MATHEMATICAL MODELING AND COMPUTING, Vol. 12, No. 2, pp. 468–480 (2025)

Weighted random 3-satisfiability in discrete Hopfield neural network

Liu X.[1,2], Chang Y.[2,3], Jiang X.[2]

[1] *School of General Education, Guangzhou College of Technology and Business, 510850 Guangzhou, China*
[2] *School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800 USM, Malaysia*
[3] *School of Computer Science and Engineering, Hunan Institute of Technology, 421002 Hengyang, China*

(Received 18 December 2024; Revised 13 May 2025; Accepted 19 May 2025)

Satisfiability (SAT) is remarkable in the field of computational mathematics because it can be utilized to represent the information of any categories of any datasets. Recent research about this paradigm has tended to model Discrete Hopfield Neural Network (DHNN) via SAT. Despite the widespread implementations of SAT in DHNN, there are limitations to the control of the distribution of negative and positive literals in the logical rule and this aspect has been rarely discussed. In this paper, a novel logic rule named weighted 3 satisfiability is proposed by implementing ratio to the negative literals in the satisfiability clauses. The proposed weighted 3 satisfiability was implemented into DHNN where the cost function was derived by minimizing the inconsistency of the logic. The effectiveness of the novel weighted 3 satisfiability was analyzed by various metrics which was settled before. Errors in each phase reduced when the ratio is higher than 0.2. When the ratio within a low value $r = 0.1$, the similarity index shows a high level nearly with RAN3SAT. Based on the results of the metrics, the proposed logic has outperformed most of the existing logic models and it also has a more positive impact ability to gain the global minimum solutions with special ratio of negative literals.

Keywords: *weighted; satisfiability; discrete Hopfield neural network; systematic.*
2010 MSC: 68T05, 68Q25, 62M20 **DOI:** 10.23939/mmc2025.02.468

1. Introduction

In decades, researchers have focused on the Artificial Intelligence (AI) because of the simulation of human intelligence. AI is a broad field within computer science aimed at enabling machines to perform tasks that typically require human intelligence [1]. Artificial Neural Network (ANN) is a computational model that simulates the structure of biological neural networks to handle nonlinear problems and learn complex data patterns [2]. This network is important in machine learning. ANN consists of multiple layers, including an input layer, hidden layers, and an output layer. Each layer is composed of neurons that are connected by weights [3]. ANNs are widely used in tasks such as image recognition [4], natural language processing [5], and predictive modeling [6]. Hopfield and Tank (1985) introduced a model first added ANN into the network of biological or microelectronic neurons [7]. This network could offer computational capabilities for a broad range of combinatorically complex problems. The inherent power and speed of these networks may enhance the efficiency of biological information processing. This made the Hopfield Neural Network (HNN) one of the earliest ANNs which consists of interconnected neurons with input and output but no hidden neurons. Each neuron in HNN is connected with synaptic weight. In the context of HNN, Content Addressable Memory (CAM) plays a crucial role in storing and retrieving patterns [8]. CAM allows the network to retrieve stored patterns based on partial or noisy input, making it particularly useful for associative memory tasks. When a pattern is presented to the network, CAM helps in locating and retrieving the most similar stored pattern, even if the input is incomplete or corrupted. Joya et al. [9] later improved a HNN for optimization problems. This work focused on the incoherence between the network dynamics and the energy function. The model indicates that convergence depends on the coefficients weighting the cost function terms. Discrete and Continuous Hopfield Neural Networks have been used earlier to solve problems separately [10]. Discrete Hopfield Neural Network (DHNN) are effective at retrieving stored patterns even from noisy or incomplete inputs [11], making them useful for associative memory tasks because of the usage of CAM. DHNN can also reduce computational complexity compared to continuous-valued networks by operating with discrete states [12]. In the work of this article, the scope of the discussion in this paper is within the domain of DHNN.

