BAREKENG: Journal of Mathematics and Its Applications
September 2026 Volume 20 Issue 3 Page 2349-2362
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol20iss3pp2349-2362

IMPROVING NEURON STATE DIVERSIFICATION IN LOGIC SATISFIABILITY VIA SMISH ACTIVATION AND AN ENHANCED UPDATING RULE

Nurshazneem Roslan[1*], **Saratha Sathasivam**[2]

[1]*Department of Mathematical Sciences, Faculty of Intelligent Computing, Universiti Malaysia Perlis Kampus Alam UniMAP, Pauh Putra, Perlis, Arau 02600, Malaysia*

[2]*School of Mathematical Sciences, Universiti Sains Malaysia Penang, USM, 11800, Malaysia*

Corresponding author's e-mail: * **nurshazneem@unimap.edu.my**

Article Info

Article History:

Received: 25th August 2025
Revised: 20th January 2026
Accepted: 16th March 2026
Available online: 8th April 2026

Keywords:

Conditional Random 2-Satisfiability;
Discrete Hopfield Neural Network;
Election Algorithm;
Neuron state diversification;
Similarity measure.

ABSTRACT

A deeper understanding of the learning and retrieval phase of the Discrete Hopfield Neural Network (DHNN) is essential for advancing its application in intelligent systems. This study investigates the performance of a non-systematic logical rule, namely Conditional Random 2 Satisfiability logic (CRAN2SAT) in DHNN (DHNN-CRAN2SAT) in retrieving diverse and optimal final neuron states. The findings show that the Election Algorithm consistently retrieves the maximum global minimum solution value of 1 across all tested neuron sizes, outperforms Exhaustive Search. In addition, the implementation of a new updating rule during the retrieval phase significantly enhances the diversity of final neuron states. This improvement is reflected by lower Sokal–Sneath similarity indices with an average value of 0.3809 and increased neuron state variation with an average value of 8809. These results highlight the significance of both the learning algorithm and updating strategy in the retrieval phase of DHNN. By enabling a broader range of final neuron states, this approach offers meaningful improvements for logic mining models, particularly in addressing real-world classification challenges.

How to cite this article:

N. Roslan and S. Sathasivam, "IMPROVING NEURON STATE DIVERSIFICATION IN LOGIC SATISFIABILITY VIA SMISH ACTIVATION AND AN ENHANCED UPDATING RULE", *BAREKENG: J. Math. & App.*, vol. 20, no. 3, pp. 2349-2362, Sep, 2026.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*
Research Article · Open Access

1. INTRODUCTION

An Artificial Neural Network (ANN) is a computational model inspired by the human brain, consisting of interconnected neurons that collaborate to solve complex real-world problems across domains such as medicine, business, and education. Among the various types of ANN, the Hopfield Neural Network (HNN) introduced by Hopfield and Tank in 1985 [1] has been widely utilized in applications, including the Travelling Salesman Problem (TSP), image recognition [2], [3] and combinatorial optimization [4]. HNNs are generally classified into continuous and discrete variants. The Discrete Hopfield Neural Network (DHNN) is a fully connected network where each neuron is linked to all others, and it operates without any hidden layers. The main characteristic of DHNN is its ability to function as an associative memory, particularly in the context of content-addressable memory (CAM), where it is used to store and recall problem patterns. Notably, DHNN employs an asynchronous updating process that allows neurons to update their states independently and thus reduces the risk of oscillations.

Due to these characteristics, DHNN has been effectively used in satisfiability (SAT) by incorporating logical rules within the ANN framework. This approach was pioneered by Wan Abdullah [5], where a logical inference method was introduced to minimize inconsistencies in the network. In addition, the Wan Abdullah method (WA method) was developed for calculating synaptic weights by comparing the cost function with the Lyapunov energy function. Subsequently, this approach was extended to a restricted subclass of SAT. In particular, Horn clauses were incorporated into the DHNN model in [6], resulting in the Horn satisfiability model (HornSAT), which represents a SAT formulation constrained by Horn clause structures. Relaxation mechanisms were further introduced during the retrieval phase of DHNN to improve performance. Since then, the WA method has been widely applied to both systematic [7], [8] and non-systematic logical rules [9], [10], [11]. One key limitation in the learning phase of DHNN is the reliance on the Exhaustive Search (ES) algorithm, which involves trial-and-error procedures and often results in suboptimal synaptic weights. These weights play a crucial role in determining the effectiveness of the retrieval phase [12]. Therefore, employing a more efficient learning algorithm is essential to generate optimal synaptic weights and, in turn, lead to an optimal final neuron state. The Election Algorithm (EA) effectively minimizes the cost function with fewer iterations, leading to an optimal synaptic weight [13]. Therefore, implementing the Election Algorithm is essential to improve the learning phase of DHNN.

A significant aspect of this study focuses on evaluating the quality of the retrieved state, which is influenced by the choice of activation function. Activation functions transform input signals into output signals. In the context of logic satisfiability, the variation of the final neuron state is closely related to the choice of activation function. Thus, selecting an appropriate activation function is crucial for promoting variation and avoiding overfitting. The Hyperbolic Tangent Activation Function (HTAF) introduced by [14] is a commonly used activation function. However, HTAF suffers from the vanishing gradient problem [15], [16], [17] which can result in overfitted outputs. More recently, the Smish activation function has been implemented in DHNN for non-systematic logic by [18]. The effectiveness of the Smish activation function in generating diversified final neuron states for non-systematic logical rules was demonstrated. These findings suggest that the use of a suitable activation function can significantly enhance variation and diversification in the resulting neuron states.

Neuron state diversification in DHNN expands the solution space of the network. This is crucial for logic mining tasks, where a larger solution space increases the chance of obtaining an optimal induced logic. Although the Smish activation function has improved the diversity of neuron states, further enhancement is still required in terms of similarity, which reflects how closely the retrieved neuron states match the ideal benchmark neuron states. Evaluating the uniqueness of the retrieved states to the benchmark requires similarity analysis. Recent advancements by [19] and [20] have proposed improvements to the updating rule in DHNN. The standard updating rule performs poorly when the local field is zero, often leading to overfitting of the final neuron state. In contrast, the new updating rule maintains the neuron state when a zero local field occurs, which leads to better quality in the retrieved states as measured by the similarity index. Consequently, the overall quality of the retrieved neuron states is significantly improved by adopting the updated rule. This study makes several key contributions. First, the Smish activation function is implemented into the DHNN framework to improve the effectiveness of the neuron updating rule for non-systematic logic. Second, an enhanced updating rule is employed to improve the diversification of the final neuron states during the retrieval phase. Finally, the quality and uniqueness of the retrieved neuron states are systematically evaluated with respect to benchmark solutions using similarity analysis.

