-DISTRIBUTED SMALL- WORLD WIRING METHOD

A. INSTABILITY

As mentioned in Section III, there are two energy parts to consider during the training process of the WS small-world neural network: 1) The regular lattice ring, which is entirely consistent with that of neurons at the corresponding position of the fully connected network. These neurons form a stable energy. 2) The random rewiring, which is of a highly uncertain state because these neurons are chosen randomly from the network. These rewired neurons form the unstable energy of the network. When the rewiring energy is increased to a certain proportion in the composition of the network energy, it magnifies the adverse effects in terms of convergence efficiency and learning accuracy and even generates oscillations during the training process. However, appropriate rewiring may substantially reduce the average path length for the network, and can promote convergence efficiency. In addition, when k increases, the granularity of the regular lattice ring increases, and the learning accuracy is also enhanced.

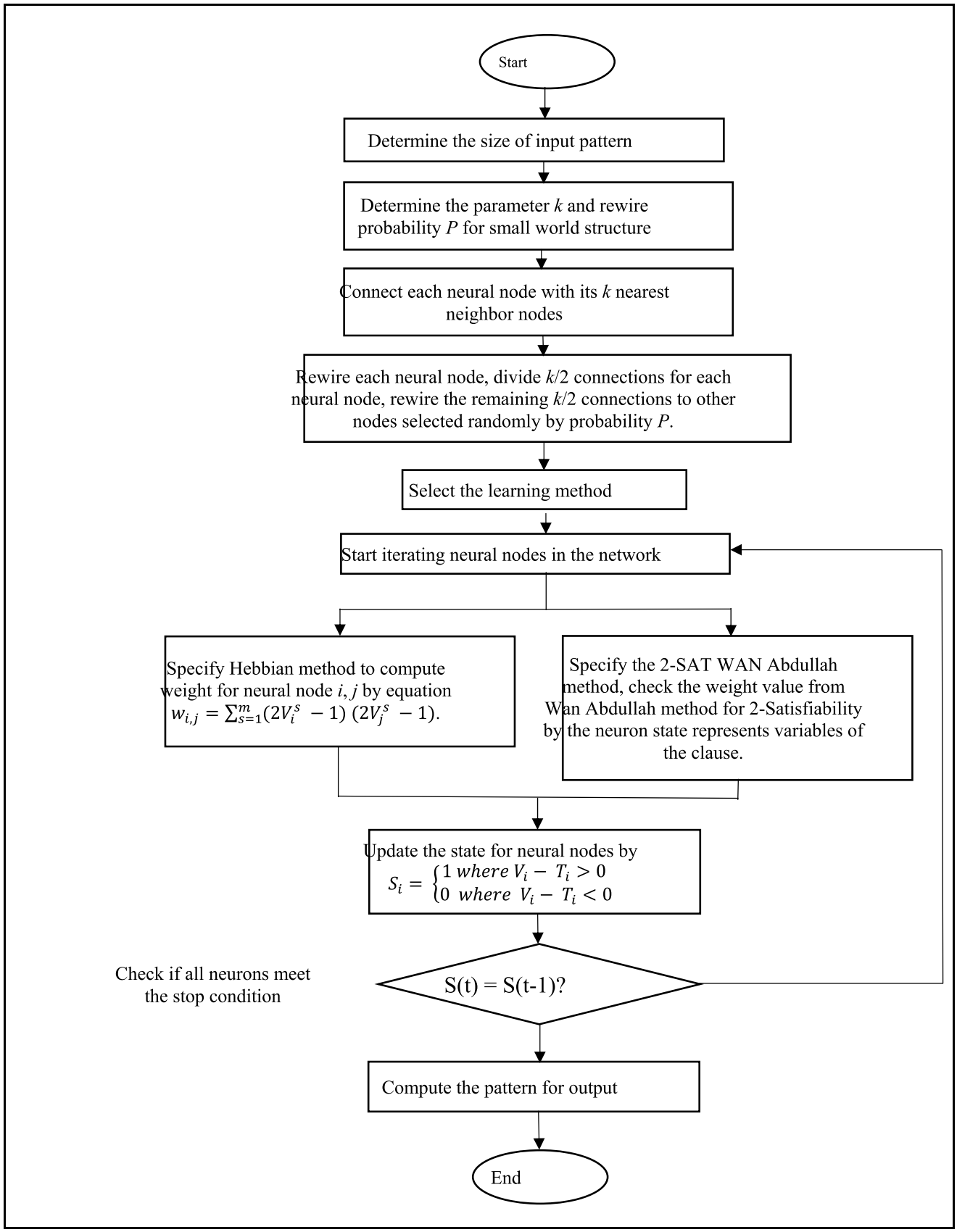

FIGURE 7. Flowchart of small-world Hopfield neural network.

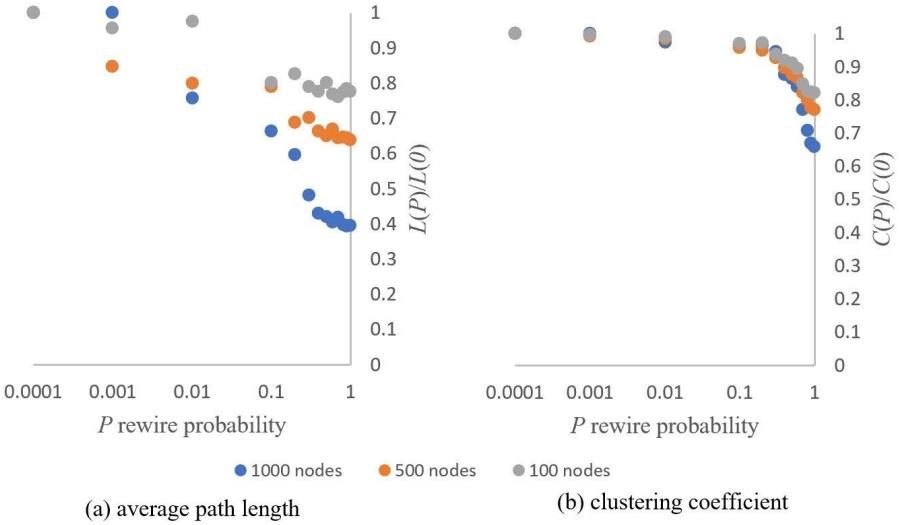

FIGURE 8. Measurement the network characteristics: average path length and clustering coefficient.

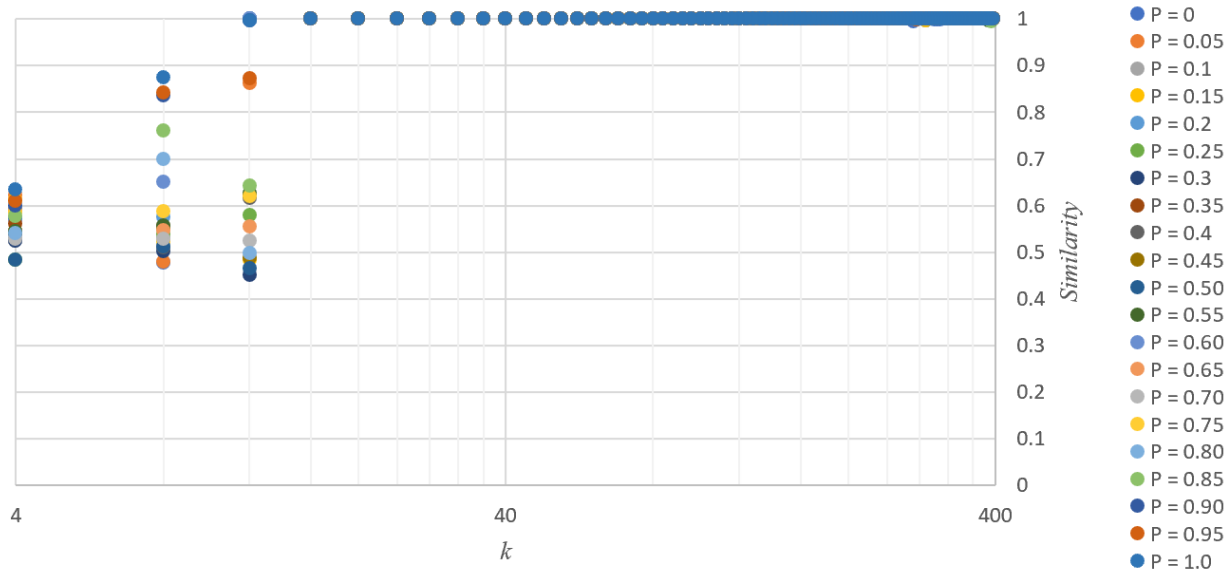

FIGURE 9. Measurement of learning results with small world combinations of k, P.