One of the first attempts to model neurons using symbolic logic was introduced by Abdullah [13]. In this approach, DHNN was framed as a computational paradigm and symbolic rule system, rather than a method for addressing optimization problems. Boolean satisfiability (SAT) is known as the task of determining whether there exists an assignment satisfying a given Boolean formula [14]. Boolean formula could lead to an operation with outcome whether is Ture or False. To solve the formula is a fundamental intractable problem in computer science. Logic can be used as a symbolic rule to prove the goal of network. This introduction of logic lead to the Wan Abdullah method to find the optimal synaptic weight. Each clause in the network was combined with conjunctive normal formula (CNF). Each literal in one clause was combined with disjunctive normal formula (DNF). The satisfiability of the formula depends on the Boolean variables. Logic rules can be separated into two domains which are namely systematic and non-systematic [15]. Kasihmuddin et al. [16] first proposed the systematic logic rule named 2-Satisfiability (2SAT) by creating a cost function that capitalizes the symmetric neuron connection. Due to the use of Estimated Distribution Algorithm during the retrieval phase, the proposed DHNN achieved a high rate of global minimum solution. Soon after, Mansor et al. [17] has proved that DHNN would reach the highest capacity when the structure of logical function refers to 3-Satisfiability (3SAT). In this specific systematic logical rule, each clause has strictly 2 literals in 2SAT logic and 3 literals in 3SAT logic. Each literal is joined by conjunctions in each clause and all the clauses are connected by disjunctive. This work also proposed that when the number of literals in each clause is bigger than 3, it can be reduced to a combination of clauses with 3 literals or lower number of literals due to the reduction theorem. More flexible logic rule implemented in DHNN can help networks solving complex problems widely.

Different from the strict number of literals in each clause in systematic logic rules, non-systematic logic has a more flexible structure with different number of literals in each clause. Maneva et al. [18] proposed a structure of satisfying assignments of random 3SAT formula which consists of random instances. Since then, the non-systematic satisfiability has become more popular to discuss. Sathasivam et al. later introduced a flexible logic rule named Random 2 Satisfiability (RAN2SAT) in DHNN [19]. This novel logic rule was proposed by conducting the random structure that include first-order and second-order clauses. This work has been improved later by extending the higher order logic into DHNN. Karim et al. [20] proposed various combinations of random 3-Satisfiability (RAN3SAT). Each of these combinations were discussed the effectiveness in DHNN. RAN3SAT has high variation value, high global minimum ratio of solution and low errors. This logic rule was considered one of the most effective rules implemented in DHNN. Not soon later, Guo et al. [21] has first attempt to wield both systematic logic rule 2SAT and non-systematic logical rule RAN2SAT into DHNN. Furthermore, Gao et al. [22] proposed a hybrid logic rule named G-Type Random 3-Satisfiability (GRAN3SAT) which capitalized both higher-order systematic and non-systematic logical rules in DHNN. This logic structure has large storage capacity and can be explored to high dimensional problems. Zamri et al. [23] concluded the importance of the ratio of negative literals. This work proposed a logic by inserting weight to indicate the distribution of negative literals. Despite the rapid development of the logic in DHNN, less study has mentioned the importance of logic with higher order clauses which are classified with different ratio of negative literals.

Higher-order clauses can more effectively represent and solve complex problems by combining multiple lower-order facts and relationships into a single, more comprehensive statement [24]. They provide greater flexibility in modeling and reasoning, enabling more sophisticated logical constructs and accommodating a wider range of scenarios. In this study of research, we proposed a systematic satisfiability with higher order clauses by adding the ratio of negative literals in the logic. This work has not been explored previously. This research focus on the role of ratio which can show the distribution of negative literals under the condition of strictly 3 literals per clause. By using the higher order clauses, the capacity of information is larger. Due to the presenting novel findings in this previously unexplored area, we provide valuable insights that could significantly enhance the effectiveness and accuracy of future learning and retrieval processes. Our work offers a superior solution to improve performance in Discrete Hopfield Neural Network. The proposed model adding the ratio of negative literals is implemented in DHNN. Thus, the contributions of this paper are stated as follows:

(a) To formulate a novel systematic logic rule named weighted 3-satisfiability (w3SAT) by adding the ratio of negative literals into the logical formula. In this context, this logic involves only the third-order clauses with different ratio of negative literals.

(b) To implement the proposed logic into Discrete Hopfield Neural Network by finding the minimum cost function. In this approach, neurons are represented as literals in the logic rule. The synaptic weight of the neurons can be found by comparing the cost function with the Lyapunov energy function.

(c) To evaluate the performance of the proposed network by simulating datasets. This hybrid network consisted of weighted 3-satisfiability and Discrete Hopfield Neural Network will be evaluated by various metrics to show how this logic structure performs better in the learning and retrieval phase.

(d) To compare the performance of the novel proposed w3SAT with the existing non-systematic logic rule RAN3SAT in the learning and retrieval phase. The similarity index is utilized to evaluate the similarity between the benchmark and final neuron state.