Therefore, the combination of an effective activation function and a new updating rule in DHNN can lead to diversified states. The structure of this paper can be described as follows: Section 2 explores how non-systematic logic is formulated. Section 3 covers topics, including the implementation of Conditional Random 2 Satisfiability with Smish and a new updating rule in DHNN. The experimental setup is detailed in Section 4, followed by the presentation and analysis of results in Section 5. The paper concludes in Section 6, which also discusses possible future research directions.

2. RESEARCH METHODS

2.1 The Formulation of Conditional Random 2 Satisfiability Representation

Conditional Random k Satisfiability for $k = 1{,}2$ (CRAN2SAT) proposed by [18] is a non-systematic logic that consists of first-order and second-order clauses under a specific condition. CRAN2SAT provides flexibility by eliminating the use of two positive literals in second-order clauses and allowing unrestricted structure in first-order clauses. CRAN2SAT uses restricted negation to strengthen the control of negative neuron states within the network. This approach ensures that second-order clauses contain at least one negation operator while maintaining flexibility in first-order clauses. CRAN2SAT is structured based on the following key components:

1. A set of m second-order clauses, with each clause consisting of exactly two literals, is represented as $C_1^{(2)}, C_2^{(2)}, \dots, C_m^{(2)}$ such that $C_m^{(2)} = (B_i \vee D_i)$, $m \in Z^+$.
2. A set of n first-order clauses, with each clause consisting of exactly one literal, is represented as $C_1^{(1)}, C_2^{(1)}, \dots, C_n^{(1)}$ such that $C_n^{(1)} = (A_i)$, $n \in Z^+$.

Note that i represents the independent literals. The $C_n^{(1)}$ and $C_m^{(2)}$ denote the first-order and second-order clauses, respectively, where each literal can be represented as a positive or negative literal, such $B_i \in \{B_i, \neg B_i\}$, $D_i \in \{D_i, \neg D_i\}$ and $A_i \in \{A_i, \neg A_i\}$. The possible combinations of the first-order clauses are highlighted as in Eq. (1):

$$C_i^{(1)} \in \{(A_i), (\neg A_i)\}. \tag{1}$$

While the possible combinations of the second-order clauses are highlighted as in Eq. (2):

$$C_i^{(2)} \in \{(\neg B_i \vee \neg D_i), (B_i \vee \neg D_i), (\neg B_i \vee D_i)\}, \;\; C_i^{(2)} \notin \{(B_i \vee D_i)\}. \tag{2}$$

Based on (2), the possible combinations of the second-order clauses that consist of both positive literals are denoted as $(B_i \vee D_i)$ are excluded from the logical structure of CRAN2SAT. Each clause is combined with others using logical AND ($\wedge$) and the literals within a clause are connected by logical OR ($\vee$). Therefore, the general formulation of CRAN2SAT denoted as P_{CRAN2SAT} is presented in Eq. (3) and the definition of the clauses in P_{CRAN2SAT} is illustrated in Eq. (4):

$$P_{\text{CRAN2SAT}} = \wedge_{i=1}^{m} C_i^{(2)} \wedge_{i=1}^{n} C_i^{(1)}, \tag{3}$$

$$C_i^{(k)} = \begin{cases} (B_i \vee D_i), & k = 2 \\ A_i, & k = 1 \end{cases}. \tag{4}$$

Note that m refers to the total number of the second-order clauses, while the variable n corresponds to the total number of first-order clauses. Based on this formulation, Eqs. (5) until (8) illustrate the specific structure of P_{CRAN2SAT} generated for different values of the number m and n. These examples demonstrate how the general formulation can be extended to represent various structures of P_{CRAN2SAT}.

$$P_{\text{CRAN2SAT}} = (B_1 \vee \neg D_1) \wedge \neg A_1, \quad m = 1, \quad n = 1 \;, \tag{5}$$

$$P_{\text{CRAN2SAT}} = (\neg B_1 \vee D_1) \wedge (\neg B_2 \vee D_2) \wedge A_1 \wedge A_2 \quad m = 2, \quad n = 2 \;, \tag{6}$$

$$P_{\text{CRAN2SAT}} = (\neg B_1 \vee D_1) \wedge (\neg B_2 \vee \neg D_2) \wedge (B_3 \vee \neg D_3) \wedge A_1 \wedge \neg A_2 \wedge \neg A_3, \quad m = 3, \quad n = 3 \;, \tag{7}$$

$$P_{\text{CRAN2SAT}} = (\neg B_1 \vee \neg D_1) \wedge (B_2 \vee \neg D_2) \wedge (\neg B_3 \vee D_3) \wedge \neg A_1 \wedge \neg A_2 \wedge \neg A_3, \quad m = 3, \quad n = 3 \;. \tag{8}$$

The random distribution of $C_i^{(1)}$ and $C_i^{(2)}$ will be embedded into the DHNN as a neuron representation, which will be further explained in the next section. The next section will also highlight the implementation of P_{CRAN2SAT} in DHNN. This includes the explanation based on the process involved during the learning phase and retrieval phase of DHNN.

2.2 The Implementation of Conditional Random 2 Satisfiability In DHNN

In this section, the proposed logical rule CRAN2SAT is embedded into the DHNN, producing the model referred to as DHNN-CRAN2SAT. The DHNN-CRAN2SAT model operates in two main phases: the learning phase and the retrieval phase. In the learning phase, the objective is to determine the optimal synaptic weight that reflects the strength of connections between neurons. In the retrieval phase, the final neuron state is produced to evaluate whether it corresponds to a global or local minimum. This state reflects the energy level of the network.

In the learning phase, the logical rule of P_{CRAN2SAT} is embedded into the DHNN, where each literal of P_{CRAN2SAT} corresponds to a neuron state in the network. Synaptic weights are generated using the WA method [5], with the objective of minimizing logical inconsistencies in P_{CRAN2SAT}. The cost function of the DHNN-CRAN2SAT model is denoted as $E_{P_{\text{CRAN2SAT}}}$ and formulated in Eq. (9). This cost function measures the logical inconsistency in the CRAN2SAT formulation by considering both second-order and first-order clauses. The terms m and n represent the number of second-order and first-order clauses, respectively.

$$E_{P_{\text{CRAN2SAT}}} = \sum_{i=1}^{m}\left(\prod_{j=1}^{2} Q_{ij}\right) + \sum_{i=1}^{n}\left(\prod_{j=1}^{1} Q_{ij}\right). \tag{9}$$

The term Q_{ij} defined in Eq. (10) represents the logical inconsistency contributed by an individual literal within a clause and serves as a component of the cost function $E_{P_{\text{CRAN2SAT}}}$. In Eq. (10), S_x denoted the neuron state corresponding to the literal X, where $\neg X$ and X represent negative and positive literals, respectively. Noted that X is composed of arbitrary literals of $\{A_i,\ B_i,\ C_i\}$.

$$Q_{ij} = \begin{cases} \frac{1}{2}(1 - S_x), & \text{if } \neg X \\ \frac{1}{2}(1 + S_x), & \text{if } X \end{cases}. \tag{10}$$