The current WS small-world model uses the random rewiring mechanism. However, as the topology of the neural network, the assurance of consistent energy is yet to be considered. We propose two improvements to the WS small-world model to promote stable network energy and learning accuracy. The first is to increase the proportion of the energy for the regular lattice ring (which means increasing the granularity for the regular lattice ring part) and reduce the unstable energy formed by random reconnections. The second is to add an energy pledge step for the WS small-world model to ensure consistent network energy.

For the first requirement, we take a neuron from the network as the centroid, compose the regular lattice ring, and then examine the neurons along the radial in terms of their impact on the network characteristics. We observed that in the WS small-world model, the neurons near the centroid have more overlapped neighbors with the regular lattice ring. While extrapolating along the radial, the increase in the clustering coefficient showed an exponential decline, and the neurons far from the centroid barely benefited from increasing the clustering coefficient. However, slight wiring with distant neurons may substantially reduce the average path length. Therefore, we assumed that the connection quantities along the radial obey the Gaussian distribution. For the second requirement, we integrate the validation step using the 2SAT Wan Abdullah method in the new small-world rewiring method to ensure consistent energy.

B. GAUSSIAN-DISTRIBUED WIRING FOR SMALL-WORLD NETWORK

We propose a new rewiring method for the WS small-world model to promote the stability of the network energy and the learning result accuracy. The method starts by selecting a neuron as the pattern centroid along the radial, which divides the pattern into data layers. Considering the impact of network properties, we assumed that the quantities of connections on each data layer obey the Gaussian distribution. x represents the data layer variable, and $f(x)$ represents the connection quantities on each data layer and can be written as equation (8).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{8}$$

$$\sigma = \frac{1}{C\sqrt{2\pi}} \tag{9}$$

$$f(x) = C \cdot e^{-\frac{1}{2}\left(C\sqrt{2\pi}\cdot x\right)^2} \tag{10}$$

C is the size of the regular lattice ring, μ is initialized to 0, andσ is the parameter that can adjust the centrality of distribution. While $x = 0$, we may decide σ by C. Substitute (9) in (8), and the connection distribution of each layer may be written as equation (10). Figure 10 plots the Gaussian distribution of neuron connections; we see that the connections are denser near the centroid, forming a highly clustered region that dominates the main part of the network energy. But in the distance area from the centroid, only a few connections substantially reduce the average path length. Therefore, the distribution of neuron connections changes the proportion of network energy, and part of random energy is shrunk. However, the regular lattice ring is expanded, thus improving the stability of the learning accuracy. In addition, to ensure consistent network energy, we add the energy validation step by the 2SAT Wan Abdullah method.

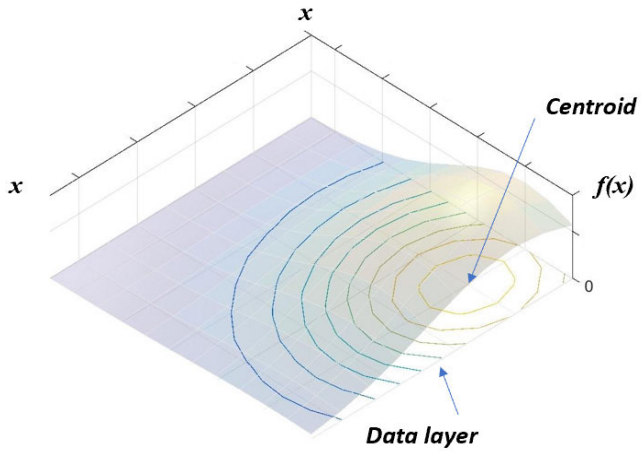

FIGURE 10. Gaussian distribution small-world neuron wiring.

The detailed steps of the Gaussian-distributed small-world rewiring algorithm are as below:

Step 1: Choose parameter k and rewiring probability P for the small-world topology.

Step 2: Take the neuron as the centroid of the pattern, Divide the pattern into x data layers.

Step 3: Connect the neuron with its k nearest neighbor neurons as a cluster. Denote C as the size of the cluster.

Step 4: Initialize the rewiring quantity $R = 0$ and generate $K/2$ random numbers. Check for every random number. If it is smaller than the preset rewiring probability P, increase one to R.

Step 5: If the random number is greater than the rewiring probability P, connect the neuron with the follow-up neuron. Increase the cluster size C by one.

Step 6: Taking the current neuron as the cluster center, divide the pattern into data layers along the radial. Initialize σ by equations (8) and (9).

Step 7: Compute the connection quantity for each layer by equation (10) and initialize $x = 1$.

Step 8: Wire neurons randomly to the related layer.

Step 9: Validate the energy for the neuron by the 2SAT Wan Abdullah method. Restart rewiring the neurons to the related layer if energy is inconsistent.

C. EVALUATIONS AND RESULTS

We evaluated the Gaussian-distributed small-world rewiring method using two aspects. 1) The coincidence degree with the fully connected neuron series - The similarity of the new small-world series with the fully connected series was measured to demonstrate that the small-world may obtain the same accurate result as the fully connected series. 2) The consistency of the energy- By evaluating the consistent energy of the small-world series formed by the new method, we further confirm the stability of the Gaussian distribution rewiring method.

To evaluate the approximation between the small-world series generated by the new method with the fully connected structure, we employed dynamic time warping (DTW) to measure the coincidence for these two series with different sizes. Figure 11 illustrates three small-world neuron series compared with the fully connected series, which formed on different regions of 100 neurons. In most places, the neuron series of the small-world highly coincides with the fully connected series, while the differentials appear at only a few positions. This explains the reason for the new rewiring mechanism's high learning accuracy. The new Gaussian-distributed rewiring mechanism ensures high clustering centrality in the near region of the centroid and slight rewiring in the distant region, therefore maintaining stability. The rewire mechanism reserved randomness regarding the rewire probability P to ensure that the disordered small-world topology is formed, further measuring its convergence trends to ensure the stability of output results.