The main objective is to minimize the logical inconsistency of P_{CRAN2SAT} which corresponds to minimizing the cost function $E_{P_{\text{CRAN2SAT}}}$. Minimization occurs when the maximum number of satisfied clauses in P_{CRAN2SAT} is achieved. Election Algorithm is an optimal learning algorithm that guarantees the determination of optimal synaptic weights [13]. Therefore, the Election Algorithm is employed to maximize the number of satisfied clauses in P_{CRAN2SAT} as defined by the fitness function in Eq. (11).

$$f_{P_i} = \sum_{i=1}^{m} C_i^{(2)} + \sum_{i=1}^{n} C_i^{(1)}. \tag{11}$$

where f_{P_i} denotes the fitness of achieving satisfied clauses of P_{CRAN2SAT}. The satisfaction of the second-order and first-order clauses is represented by $C_i^{(2)}$ and $C_i^{(1)}$, respectively as defined in Eqs. (12) and (13):

$$C_i^{(2)} = \begin{cases} 1, & \text{if } E_{P_{\text{CRAN2SAT}}} = 0 \\ 0, & \text{otherwise} \end{cases}. \tag{12}$$

$$C_i^{(1)} = \begin{cases} 1, & \text{if } E_{P_{\text{CRAN2SAT}}} = 0 \\ 0, & \text{otherwise} \end{cases}. \tag{13}$$

Eqs. (12) and (13) show that $E_{P_{\text{CRAN2SAT}}} = 0$ when all clauses in P_{CRAN2SAT} are satisfied. The value of $E_{P_{\text{CRAN2SAT}}}$ will increase with the number of unsatisfied clauses. Therefore, achieving a zero-cost function $\left(E_{P_{\text{CRAN2SAT}}} = 0\right)$ signifies that the logical inconsistency of P_{CRAN2SAT} has been fully minimized and highlighted that all clauses are satisfied. Successfully minimizing the cost function ensures the optimal generation of synaptic weights. The synaptic weights are computed using the WA method [5], which determines the weights by comparing the coefficients of $E_{P_{\text{CRAN2SAT}}}$ with the Lyapunov energy function of the DHNN denoted by $H_{P_{\text{CRAN2SAT}}}$, as formulated in Eq. (14):

$$H_{P_{\text{CRAN2SAT}}} = -\frac{1}{2}\sum_{i=1,i\neq j}^{N}\sum_{j=1,i\neq j}^{N} W_{ij}^{(2)} S_i S_j - \sum_{i=1}^{N} W_i^{(1)} S_i . \tag{14}$$

Based on Eq. (14), $W_{ij}^{(2)}$ and $W_i^{(1)}$ represent synaptic weights for the second-order and first-order, respectively, while S_j refers to the neuron state in P_{CRAN2SAT}. Two key characteristics of synaptic weights in DHNN are symmetrical weights, such that $W_{ij} = W_{ji}$ and there is no self-connection, meaning that $W_{ii} = W_{jj} = 0$. Additionally, the value of $W_{ij}^{(2)}$ and $W_i^{(1)}$ are stored in matrix form as a Content Addressable Memory (CAM), which allows DHNN to memorize the optimal neuron connectivity. This stored information is subsequently used in the computation of the local field.

During the retrieval phase of DHNN-CRAN2SAT, the goal is to produce an optimal final neuron state. To achieve this, the local field (h_i) of DHNN is computed using the values of $W_{ij}^{(2)}$ and $W_i^{(1)}$ stored in CAM, as shown in Eq. (15):

$$h_i = \sum_{j=1,j\neq i}^{N} W_{ij}^{(2)} S_j + W_i^{(1)} . \tag{15}$$

Subsequently, the activation function is applied to squash the value of h_i into a bipolar neuron state, either 1 or -1, as illustrated in Eq. (16). In this context, the final neuron state remains unchanged from the previous state $\left(S_i(t-1)\right)$ if the computation of $g(h_i) = 0$. According to [20], this condition helps maintain state balance in the retrieval process of the network.

$$S_i^f(t) = \begin{cases} 1, & \text{if } g(h_i) > 0 \\ S_i(t-1), & \text{if } g(h_i) = 0 \\ -1, & \text{if } g(h_i) < 0 \end{cases} . \tag{16}$$

Based on Eq. (16), $g(h_i)$ is computed using the Smish activation function as shown in Eq. (17). This function has demonstrated strong capability in retrieving diversified neuron states while effectively addressing the overfitting issue [18].

$$g(h_i) = \alpha h_i \cdot tanh\left[\text{In}\left(1 + sigmoid(\beta h_i)\right)\right]. \tag{17}$$

Next, the retrieved final neuron state $S_i^f(t)$ will be used to evaluate the final energy of the network using Eq. (18). The final energy $H_{P_{\text{CRAN2SAT}}}$ is calculated based on the updated value of $S_i^f(t)$, where S_j represents the neuron state in P_{CRAN2SAT} as shown in Eq. (18):

$$H_{P_{\text{CRAN2SAT}}} = -\frac{1}{2}\sum_{i=1,i\neq j}^{N}\sum_{j=1,i\neq j}^{N} W_{ij}^{(2)} S_i^f S_j^f - \sum_{i=1}^{N} W_i^{(1)} S_i^f . \tag{18}$$

Furthermore, the minimum energy of the network denoted as $H_{P_{\text{CRAN2SAT}}}^{min}$ is computed using Eq. (19).

$$H_{P_{\text{CRAN2SAT}}}^{min} = -\left(\frac{m+2n}{4}\right). \tag{19}$$

According to [21], $H_{P_{\text{CRAN2SAT}}}^{min}$ is the energy value that corresponds to $E_{P_{\text{CRAN2SAT}}} = 0$. Therefore, the DHNN-CRAN2SAT model is said to achieve global minimum energy when its final neuron state converges such that $H_{P_{\text{CRAN2SAT}}} \to H_{P_{\text{CRAN2SAT}}}^{min}$. Eq. (20) is used to evaluate the quality of the retrieved final neuron state. To assess the final neuron state, the tolerance value $Tol = 0.001$ [6] is applied.

$$\left|H_{P_{\text{CRAN2SAT}}} - H_{P_{\text{CRAN2SAT}}}^{min}\right| \leq Tol. \tag{20}$$

The energy value produced by the network is classified as either a global minimum energy or a local minimum energy. If Eq. (20) is satisfied, the final neuron state $S_i^f(t)$ retrieved by the DHNN-CRAN2SAT model is considered a global minima solution. Otherwise, $S_i^f(t)$ is classified as a local minima solution. The general flowchart of the DHNN-CRAN2SAT model proposed in this study, which utilizes Exhaustive Search and the Election Algorithm as learning algorithms and employs the Smish activation function, is presented in Fig. 1. Based on this proposed model, different updating rules are applied to update the final neuron state.