To verify that the neuron series of the small-world structure is accurately converged, we evaluated its logic inconsistent energy using the Wan Abdullah method [7]. We randomly chose a neuron (state is 1, occasionally) from the network,

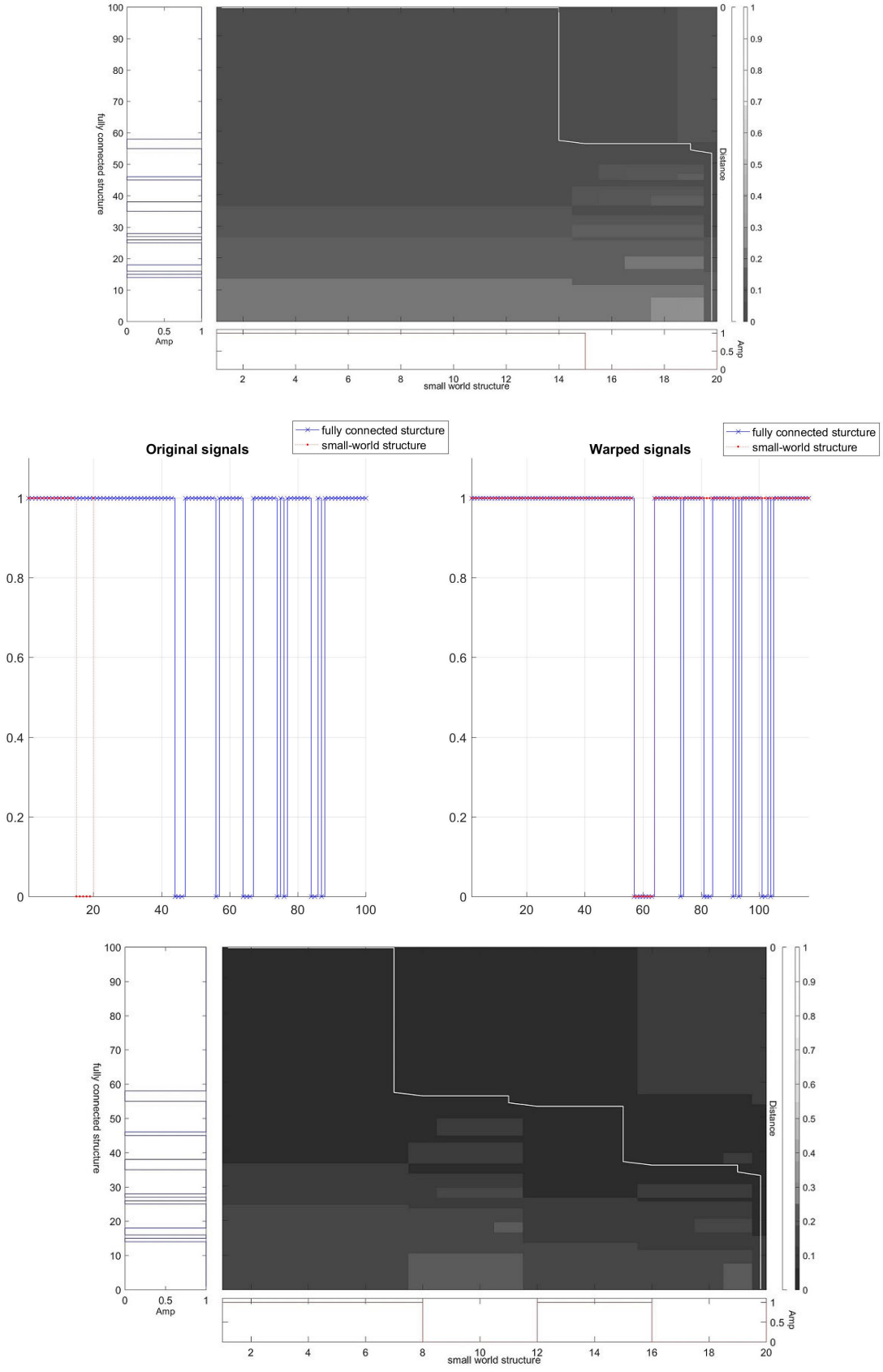

FIGURE 11. Comparison of the small-world series and the fully connected series by dynamic time warping.

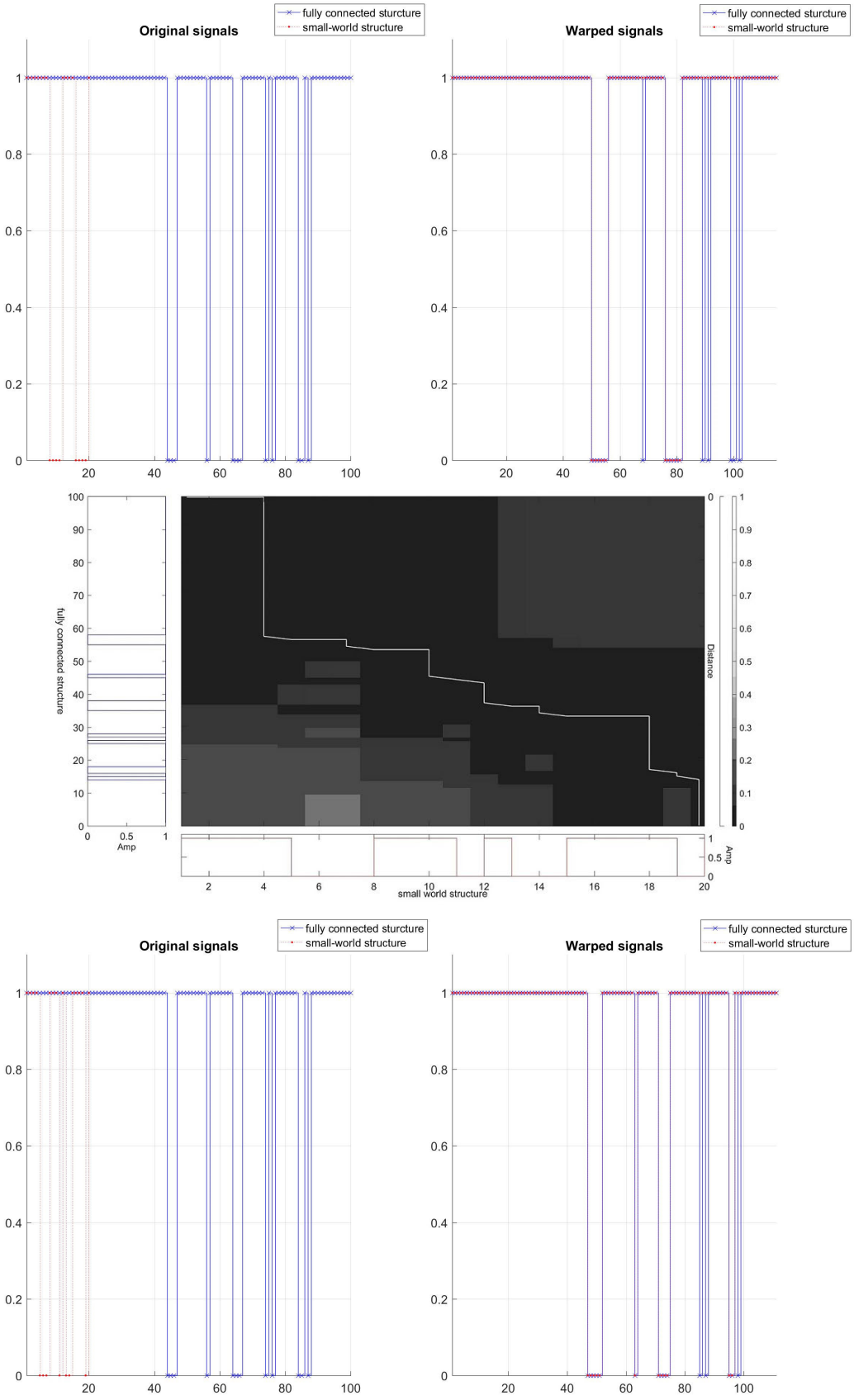

FIGURE 11. *(Continued.)* **Comparison of the small-world series and the fully connected series by dynamic time warping.**

TABLE 3. Wan Abdullah method for 3-satisfiability.