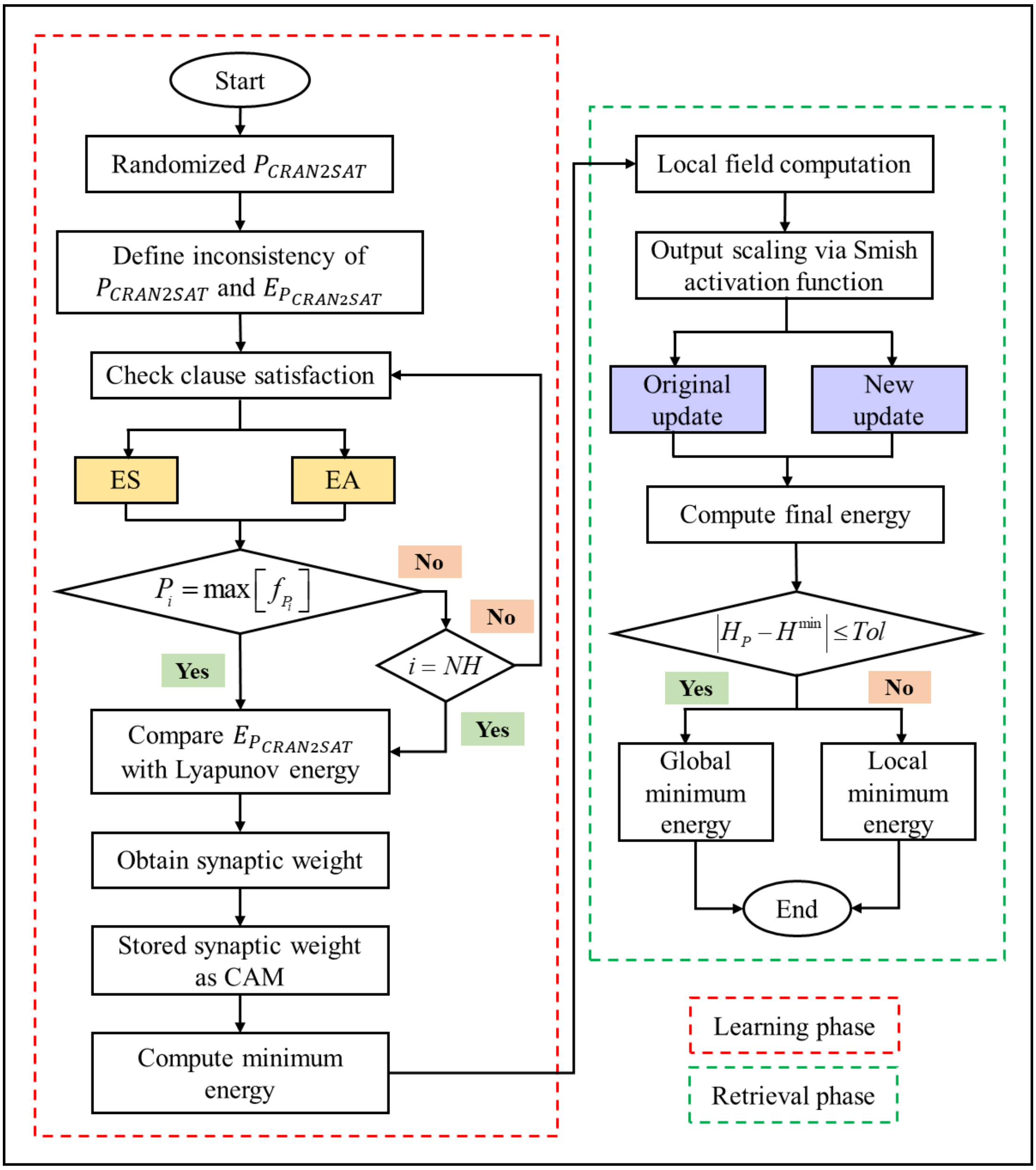

Figure 1. The General Flowchart of DHNN-CRAN2SAT

2.3 Experimental Setup and Implementation

This section presents the experimental setup and describes the performance metrics employed to evaluate the proposed model in terms of global solution and similarity analysis. Global minimum ratio (Z_m) is the ratio between the total number of global minimum solutions and the total number of runs in the retrieval phase. During the retrieval phase, the final neuron state will be analyzed, whether it will converge to the global solution or the local solution, using the formulation of Eq. (20). When the equation is satisfied, the solution is identified as a global solution. If not, it is categorized as a local solution. The formula for calculating the global minimum ratio is given below:

$$Z_m = \frac{1}{NT \cdot C} \sum_{i=1}^{NT \cdot C} NG_{Solution}. \tag{21}$$

The total number of runs $(NT \cdot C)$ is also equivalent to the summation of number of global solutions with the number of local solutions $(NT_{Solution} + NG_{Solution})$. The final neuron states produced by the activation function were analyzed using similarity analysis. The quality of these retrieved states was assessed in terms of total neuron variation, TV. Following the approach of [22], this analysis evaluates how closely the retrieved final neuron state matches the benchmark neuron state. The assessment focuses on individual neuron states. The benchmark neuron state, S_i^{max} commonly known as the ideal neuron state, is defined as:

$$S_i^{max} = \begin{cases} 1, & \text{if } A_i \\ -1, & \text{if } \neg A_i \end{cases}. \tag{22}$$

Based on Eq. (22), the notation of A and $\neg A$ represent the positive and negative literals within the clauses of P_{CRAN2SAT}, respectively. S_i^{max} is defined as the benchmark neuron state initialized in P_{CRAN2SAT}, while S_i^f is the final neuron state retrieved by the DHNN. The similarity index only assesses those S_i^f which reaches a global minimum solution. Eq. (23) defines the Sokal–Sneath (SS) similarity index, which is applied to quantify the similarity between the retrieved neuron states and the benchmark state.

$$SS = \frac{a}{a + 2(b + c)}. \tag{23}$$

The description of the parameter value of a, b, c, d that is associated with the similarity index is highlighted in Table 1.

Table 1. Parameters Used in The Similarity Index

Parameter	Parameter Remarks
a	Number of (S_i^{max}, S_i^f) where $S_i^{max} = 1$ and $S_i^f = 1$
b	Number of (S_i^{max}, S_i^f) where $S_i^{max} = 1$ and $S_i^f = -1$
c	Number of (S_i^{max}, S_i^f) where $S_i^{max} = -1$ and $S_i^f = 1$
d	Number of (S_i^{max}, S_i^f) where $S_i^{max} = -1$ and $S_i^f = -1$

SS evaluates similarity by emphasizing the alignment of positive states, applying a strong penalty to differences between states, and disregarding negative states. Although the Jaccard index also considers similarity based on positive states, the Jaccard index does not impose a strong penalty on differences between states. The range of SS is between $(0,1)$ where a lower SS indicates low similarity between S_i^f and S_i^{max}. Therefore, a low SS value implies that S_i^f demonstrates high diversification with respect to S_i^{max}. The formulation of total neuron variation (TV) is given by Eq. (24):

$$TV = \sum_{i=0}^{\omega} F_i, \tag{24}$$

where

$$F_i = \begin{cases} 1, & S_i^f \neq S_i^{max} \\ 0, & S_i^f = S_i^{max} \end{cases}.$$

Here, ω denotes the total number of solution strings of the final neuron states produced by DHNN-CRAN2SAT that correspond to the global minima solution.

Table 2. Symbols Used in The Experimental Setup

Parameter	Parameter Name
$NT_{Solution}$	Number of total solutions
$NG_{Solution}$	Number of global minima solutions
$H_{P_{\text{CRAN2SAT}}}^{min}$	Minimum energy achieved in P_{CRAN2SAT}
$H_{P_{\text{CRAN2SAT}}}$	Final energy achieved in P_{CRAN2SAT}
S_i^{max}	Benchmark neuron state
S_i^f	Final neuron state
TV	Total neuron variation

It is worth noting that the computation of TV is evaluated based on the final neuron states S_i^f that achieve a global state. The term F_i is used to measure the frequency of distinct neuron states defined by the condition $(S_i^f \neq S_i^{max})$, encountered during the retrieval phase of DHNN. As an example, for each of the solution strings, S_i^f will be compared with S_i^{max} of P_{CRAN2SAT}. Then, if there is a difference between the neuron states, which is even at least for one neuron state, this will contribute to the computational number value for TV. Table 2 summarizes the symbols used during the retrieval phase of the DHNN-CRAN2SAT model.

3. RESULTS AND DISCUSSION

The performance of DHNN-CRAN2SAT is examined in the retrieval phases. The model is developed in a general setting that employs Exhaustive Search and Election Algorithm as learning algorithms. In addition, both models will utilize a different updating rule, namely "Original update" [18], [23] and the "New update" [19], [20] to update the final neuron state. Table 3 until Table 5 summarize the experimental setup and the parameter values applied in this research.

Table 3. Parameter Settings for DHNN-CRAN2SAT [12]

Setting	Parameter Value
Neuron combination (C)	100
Number of trials (NT)	100
Number of learning (NH)	100
Tolerance value (Tol)	0.001
Initialization of neuron states	Random
Activation function	Smish activation function

Table 4. Parameters Used in DHNN-CRAN2SAT (Election Algorithm) [24]

Setting	Parameter Value
Optimization operator	Positive advertisement, Negative advertisement, Coalition
Number of populations	120
Number of parties	4
Positive advertisement rate	0.5
Negative advertisement rate	0.5
Voter attraction strategy	Random
State flip strategy	Random

Table 5. Parameters Used in DHNN-CRAN2SAT (Exhaustive Search) [12]

Setting	Parameter Value
Optimization operator	None (trial and error)

In this section, the discussion is divided into two parts. The first part examines the comparative effectiveness of the Election Algorithm and the Exhaustive Search as learning algorithms. Second, the capability of the new updating rule will be assessed within both learning algorithms. This analysis highlights that improving diversification depends on both the learning algorithm and the updating rule. Since the focus of this paper is on the effectiveness of the updating rule in the DHNN retrieval phase, the discussion emphasizes the effectiveness of DHNN-CRAN2SAT in retrieving maximum global minima, minimizing the similarity index, and maximizing neuron variation.

Fig. 2 shows the Z_m values for Exhaustive Search under both updating rules. At lower NN values (3 to 9), both updating rules achieve a maximum Z_m value of 1, indicating successful retrieval of maximum global minimum solutions. However, as NN increases, Z_m decreases for both the original and new updating rules, demonstrating that DHNN-CRAN2SAT struggles to retrieve global solutions at higher NN. The primary reason Z_m approaching zero beyond $NN = 33$ is that Exhaustive Search tends to generate suboptimal synaptic weights at higher value of NN. As NN increases, Exhaustive Search fails to effectively minimize the cost function, resulting in suboptimal weights. This consequently leads the neuron state to converge to a suboptimal local solution [18]. Although the new updating rule slightly improves Z_m compared to the original rule in the mid-range value of NN, the overall ability of the DHNN-CRAN2SAT model to retrieve maximum global minima is primarily influenced by the choice of learning algorithm. Obtaining a global solution is essential because, as noted in [12], only a final neuron state that reaches a global solution can be assessed in terms of diversification and similarity index. Accordingly, implementing the Election Algorithm in the learning phase strengthens the ability of DHNN-CRAN2SAT to retrieve such solutions.

The Z_m values for the Election Algorithm, which consistently remain at the maximum value of 1 across all NN values for both updating rules are also presented in Fig. 2. This result confirms the capability of the Election Algorithm to consistently retrieve global minimum solutions across different values of NN and updating rules. These findings emphasize that the Election Algorithm, as a learning algorithm, consistently retrieves global minimum solutions across all NN, whereas the performance of the Exhaustive Search decreases as NN increases. By employing the Election Algorithm as the learning algorithm, DHNN-CRAN2SAT achieves consistent global minimum retrievals up to $NN = 90$. This illustrates the capability of the Election Algorithm in minimizing the cost function. The results show that the Election Algorithm significantly improves the learning phase by obtaining a satisfactory interpretation with fewer iterations [13]. This is due to the effectiveness of the Election Algorithm, which has a balance of exploration and exploitation, which leads to a $E_{P_{CRAN2SAT}} = 0$ representing that every clause in P_{CRAN2SAT} are satisfied.

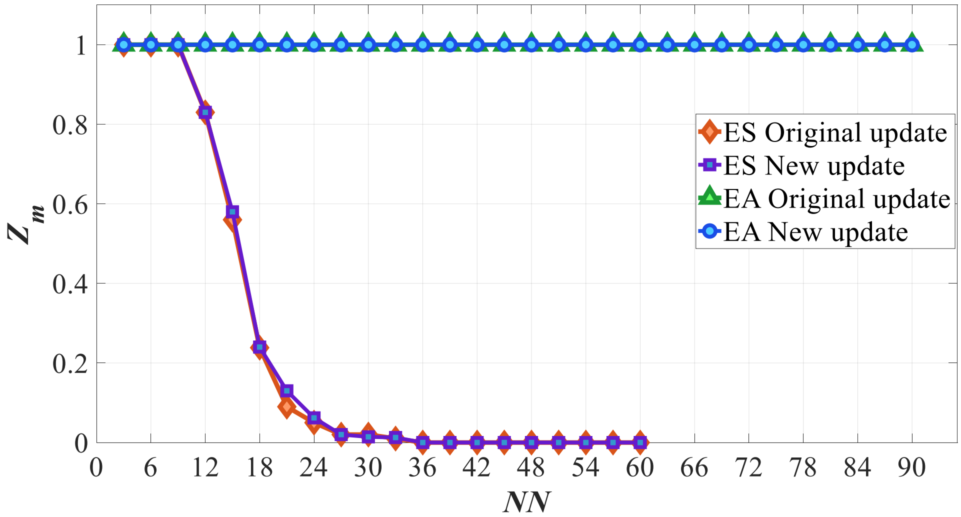

Figure 2. Z_m for DHNN-CRAN2SAT

The optimal learning phase allows for the storage of correct synaptic weights in the CAM and thus contributes to the retrieval of global minimum solutions. Therefore, the proposed model achieves the highest number of global minimum solutions. Thus, DHNN-CRAN2SAT demonstrates the ability to attain the optimal number of global minimum solutions. These findings highlight that achieving global minimum solutions primarily depends on the choice of learning algorithm, because an optimal learning algorithm can generate synaptic weights that ensure the retrieval of an optimal final neuron state. Although the Election Algorithm maintains strong performance, the updating rule has a greater influence on increasing the diversity of retrieved neuron states. This diversification is further evaluated in terms of TV and SS similarity index.