	C1 $\neg P \wedge \neg Q \wedge \neg R$	C2 $P \wedge \neg Q \wedge \neg R$	C3 $\neg P \wedge Q \wedge \neg R$	C4 $\neg P \wedge \neg Q \wedge R$	C5 $P \wedge \neg Q \wedge \neg R$	C6 $P \wedge \neg Q \wedge R$	C7 $\neg P \wedge Q \wedge R$	C8 $P \wedge Q \wedge R$	C9 $\neg S \wedge \neg T \wedge \neg U$
W_P	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	0
W_Q	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	0
W_R	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	0
W_{PQ}	$-\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	0
W_{QR}	$-\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	0
W_{PR}	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	0
W_{PQR}	$\frac{1}{16}$	$-\frac{1}{16}$	$-\frac{1}{16}$	$-\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	0
W_S	0	0	0	0	0	0	0	0	$\frac{1}{8}$
W_T	0	0	0	0	0	0	0	0	$\frac{1}{8}$
W_U	0	0	0	0	0	0	0	0	$\frac{1}{8}$
W_{ST}	0	0	0	0	0	0	0	0	$-\frac{1}{8}$
W_{TU}	0	0	0	0	0	0	0	0	$-\frac{1}{8}$
W_{SU}	0	0	0	0	0	0	0	0	$-\frac{1}{8}$
W_{STU}	0	0	0	0	0	0	0	0	$\frac{1}{16}$

and then set a small value of k (k = 6) as the starting value. This forms the neuron series of the small-world structure for this neuron, and then k increases gradually (interval = 10) till k equals the size of the entire network. We verified the logic inconsistent energy on the neuron series and calculated the state of the neuron. The CNF starts from k variables corresponding to k neuron small-world series. Each clause is composed of three neurons, and the neuron state represents each literal of the clause. For example, the RAN3-SAT CNF may be written as $\alpha = (S_1 \vee S_2 \vee \neg S_3) \wedge (S_4 \vee \neg S_5 \vee S_6)$, where k starts from 6, and the small-world series may be "110101." By negation of the CNF, we obtain $\neg\alpha = (\neg S_1 \wedge \neg S_2 \wedge S_3) \vee (\neg S_4 \wedge S_5 \wedge \neg S_6)$. We can compute the weight between neurons by applying the 3-SAT Wan Abdullah method in Table 3. Hence, we can verify the logic inconsistent energy by computing the state of the neuron. Then, we increase parameter k and the variables for the CNF, and we verify the logic inconsistent energy till k equals n, which is the network size.

On the left-hand side of Figure 12, the curve shows the convergence trends of the standard deviations obtained from twelve rounds of repeat testing. The abscissa denotes the k value, and the data points of the ordinate represent the corresponding standard deviation. When $k < 26$, the results of the remaining incorrect neuron states (state of 0) can be

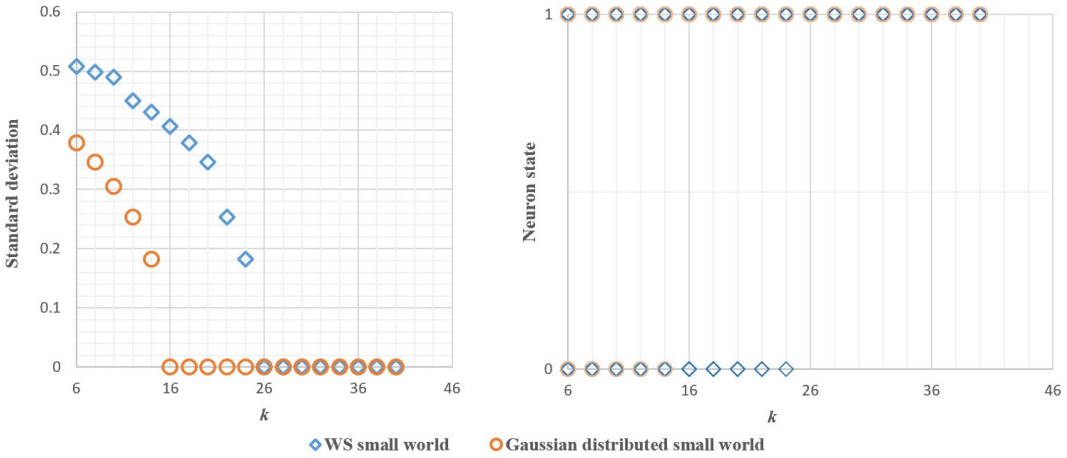

FIGURE 12. Standard deviation and neuron state convergence trends.

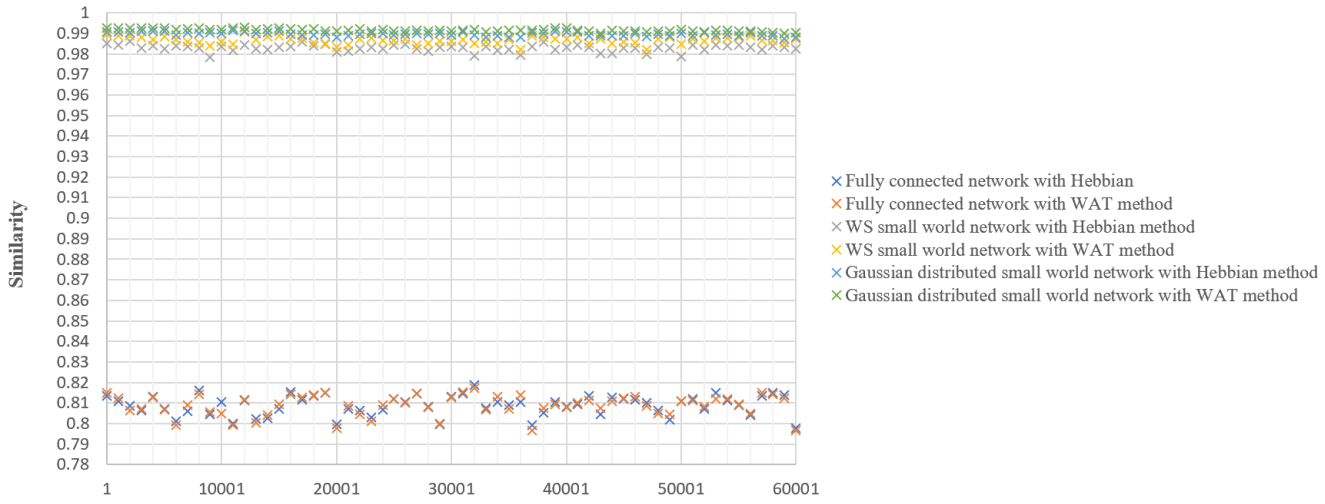

FIGURE 13. Similarities obtained from two training cycles.

obtained. By increasing k, the number of results with the incorrect state is reduced, and the related standard deviation converges toward zero rapidly. In the figure on the left-hand side, the data points represent the corresponding neuron states obtained from the twelve rounds of testing. The straight line displays the trend that when k increases, the inconsistent energy drives the neuron convergence towards its original state. Because neurons are sampled randomly in repeated testing, when the sample neuron has 0 as the original state, it accurately displays the same trend and convergence toward zero.

D. DIGIT RECOGNITION EXPERIMENT

1) MEMORIZATION AND RECOGNITION

To clarify the memorization and the recognition capability of the Gaussian-distributed small-world HNN, we tested it on a handwritten digit dataset in terms of the learning accuracy and convergence efficiency. Digit recognition is a classic property of the HNN. To measure the performance of the Gaussian-distributed small-world HNN algorithm, we used the MNIST dataset [27], [28] as the testing dataset. It is a mainstream dataset in digit recognition, which contains 60,000 handwritten digit training samples and 10,000 testing samples. Each sample has been standardized to a 28×28 pixels grayscale image. We measured similarity in the MNIST dataset by comparing the retrieved and original patterns. The measurement covered 60,000 training samples and was conducted using different configurations of learning methods and topologies. However, when measuring the similarity under these configurations, we observed that all the configurations could obtain a result that has a similarity equal to one, after just a few rounds of training. Meanwhile, we obtained accurate recognition results on almost all of the testing images.

In Figure 13, from the plotted digit testing results, we can observe that the small-world network is significantly ahead of the fully connected topology in terms of learning accuracy.

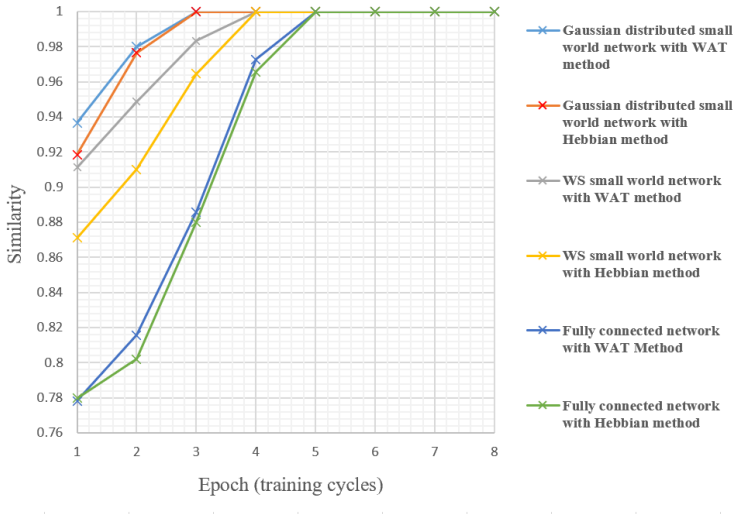

FIGURE 14. Convergence efficiencies of training cycles.

Under the Gaussian-distributed small-world network, the similarity obtained by the Wan Abdullah method and the Hebbian method exceeds 0.99. Compared with the WS small-world network, the similarity obtained are more stable, with only a few changes. Under the WS small-world network, the median value of the similarity obtained by the Wan Abdullah method was 0.988 and that of the Hebbian method was 0.983, but the extent of change is more significant.

Because these configurations converged after just a few rounds of training (epoch $<$ 10), we dumped the data of 10 training cycles and computed the median value for each configuration to further compare the performance of these configurations. In Figure 14, the Gaussian-distributed small-world configuration shows convergence to a target pattern within the 3rd training cycle, which is faster than the WS small-world configuration, and the fully connected neural network converged at the 5th training cycle. Therefore, after comparisons, it can be concluded that the Gaussian-distributed small-world neural network displays better convergence efficiency. Meantime, the Wan Abdullah method, which computes the synaptic weight by logical inconsistency energy of neurons, always shows a slight advantage throughout the testing, especially in terms of convergence efficiency.

Using the MNIST dataset, we confirmed the applicability of the Gaussian-distributed small-world neural network in terms of the learning accuracy of handwritten digit recognition. The Gaussian-distributed wiring mechanism improved the clustering coefficient for the small-world network and reduced the unstable energy caused by random rewiring. The reserved slight rewiring substantially reduced the average path length; therefore, higher learning accuracy and convergence efficiency were achieved compared with the WS small-world and fully connected networks. In addition, the small-world network is the only complex network that has confirmed its biological existence; therefore, we did not perform comparisons with other complex networks.

2) HANDWRITTEN DIGIT IMAGE CLASSIFICATION

Many methods have achieved impressive results in handwritten digit image classification. The MNIST benchmark database [28] shows that the convolutional neural network is the most accurate model for handwritten digit classification. Jarrett *et al.* [29] reported their CNN model with multistage feature exaction obtained a test error rate of 0.53. Cireşan *et al.* [30] employed a plain multi-layer perceptron (MLP) in their DNN (deep neural network) model and yielded a 0.35% test error rate. In this section, we test the performance of the proposed Gaussian-distributed small-world HNN from a practical application view. Because the HNN is a feedback neural network [1] [31], to better test performance, we integrated the Gaussian-distributed small-world HNN with Cireşan's DNN model as a solution for handwritten digit image classification [32]. Figure 15 illustrates the architecture of the Gaussian-distributed SWHNN-DNN of the training and testing stages.

The training architecture of Gaussian-distributed SWHNN-DNN contains a Hopfield layer [31] and a DNN layer [30]. The Hopfield layer contains ten units of the Gaussian-distributed SWHNN, which are used for training the digit images of the digits from "0" to "9", respectively. As shown in Figure 15, the Gaussian-distributed SWHNN unit organizes 784 neurons with Gaussian-distributed small-world wiring. The DNN layer is trained in a separate process. Its hidden layer is initialized by five fully connected layers in which the number of neurons of each layer is 2500, 2000, 1500, 1000, and 500. The output layer (SoftMax layer) contains ten neurons that use the "SoftMax" function as the activation function to output classification results.

Under the testing architecture, each test image is input into the Gaussian-distributed SWHNN units in parallel. We restricted the Gaussian-distributed SWHNN unit from memorizing test images, which means the weight matrix in the SWHNN unit is protected and test images are running on

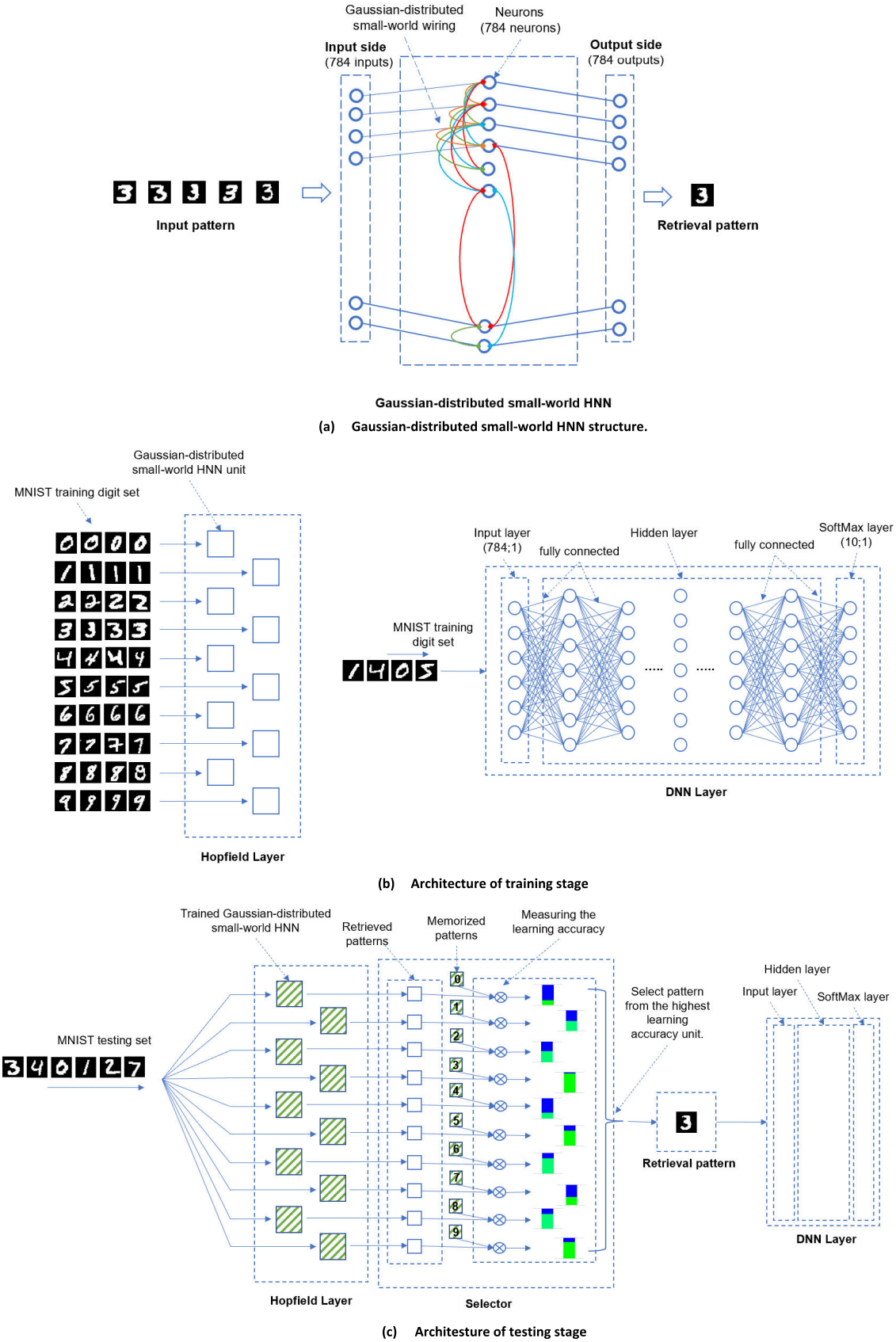

FIGURE 15. Architecture of the Gaussian-distributed SWHNN-DNN.