Fig. 3 illustrates the SS values for DHNN-CRAN2SAT under different learning algorithms and updating rules. The results clearly demonstrate that the new updating rule enhances the diversity of the final neuron states, as reflected by consistently lower SS value across all NN with an average SS value of 0.3809. This trend is evident for both the Exhaustive Search and Election Algorithm learning algorithms, with the "ES New update" and "EA New update" lines remaining below their respective original update lines throughout the evaluated range of NN. The SS index, which measures similarity by considering both positive and negative neuron states, is particularly effective for detecting overfitting in DHNN. High SS values correspond to repetitive or uniform final neuron states, while lower values indicate greater state diversity and a reduced tendency toward overfitting. The conventional updating rule consistently produces higher SS values, with an average of 0.4119, particularly when the local field equals zero. This condition often causes the network to converge on repetitive states.

The new updating rule addresses this limitation by preserving the previous neuron state when the local field equals zero. This mechanism prevents the network from becoming trapped in a limited set of outcomes and promotes greater variation in the retrieval phase. As a result, the final neuron states become more diverse, which is crucial for ensuring robust performance. Additionally, the figures show that Exhaustive Search can only evaluate SS values up to $NN = 33$, as it tends to get stuck in local solutions. In contrast, the Election

Algorithm supports evaluation up to $NN = 90$ and consistently achieves lower SS values when using the new updating rule. This indicates that the Election Algorithm is more capable of finding global solutions, which can then be further analyzed for the quality of the final neuron state.

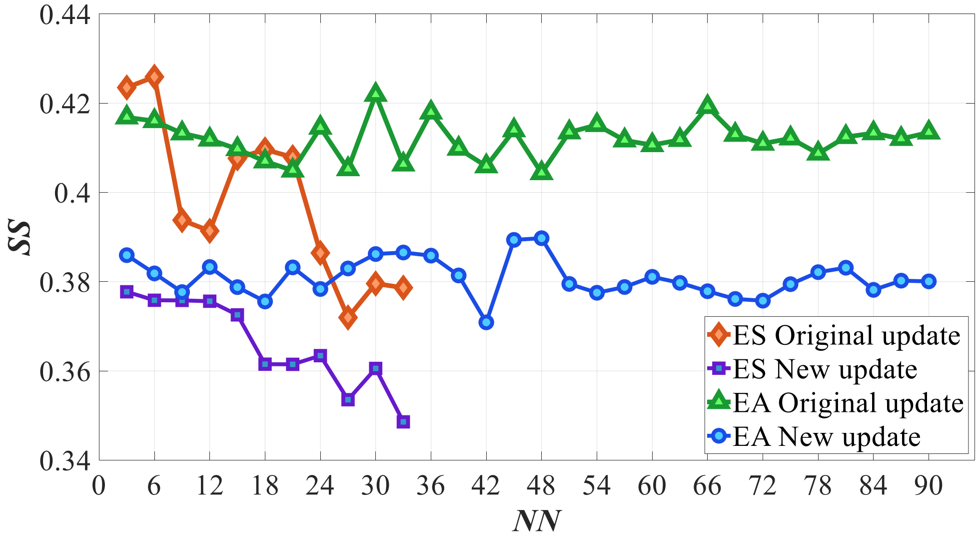

Figure 3. SS **for DHNN-CRAN2SAT**

In summary, across all evaluated NN, the new updating rule produces the lowest SS values for both Exhaustive Search and Election Algorithm, confirming significant diversification of the final neuron states. This improvement is achieved by ensuring that when the local field is zero, the neuron state follows the previous state, which aligns with the logical structure of CRAN2SAT and further reduces similarity in both positive and negative states. While the Smish activation function helps address the vanishing gradient problem [18], it is the implementation of the new updating rule during the retrieval phase that most effectively minimizes the Sokal value and promotes diversity.

As shown in Fig. 4, DHNN-CRAN2SAT shows neuron variation when applying different learning algorithms and updating rules in the retrieval phase. This variation reflects both the quality and diversity of the final neuron states. An increased TV implies that more solutions differ from the benchmark states, resulting in reduced overfitting and improved model performance. The limitations of Exhaustive Search become evident when analyzing its performance in evaluating neuron variation. As shown in Fig. 4, Exhaustive Search is ineffective beyond $NN = 33$ where it frequently becomes trapped in local minima, thus resulting in a TV value of zero. Although Exhaustive Search can only compute TV up to $NN = 33$, the results show that the "ES New Update" outperforms the "ES Original Update" in terms of neuron variation.

The updating rule used during the retrieval phase significantly influences neuron variation. The original updating rule results in significantly lower TV. This is due to its tendency to repeatedly retrieve the same positive neuron state when facing a zero local field. Such repetition increases the risk of overfitting and limits the ability of the network to explore different states. The introduction of the new updating rule significantly improves neuron variation, as reflected by the higher TV values shown in Fig. 4. This new updating rule changes how the network behaves when a zero local field is encountered by maintaining the current neuron state instead of forcing a fixed update. As a result, the network can explore more neuron states and retrieve a wider range of solutions. Additionally, the flexible structure of CRAN2SAT contributes significantly to this improvement. The flexible logical structure of CRAN2SAT expands the possible neuron state combinations, leading to global minimum solutions, enabling the network to obtain solutions that diverge considerably from the benchmark state. As a result, the overall variation among the retrieved neuron states will increase.

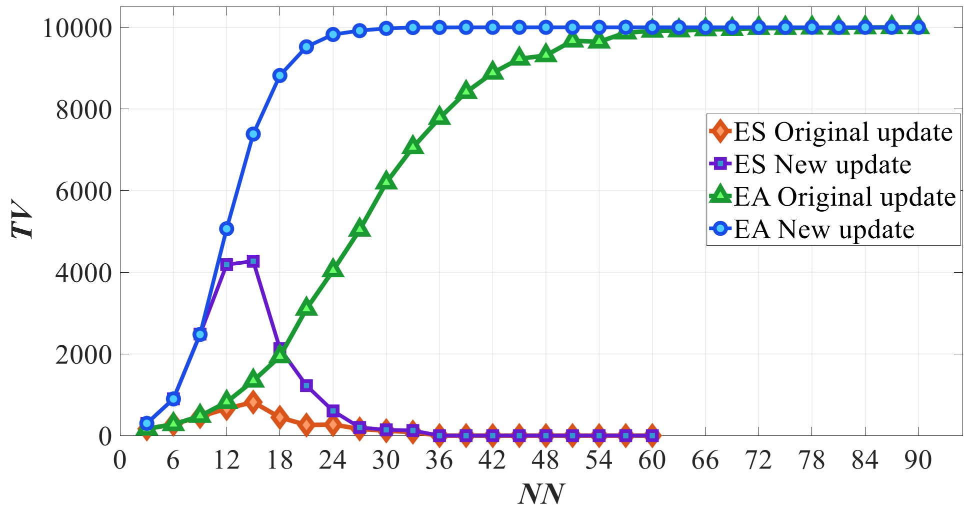

Figure 4. ***TV*** **for DHNN-CRAN2SAT**

The Friedman Test was conducted to statistically analyze the significance of the proposed updating rule in retrieving diversified final neuron states compared to the original updating rule. Table 6 presents the average rank results and corresponding p-values for the performance metrics Z_m, SS and TV. Since the obtained p-values are less than 0.05, the results show statistically significant differences between the new and original updating rules. Based on the average rank results, the new updating rule achieves higher ranks for Z_m and TV under both ES and EA learning algorithms. This highlight improved the diversification of the neuron states. In addition, the new updating rule reports lower average rank values for SS, demonstrating improved similarity of the retrieved neuron states. These findings demonstrate that the proposed updating rule significantly enhances the diversification of the retrieved final neuron states compared to the original updating rule.

Table 6. Detailed Findings of Average Rank for the Friedman Test

Model	**Performance Metrics**		
	Z_m	SS	TV
ES Original Update	1.55	3.09	1.28
ES New Update	$\mathbf{1.75}$	$\mathbf{1.00}$	$\mathbf{2.20}$
EA Original Update	3.35	3.64	2.65
EA New Update	$\mathbf{3.35}$	$\mathbf{2.27}$	$\mathbf{3.88}$
p-value	1.192×10^{-10}	9.454×10^{-6}	1.330×10^{-9}

The computational complexity of the proposed work is primarily influenced by the number of initialized solution strings, which is determined by the number of neurons in the network. The EA employs a balanced exploration and exploitation strategy, which effectively enhances cost function minimization and leads to an optimal learning phase. In contrast, ES relies on a trial-and-error mechanism that often results in suboptimal synaptic weights. Consequently, the solutions obtained by ES tend to converge to local optima and are limited in terms of diversification and similarity evaluation. During the retrieval phase, the utilization of the new updating rule reduces overfitting behavior in the retrieved final neuron states, which improves solution diversification and enhances the overall quality of the retrieved neuron states. The computational complexity is analyzed with respect to the neuron initialization (N), the new updating rules (U_{New}) and the original updating rule (U_{Ori}) as summarized in Table 7.

Table 7. The Computational Complexity of Each Algorithm

Model	**Computational Complexity**
ES Original Update	$O(2^N + U_{Ori})$
ES New Update	$O(2^N + U_{New})$
EA Original Update	$O(N^2 + U_{Ori})$
EA New Update	$(N^2 + U_{New})$

4. CONCLUSION

This study demonstrates that the DHNN-CRAN2SAT model can produce both optimal and diverse neuron states. This study also highlights the importance of the learning algorithm and the updating rule in producing these outcomes. The results indicate that the Election Algorithm plays a key role in guiding the network toward global solutions. The superior performance of the Election Algorithm over the Exhaustive Search across all NN is mainly attributed to its balanced exploration and exploitation strategy, which enhances cost function minimization. Furthermore, the newly proposed updating rule enhances the diversity across the final neuron states. This is supported by the lower SS similarity index and the increase in TV. With the original updating rule, the network often overfits when the local field value is zero. The new updating rule avoids this issue by keeping the neuron state unchanged under those conditions. As a result, the new updating rule allows the model to search for a wider range of possible solutions. Additionally, the flexible logical structure of CRAN2SAT also facilitates this process by enabling a wide range of neuron state combinations that correspond to global solutions. In conclusion, the Election Algorithm enhances the learning phase of DHNN-CRAN2SAT, which subsequently improves the effectiveness of the retrieval phase. However, the quality of the retrieved state is further improved by the new updating rule. Diversification in the final neuron states plays a crucial role in logic mining as it contributes to generating optimal induced logic. The proposed model can be applied within the context of logic mining frameworks and further applied to real-world classification tasks.

Author Contributions

Nurshazneem Roslan: Writing – Original Draft, Conceptualization, Methodology, Validation, Investigation, Visualization, Formal Analysis. Saratha Sathasivam: Conceptualization, Validation, Writing – Review and Editing, Supervision. The results were discussed by all authors, who also contributed to the final version of the manuscript.

Funding Statement

This research was financially supported by Universiti Sains Malaysia through the Bridging Grant Scheme (R501-LR-RND003-0000002089-0000).

Acknowledgment

The authors sincerely thank the editor and reviewers for their valuable suggestions, which have helped improve the quality of this manuscript. The author also gratefully acknowledges the financial support provided by Universiti Sains Malaysia through the Bridging Grant Scheme (R501-LR-RND003-0000002089-0000).

Declarations

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of Generative AI and AI-Assisted Technologies

Generative AI tools (e.g., ChatGPT) were used solely for language refinement, including grammar, spelling, and clarity. The scientific content, analysis, interpretation, and conclusions were developed entirely by the authors. All final text was reviewed and approved by the authors.

REFERENCES

[1] J. J. Hopfield and D. W. Tank, "'NEURAL' COMPUTATION OF DECISIONS IN OPTIMIZATION PROBLEMS," *Biol Cybern*, vol. 52, no. 3, pp. 141–152, 1985. doi: https://doi.org/10.1007/BF00339943

[2] Z. T. Njitacke, S. D. Isaac, T. Nestor, and J. Kengne, "WINDOW OF MULTISTABILITY AND ITS CONTROL IN A SIMPLE 3D HOPFIELD NEURAL NETWORK: APPLICATION TO BIOMEDICAL IMAGE ENCRYPTION," *Neural Comput Appl*, vol. 33, no. 12, pp. 6733–6752, 2021. doi: https://doi.org/10.1007/s00521-020-05451-z

[3] N. Soni, E. K. Sharma, and A. Kapoor, "APPLICATION OF HOPFIELD NEURAL NETWORK FOR FACIAL IMAGE RECOGNITION," *International Journal of Recent Technology and Engineering*, vol. 8, no. 1, pp. 3101–3105, 2019.