TABLE 4. Test error rates on MNIST handwritten digit dataset.

Architecture of DNN layer			Test error (%) in [30]	Training simulation time (h) in [30]	Training simulation time (h) in this study	Weights (millions) in [30]
Input layer	Hidden layer	SoftMax layer				
784	{1000,500}	10	0.49	23.4	10.6	1.34
	{1500,1000,500}		0.46	44.2	18.5	3.26
	{2000,1500,1000,500}		0.41	66.7	30.5	6.69
	{2500,2000,1500,1000,500}		0.35	114.5	55.1	12.11
	{9×1000}		0.44	107.7	50.8	8.86

DNN layer {input = 784, SoftMax = 10}	Test error (%) Gaussian-distributed SWHNN-DNN		Training simulation time (h) Gaussian-distributed SWHNN	Weights (thousands) Gaussian-distributed SWHNN
	Best test error (%)	Average test error (%)		
{1000,500}	0.28	0.37	2.1	203.84
{1500,1000,500}	0.22	0.31		
{2000,1500,1000,500}	0.19	0.24		
{2500,2000,1500,1000,500}	0.16	0.19		
{9×1000}	0.19	0.23		

the copy of weight matrices. The selector module is applied to select the SWHNN channel with the highest learning accuracy. Learning accuracy is the similarity between the memorized pattern and the pattern retrieved from SWHNN units [32], [33], [34]. The pattern from the selected channel is retrieved and submitted to the DNN layer for classification.

During the training stage, first, we trained the HNN layer. We manually divided the MNIST training set into ten clusters of digits "0" to "9", and these digit images were submitted to Gaussian-distributed SWHNN units for training. The Hebbian method was employed to update the weight matrix of the Gaussian-distributed SWHNN unit, and these matrices were saved for further tests. Then the DNN layer was trained under diverse configurations. Table 4 lists the details of the configurations of the DNN layer.

We tested on an Intel Core i7 11800, 16GB memory, and Nvidia GTX3060 GPU hardware platform. As shown in Table 4, the simulation time is ahead of that measured in [30] due to the difference in hardware platforms. Through testing simulations, we observed that the Gaussian-distributed SWHNN unit may effectively attract test images to converge to the trained pattern; hence, it improved the test error of the DNN layer during the testing process. We listed the test error rates of the Gaussian-distributed SWHNN-DNN with different structures of the hidden layer at the bottom of Table 4. The lowest test error rate of the Gaussian-distributed SHWNN-DNN appeared in the hidden layer of structure {2500,2000,1500,1000,500}, and it arrived at 0.16%. Its average test error rate collected from five rounds of repeating tests is 0.19, is listed in the right column. The test error rates of the rest of the hidden layers are lower than that of the DNN in [30]. The test error rate of {1000,500} is 0.28 and its average test error is 0.37, {1500,1000,500} is 0.22 and 0.31, {2000,1500,1000,500} is 0.19 and 0.24, and {9 × 1000} is 0.19 and 0.23, respectively. In addition, from the memory occupation aspect, the weight matrix of the Gaussian-distributed SWHNN unit is set to 26 × 784. Therefore, 203.84K weights are used in the ten Gaussian-distributed SWHNN units in the Hopfield layer. Through the comparison testing, we noted that the associative memory property of SWHNN units provides enhancement and correction functions for testing patterns; hence, it reduced the classification complexity for the DNN layer. The above comparison data shows that the Gaussian-distributed SWHNN-DNN obtained a lower test error rate than the DNN model. We further compare the Gaussian-distributed SWHNN-DNN with other methods to know its performance.

The comparison of the Gaussian-distributed SWHNN-DNN with other methods for handwritten digit image classification is listed in Table 5. From the test error rates in Table 5, we can observe that the best test error rate of

TABLE 5. Comparison of the Gaussian-distributed SWHNN-DNN with other methods on MNIST handwritten digit dataset.

Methods	References	Test error (%)
CNN	[29]	0.53
CNN	[35]	0.23
DNN	[30]	0.35
Gaussian-distributed SWHNN-DNN	this paper	0.16
Hybrid CNN-SVM	[36]	0.19

the Gaussian-distributed SWHNN-DNN is lower than other listed CNN methods. From training complexity aspects, the architecture of the proposed Gaussian-distributed SWHNN-DNN is even simpler, including only a plain MLP network and an HNN layer. Therefore, its computational cost and training complexity are also lower than CNN models. Thus, through the above comparison, we conclude that the proposed Gaussian-distributed SWHNN-DNN outperforms other CNN models on the MNIST handwritten digit set.

In addition, Namane *et al.* [32] proposed a method combined fully connected HNN and MLP for degraded machine-printed digit recognition, they report obtained a 98.62% recognition rate. We tested HNN-DNN under hidden layer of {2000,1500,1000,500}. We obtained 92 misclassified images as the lowest test error. The best test error rate is 0.92%, and the average test error of five rounds repeat testing is 1.06%.

3) DISCUSSION

The proposed Gaussian distribution small-world model optimizes the small-world network in terms of the following aspects: 1) short average path length- during the training process of a small-world HNN, the average path length plays an important role in reducing training cycles. The proposed Gaussian-distributed small-world network contains only slight random wirings to the distant neurons, significantly reducing the average path length, thus improving training efficiency. In the memorization capability test, the proposed neural network converged in only a few training cycles (epoch $<$ 10). 2) High clustering coefficient- in the small-world model, the network clustering coefficient illustrates the degree to which the adjacent neurons of a neuron are connected. During the training process, the clustering coefficient can significantly affect the transfer efficacy of network energy. The proposed method arranges neuron connections obeying Gaussian distribution, which concentrates connections in the near neuron region, thus improving network energy's transfer efficacy and obtaining high learning accuracy and fast convergence speed. 3) Stable network energy- the proposed method changed the distribution of neuron connections in the small-world network, altering the network energy proportion. It reduced the random connections, thus reducing the random energy in the network. Therefore, the stable energy formed by the neurons in the near region of the centroid neuron dominated the main part of the network.

In addition, the Gaussian-distribution small-world wiring method is dedicated to improving the instability deficiency of the WS small-world model. However, it differs from the hyper-parameter optimization method in deep learning. We summarize the key points. 1) The proposed Gaussian-distributed small-world network does not leverage the optimal parameter combinations for improving training performance. The parameter combinations of k and P cannot determine the connection between a pair of neurons; one combination of k and P may form different small-world networks. 2) The workflow of the Gaussian-distributed small-world wiring method differs from that of the hyper-parameter optimization method. The hyper-parameter optimization method usually contains an expensive step of finding the optimal parameters. The Gaussian-distributed small-world wiring method does not include such steps. 3) These two methods have different perspectives on optimizing the training of neural networks. The hyper-parameter optimization method more focuses on the pros and cons of the combination of configurations of neural networks. However, the Gaussian-distributed small-world method emphasizes the biological reality and the stability of the network energy of the HNN.

V. CONCLUSION

In this study, to address the instability issue of the small-world network, we examined the influence of the small-world topology on the HNN learning process and conducted computer simulations. We observed that instabilities due to the random existence of neuron connections cause unstable network energies, which may generate oscillations during the WS small-world neural network training process. Therefore, we proposed the Gaussian-distributed small-world wiring method to improve the stability of WS small-world networks. The proposed method organizes neuron connections in compliance with the Gaussian-distribution, which reduces random connections from a distant area and makes the short-range connections dominate the main part of network energy, thus improving the stability of small-world networks. To evaluate the new small-world rewiring method, we compared the new small-world series with the fully connected series by applying DTW. The new small-world series displayed high levels of coincidence compared with the fully connected series in terms of the distributions of neuron states

and the logical energy value. The new Gaussian-distribution rewiring method was compared with WS small-world method and the fully connected neural network on the MNIST handwritten digit dataset. The experimental result revealed that the Gaussian-distribution rewiring method performed with higher stability in terms of the learning accuracy and had a higher convergence speed than the WS small-world model and the fully connected network. In the application comparison experiment, we integrated the Gaussian-distributed small-world neural network with Ciresan's DNN model and obtained the best test error rate of 0.16%.

REFERENCES

[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982, doi: 10.1073/pnas.79.8.2554.

[2] N. M. Nasrabadi and W. Li, "Object recognition by a Hopfield neural network," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 6, pp. 1523–1535, Nov./Dec. 1991, doi: 10.1109/21.135694.

[3] J. K. Paik and A. K. Katsaggelos, "Image restoration using a modified Hopfield network," *IEEE Trans. Image Process.*, vol. 1, no. 1, pp. 49–63, Jan. 1992, doi: 10.1109/83.128030.

[4] S. M. Gowda, B. W. Lee, and B. J. Sheu, "An improved neural network approach to the traveling salesman problem," in *Proc. 4th IEEE Region Int. Conf. (TENCON)*, Bombay, India, Nov. 1989, pp. 552–555.

[5] M. Verleysen and P. G. A. Jespers, "An analog VLSI implementation of Hopfield's neural network," *IEEE Micro*, vol. 9, no. 6, pp. 46–55, Dec. 1989, doi: 10.1109/40.42986.

[6] W. A. T. W. Abdullah, "Logic programming on a neural network," *Int. J. Intell. Syst.*, vol. 7, no. 6, pp. 513–519, Aug. 1992.

[7] S. Sathasivam and W. A. T. Wan Abdullah, "Logic mining in neural network: Reverse analysis method," *Computing*, vol. 91, no. 2, pp. 119–133, Feb. 2011.

[8] M. A. Mansor, M. S. M. Kasihmuddin, and S. Sathasivam, "Enhanced Hopfield network for pattern satisfiability optimization," *Int. J. Intell. Syst. Appl.*, vol. 8, no. 11, pp. 27–33, Nov. 2016, doi: 10.5815/ijisa.2016.11.04.

[9] M. S. M. Kasihmuddin, M. A. Mansor, and S. Sathasivam, "Hybrid genetic algorithm in the Hopfield network for logic satisfiability problem," *Pertanika J. Sci. Technol.*, vol. 25, pp. 139–152, Jan. 2017.

[10] S. Sathasivam, M. A. Mansor, A. I. M. Ismail, S. Z. M. Jamaludin, M. S. M. Kasihmuddin, and M. Mamat, "Novel random K satisfiability for in Hopfield neural network," *Sains Malaysiana*, vol. 49, no. 11, pp. 2847–2857, Nov. 2020, doi: 10.17576/jsm-2020-4911-23.

[11] G. Serpen, "Managing spatio-temporal complexity in Hopfield neural network simulations for large-scale static optimization," *Math. Comput. Simul.*, vol. 64, no. 2, pp. 279–293, Jan. 2004, doi: 10.1016/j.matcom.2003.09.023.

[12] L. Calcraft, R. Adams, and N. Davey, "High performance associative memory models with low wiring costs," in *Proc. 3rd Int. IEEE Conf. Intell. Syst.*, London, U.K., Sep. 2006, pp. 4–6.

[13] V. Braitenberg and A. Schüz, *Cortex: Statistics and Geometry of Neuronal Connectivity*. Berlin, Germany: Springer, 1998, doi: 10.1007/978-3-662-03733-1.

[14] Y. Kawai, J. Park, and M. Asada, "A small-world topology enhances the echo state property and signal propagation in reservoir computing," *Neural Netw.*, vol. 112, pp. 15–23, Apr. 2019, doi: 10.1016/j.neunet.2019.01.002.

[15] S. Achard and E. Bullmore, "Efficiency and cost of economical brain functional networks," *PLoS Comput. Biol.*, vol. 3, no. 2, p. e17, Feb. 2007, doi: 10.1371/journal.pcbi.0030017.

[16] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998, doi: 10.1038/30918.

[17] J. Bohland and A. Minai, "Efficient associative memory using small-world architecture," *Neurocomputing*, vol. 38, pp. 489–496, Jun. 2001, doi: 10.1016/s0925-2312(01)00378-2.

[18] P. N. McGraw and M. Menzinger, "Topology and computational performance of attractor neural networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, no. 4, Oct. 2003, Art. no. 047102, doi: 10.1103/PhysRevE.68.047102.

[19] Z. Chen, R. Zhang, H. Huo, P. Liu, C. Zhang, and T. Feng, "Functional connectome of human cerebellum," *NeuroImage*, vol. 251, May 2022, Art. no. 119015, doi: 10.1016/j.neuroimage.2022.119015.

[20] B. Q. Rosen and E. Halgren, "An estimation of the absolute number of axons indicates that human cortical areas are sparsely connected," *PLOS Biol.*, vol. 20, no. 3, Mar. 2022, Art. no. e3001575, doi: 10.1371/journal.pbio.3001575.

[21] T. Pircher, B. Pircher, E. Schlücker, and A. Feigenspan, "The structure dilemma in biological and artificial neural networks," *Sci. Rep.*, vol. 11, no. 1, pp. 1–6, Dec. 2021, doi: 10.1038/s41598-021-84813-6.

[22] S. Arvin, A. N. Glud, and K. Yonehara, "Short- and long-range connections differentially modulate the dynamics and state of small-world networks," *Frontiers Comput. Neurosci.*, vol. 15, p. 124, Jan. 2022, doi: 10.3389/fncom.2021.783474.

[23] S. Rüdiger, A. Plietzsch, F. Sagués, I. M. Sokolov, and J. Kurths, "Epidemics with mutating infectivity on small-world networks," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, Dec. 2020, doi: 10.1038/s41598-020-62597-5.

[24] M. Ercsey-Ravasz, N. T. Markov, C. Lamy, D. C. Van Essen, K. Knoblauch, Z. Toroczkai, and H. Kennedy, "A predictive network model of cerebral cortical connectivity based on a distance rule," *Neuron*, vol. 80, no. 1, pp. 184–197, Oct. 2013, doi: 10.1016/j.neuron.2013.07.036.

[25] S. Oldham, B. D. Fulcher, K. Aquino, A. Arnatkevičiūtė, C. Paquola, R. Shishegar, and A. Fornito, "Modeling spatial, developmental, physiological, and topological constraints on human brain connectivity," *Sci. Adv.*, vol. 8, no. 22, p. eabm6127, Jun. 2022, doi: 10.1126/sciadv.abm6127.

[26] K. Takagi, "Energy constraints on brain network formation," *Sci. Rep.*, vol. 11, no. 1, pp. 1–8, Dec. 2021, doi: 10.1038/s41598-021-91250-y.

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[28] *MNIST Handwritten Digit Dataset*. Accessed: Mar. 25, 2022. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[29] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep. 2009, pp. 2146–2153, doi: 10.1109/ICCV.2009.5459469.

[30] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, 2010.

[31] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, T. Adler, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, V. Greiff, D. Kreil, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter, "Hopfield networks is all you need," 2020, *arXiv:2008.02217*.

[32] A. Namane and P. Meyrueis, "Multiple classifier for degraded machine printed character recognition," in *Proc. Colloque Int. Francophone Sur L'écrit et Le Document*, Rouen, France, Oct. 2008, pp. 187–192. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00334417

[33] C. Draganova, A. Lanitis, and C. Christodoulou, "Restoration of partially occluded shapes of faces using neural networks," in *Computer Recognition Systems*. Berlin, Germany: Springer, 2005, pp. 767–774, doi: 10.1007/3-540-32390-2_90.