[4] Z. Fahimi, M. R. Mahmoodi, H. Nili, V. Polishchuk, and D. B. Strukov, "COMBINATORIAL OPTIMIZATION BY WEIGHT ANNEALING IN MEMRISTIVE HOPFIELD NETWORKS," *Sci Rep*, vol. 11, no. 1, 2021. doi: https://doi.org/10.1038/s41598-020-78944-5

[5] W. A. T. W. Abdullah, "LOGIC PROGRAMMING ON A NEURAL NETWORK," *International Journal of Intelligent Systems*, vol. 7, no. 6, pp. 513–519, 1992. doi: https://doi.org/10.1002/int.4550070604

[6] S. Sathasivam, "UPGRADING LOGIC PROGRAMMING IN HOPFIELD NETWORK," *Sains Malays*, vol. 39, no. 1, pp. 115–118, 2010.

[7] 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, no. 1, pp. 139–152, 2017.

[8] M. A. Mansor, M. S. M. Kasihmuddin, and S. Sathasivam, "ARTIFICIAL IMMUNE SYSTEM PARADIGM IN THE HOPFIELD NETWORK FOR 3-SATISFIABILITY PROBLEM," *Pertanika J Sci Technol*, vol. 25, no. 4, pp. 1173–1188, 2017. doi: https://doi.org/10.9781/ijimai.2017.448

[9] 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 K ≤ 2 IN HOPFIELD NEURAL NETWORK," *Sains Malays*, vol. 49, no. 11, pp. 2847–2857, Nov. 2020. doi: https://doi.org/10.17576/jsm-2020-4911-23

[10] S. A. Karim *et al.*, "RANDOM SATISFIABILITY: A HIGHER-ORDER LOGICAL APPROACH IN DISCRETE HOPFIELD NEURAL NETWORK," *IEEE Access*, vol. 9, pp. 50831–50845, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3068998

[11] A. Alway, N. E. Zamri, S. A. Karim, M. A. Mansor, M. S. Mohd Kasihmuddin, and M. Mohammed Bazuhair, "MAJOR 2 SATISFIABILITY LOGIC IN DISCRETE HOPFIELD NEURAL NETWORK," *Int J Comput Math*, vol. 99, no. 5, pp. 924–948, 2022. doi: https://doi.org/10.1080/00207160.2021.1939870

[12] N. Roslan and S. Sathasivam, "ANALYZING PERFORMANCE OF ACTIVATION FUNCTIONS IN LOGIC SATISFIABILITY HOPFIELD NEURAL NETWORK," in *AIP Conference Proceedings*, American Institute of Physics, Aug. 2024. doi: https://doi.org/10.1063/5.0223811

[13] N. 'Afifah Rusdi *et al.*, "SYNERGIZING INTELLIGENCE AND KNOWLEDGE DISCOVERY: HYBRID BLACK HOLE ALGORITHM FOR OPTIMIZING DISCRETE HOPFIELD NEURAL NETWORK WITH NEGATIVE BASED SYSTEMATIC SATISFIABILITY," *AIMS Mathematics*, vol. 9, no. 11, pp. 29820–29882, 2024. doi: https://doi.org/10.3934/math.20241444

[14] M. A. Mansor and S. Sathasivam, "ACCELERATING ACTIVATION FUNCTION FOR 3-SATISFIABILITY LOGIC PROGRAMMING," *International Journal of Intelligent Systems and Applications*, vol. 8, no. 10, pp. 44–50, Oct. 2016. doi: https://doi.org/10.5815/ijisa.2016.10.05

[15] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "ACTIVATION FUNCTIONS IN DEEP LEARNING: A COMPREHENSIVE SURVEY AND BENCHMARK," Sep. 07, 2022, *Elsevier B.V.* doi: https://doi.org/10.1016/j.neucom.2022.06.111

[16] T. Szandała, "REVIEW AND COMPARISON OF COMMONLY USED ACTIVATION FUNCTIONS FOR DEEP NEURAL NETWORKS," in *Studies in Computational Intelligence*, vol. 903, 2021. doi: https://doi.org/10.1007/978-981-15-5495-7_11

[17] X. Wang, H. Ren, and A. Wang, "SMISH: A NOVEL ACTIVATION FUNCTION FOR DEEP LEARNING METHODS," *Electronics (Switzerland)*, vol. 11, no. 4, p. 540, Feb. 2022. doi: https://doi.org/10.3390/electronics11040540

[18] N. Roslan, S. Sathasivam, and F. L. Azizan, "CONDITIONAL RANDOM K SATISFIABILITY MODELING FOR K =1,2 (CRAN2SAT) WITH NON-MONOTONIC SMISH ACTIVATION FUNCTION IN DISCRETE HOPFIELD NEURAL NETWORK," *AIMS Mathematics*, vol. 9, no. 2, pp. 3911–3956, 2024. doi: https://doi.org/10.3934/math.2024193

[19] Y. Guo *et al.*, "DUAL OPTIMIZATION APPROACH IN DISCRETE HOPFIELD NEURAL NETWORK," *Appl Soft Comput*, vol. 164, Oct. 2024. doi: https://doi.org/10.1016/j.asoc.2024.111929

[20] B. Yang *et al.*, "BRAN2SAT: REDUNDANT SATISFIABILITY LOGIC IN LYAPUNOV-BASED DISCRETE HOPFIELD NEURAL NETWORK," *J Comput Des Eng*, vol. 12, no. 4, pp. 185–204, Apr. 2025. doi: https://doi.org/10.1093/jcde/qwaf039

[21] N. E. Zamri, S. A. Azhar, M. A. Mansor, A. Alway, and M. S. M. Kasihmuddin, "WEIGHTED RANDOM K SATISFIABILITY FOR K=1,2 (R2SAT) IN DISCRETE HOPFIELD NEURAL NETWORK," *Appl Soft Comput*, vol. 126, p. 109312, Sep. 2022. doi: https://doi.org/10.1016/j.asoc.2022.109312

[22] M. S. M. Kasihmuddin, M. A. Mansor, M. F. M. Basir, and S. Sathasivam, "DISCRETE MUTATION HOPFIELD NEURAL NETWORK IN PROPOSITIONAL SATISFIABILITY," *Mathematics*, vol. 7, no. 11, p. 1133, Nov. 2019. doi: https://doi.org/10.3390/math7111133

[23] F. L. Azizan, S. Sathasivam, M. K. M. Ali, N. Roslan, and C. Feng, "HYBRIDISED NETWORK OF FUZZY LOGIC AND A GENETIC ALGORITHM IN SOLVING 3-SATISFIABILITY HOPFIELD NEURAL NETWORKS," *Axioms*, vol. 12, no. 3, p. 250, Mar. 2023. doi: https://doi.org/10.3390/axioms12030250

[24] S. Sathasivam, M. A. Mansor, M. S. M. Kasihmuddin, and H. Abubakar, "ELECTION ALGORITHM FOR RANDOM K SATISFIABILITY IN THE HOPFIELD NEURAL NETWORK," *Processes*, vol. 8, no. 5, p. 568, May 2020. doi: https://doi.org/10.3390/pr8050568