[34] F. E. Keddous and A. Nakib, "Optimal CNN–Hopfield network for pattern recognition based on a genetic algorithm," *Algorithms*, vol. 15, no. 1, p. 11, Dec. 2021, doi: 10.3390/a15010011.

[35] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3642–3649, doi: 10.1109/CVPR.2012.6248110.

[36] X.-X. Niu and C. Y. Suen, "A novel hybrid CNN–SVM classifier for recognizing handwritten digits," *Pattern Recognit.*, vol. 45, no. 4, pp. 1318–1325, Apr. 2012, doi: 10.1016/j.patcog.2011.09.021.

[37] L. G. Morelli, G. Abramson, and M. N. Kuperman, "Associative memory on a small-world neural network," *Eur. Phys. J. B-Condens. Matter Complex Syst.*, vol. 38, no. 3, pp. 495–500, Apr. 2004, doi: 10.1140/epjb/e2004-00144-7.

[38] S. Duan, Z. Dong, X. Hu, L. Wang, and H. Li, "Small-world Hopfield neural networks with weight salience priority and memristor synapses for digit recognition," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 837–844, May 2016, doi: 10.1007/s00521-015-1899-7.

[39] D. Dominguez, K. Koroutchev, E. Serrano, and F. B. Rodríguez, "Information and topology in attractor neural networks," *Neural Comput.*, vol. 19, no. 4, pp. 956–973, Apr. 2007, doi: 10.1162/neco.2007.19.4.956.

[40] W. A. T. W. Abdullah and S. Sathasivam, "Logic learning in Hopfield networks," in *Proc. Int. Conf. Intell. Syst.*, Kuala Lumpur, Malaysia, May 2005, pp. 25–28.
[41] W. A. T. W. Abdullah, "The logic of neural networks," *Phys. Lett. A*, vol. 176, pp. 202–206, May 1993.
[42] M. S. M. Kasihmuddin, M. A. Mansor, and S. Sathasivam, "Discrete Hopfield neural network in restricted maximum K-satisfiability logic programming," *Sains Malaysiana*, vol. 47, no. 6, pp. 1327–1335, Jun. 2018, doi: 10.17576/jsm-2018-4706-30.
[43] N. Eggemann and S. D. Noble, "The clustering coefficient of a scale-free random graph," *Discrete Appl. Math.*, vol. 159, no. 10, pp. 953–965, Jun. 2011, doi: 10.1016/j.dam.2011.02.003.
[44] L. Gu, H. L. Huang, and X. D. Zhang, "The clustering coefficient and the diameter of small-world networks," *Acta Math. Sinica, English Ser.*, vol. 29, no. 1, pp. 199–208, Jan. 2013, doi: 10.1007/s10114-012-0387-6.
[45] A. Barrat and M. Weigt, "On the properties of small-world network models," *Eur. Phys. J. B-Condens. Matter Complex Syst.*, vol. 13, no. 3, pp. 547–560, Jan. 2000, doi: 10.1007/s100510050067.
[46] A. Fronczak, P. Fronczak, and J. A. Hołyst, "Average path length in random networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 5, Nov. 2004, Art. no. 056110, doi: 10.1103/physreve.70.056110.
[47] G. Mao and N. Zhang, "A multilevel simplification algorithm for computing the average shortest-path length of scale-free complex network," *J. Appl. Math.*, vol. 2014, pp. 1–6, Jun. 2014, doi: 10.1155/2014/154172.
[48] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, Mar. 2005, doi: 10.1007/s10115-004-0154-9.
[49] S. Prasanna, S. Purushothaman, and A. Jothi, "Implementation of dynamic time warping for video indexing," *Int. J. Comput. Sci. Inf. Secur.*, vol. 9, no. 8, pp. 265–269, 2011.
[50] L. A. Braunstein, S. V. Buldyrev, R. Cohen, S. Havlin, and H. E. Stanley, "Optimal paths in disordered complex networks," *Phys. Rev. Lett.*, vol. 91, no. 16, Oct. 2003, Art. no. 168701, doi: 10.1103/physrevlett.91.168701.
[51] Q. Lin, P. Cai, and F. Zhang, "Image recognition via discrete Hopfield neural network," in *Proc. Int. Conf. Adv. Energy Eng.*, Beijing, China, Jun. 2010, pp. 339–342.
[52] M. A. Mansor, M. S. M. Kasihmuddin, and S. Sathasivam, "Grey wolf optimization algorithm with discrete Hopfield neural network for 3 satisfiability analysis," *J. Phys. Conf.*, vol. 1821, Mar. 2021, Art. no. 012038, doi: 10.1088/1742-6596/1821/1/012038.
[53] M. S. M. Kasihmuddin, M. A. Mansor, M. F. M. Basir, S. Z. M. Jamaludin, and S. Sathasivam, "The effect of logical permutation in 2 satisfiability reverse analysis method," in *Proc. Int. Conf. Adv. Mater. Res. (ICAMR)*, 2020, Art. no. 040013, doi: 10.1063/5.0019158.
[54] M. A. Belyaev and A. A. Velichko, "Classify handwritten digits using Hopfield network," in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, vol. 862, 2020, Art. no. 052048.
[55] D.-L. Lee, "New stability conditions for Hopfield networks in partial simultaneous update mode," *IEEE Trans. Neural Netw.*, vol. 10, no. 4, pp. 975–978, Jul. 1999, doi: 10.1109/72.774276.
[56] M. Banyai, T. Nepusz, L. Negyessy, and F. Bazso, "Convergence properties of some random networks," in *Proc. 7th Int. Symp. Intell. Syst. Informat.*, Sep. 2009, pp. 241–245.
[57] R.-R. Liu, W.-X. Wang, Y.-C. Lai, G. Chen, and B.-H. Wang, "Optimal convergence in naming game with geography-based negotiation on small-world networks," *Phys. Lett. A*, vol. 375, no. 3, pp. 363–367, Jan. 2011, doi: 10.1016/j.physleta.2010.12.007.
[58] M. Newman and D. Watts, "Renormalization group analysis of the small-world network model," *Phys. Lett. A*, vol. 263, pp. 341–346, Dec. 1999, doi: 10.1016/s0375-9601(99)00757-4.
[59] Z. P. Neal, "How small is it? Comparing indices of small worldliness," *Netw. Sci.*, vol. 5, no. 1, pp. 30–44, Mar. 2017, doi: 10.1017/nws.2017.5.
[60] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, Mar. 2001, doi: 10.1038/35065725.

JUN SUN (Member, IEEE) received the bachelor's degree in computer science from Shanghai University, in 2005, and the master's degree in computer software from Fudan University, Shanghai, China, in 2012. He is currently pursuing the Ph.D. degree with Universiti Sains Malaysia. His research interests include neural networks, artificial intelligence, complex networks, and logic programming.

SARATHA SATHASIVAM received the B.Sc. (Ed) and M.Sc. degrees from Universiti Sains Malaysia, and the Ph.D. degree from Universiti Malaya, Malaysia. She is currently an Associate Professor with the School of Mathematical Sciences, Universiti Sains Malaysia. She has published widely in journals and proceedings and has collaboration with researchers from different countries. Her research interests include neural networks, agent-based modeling, data mining, and constrained optimization problem.

MAJID KHAN BIN MAJAHAR ALI received the Ph.D. degree in mathematics from Universiti Malaysia Sabah. He joined the School of Mathematical Sciences, Universiti Sains Malaysia, as a Lecturer (Operational Research), in May 2017. He is currently a Researcher and appointed as a fellow working in the field of complex system modeling and simulation. His research interests include big data, mathematical modeling for complex problem, and application of artificial intelligence.