The organization of this paper is shown as below: The proposed formula of w3SAT is preliminary introduced in Section 2. The process of implementing w3SAT in DHNN should be described in Section 3. In Section 4, experimental settings and the evaluation metrics are presented. The results and the compression of different models are discussed in Section 5. The conclusion is finally mentioned in Section 6.

2. Weighted 3-satisfiability

Weighted 3 satisfiability which is mentioned in this paper is a systematic logic structure presented in conjunctive normal form (CNF) where each clause is strictly limited to three literals. Weighted 3 satisfiability is based on 3SAT by adding weight as the ratio of negative literals. Each literal in this context has the binary value TRUE or FALSE, denoted as 1 or -1. The standard formulation of weighted 3-satisfiability is settled as follows:

(a) A set of n variables: $A_1, A_2, \ldots, A_n$, where $A_i \in \{-1, 1\}$ has the bipolar values, $i = 1, 2, \ldots, n$. Each literal has an equal probability of being positive or negative.

(b) A set of m clauses, denoted as $C_1, C_2, \ldots, C_m$, where m must satisfy the equation as below:

$$3m = n. \tag{1}$$

(c) The general formula of the logic structure of w3SAT is settled as:

$$G_{w3SAT} = \wedge_{i=1}^{m} C_i^{(3)}. \tag{2}$$

(d) In each clause, $C_i^{(3)}$ is given as below:

$$C_i^{(3)} = A_1 \vee A_2 \vee A_3. \tag{3}$$

The literals are randomly established whether are negative or positive. Each variable represents in the clause demonstrate either TRUE or FALSE where $A_i \in \{A_i, \neg A_i\}$. The number of literals in each clause in the proposed logical rule is strictly equal to 3.

In w3SAT, the ratio of negative literals is r. The range of the ratio should be proposed between 0.1 and 0.9. The step size is set to $\triangle\ r = 0.1$. The cases $r = 0$ and $r = 1$ have not been mentioned in this article. This logic contains no negative literals when $r = 0$ and each literal are all negative when $r = 1$. The number of negative literals can be calculated as rn [23]. Since rn is not always a natural number, some technical processing is needed to round the value. Rounding function is used to deal with the problem in this article. For example, when the ratio $r = 0.4$ and the number of literals is set up to $n = 12$, the value $rn = 0.4 \cdot 12 = 4.8$, then the number of negative literals should be 4. Possible examples of G_{w3SAT} under this condition can be given by equations (4)–(6) as shown below:

$$G_{w3SAT} = (A_1 \vee A_2 \vee \neg A_3) \wedge (\neg B_1 \vee \neg B_2 \vee \neg B_3)\,, \tag{4}$$

$$G_{w3SAT} = (\neg A_1 \vee A_2 \vee \neg A_3) \wedge (B_1 \vee \neg B_2 \vee \neg B_3)\,, \tag{5}$$

$$G_{w3SAT} = (\neg A_1 \vee \neg A_2 \vee \neg A_3) \wedge (\neg B_1 \vee B_2 \vee B_3)\,. \tag{6}$$

Although the number of negative literals is strictly equal to 4, the position where the negative literals are is separated randomly. The logic structure of w3SAT is depends mainly on the negative ratio and the total number of neurons. The weight for any $C_i^{(3)}$ is defined as the following in equation (7):

$$w_i^{(3)} = \begin{cases} 0, & \text{if} \quad (A_1 \vee A_2 \vee A_3)\,, \\ 1, & \text{if} \quad (\neg A_1 \vee A_2 \vee A_3)\,, (A_1 \vee \neg A_2 \vee A_3)\,, (A_1 \vee A_2 \vee \neg A_3)\,, \\ 2, & \text{if} \quad (\neg A_1 \vee \neg A_2 \vee A_3)\,, (A_1 \vee \neg A_2 \vee \neg A_3)\,, (\neg A_1 \vee A_2 \vee \neg A_3)\,, \\ 3, & \text{if} \quad (\neg A_1 \vee \neg A_2 \vee \neg A_3)\,. \end{cases} \tag{7}$$

Where $w_i^{(3)}$ indicates the number of negative literals in the third-order clauses. The value of $w_i^{(3)}$ is equal to zero means the number of negative literals is zero in each clause. The total number of negative literals should be calculated as in equation (8):

$$\alpha = \sum_{i=1}^{m} w_i^{(3)}. \tag{8}$$

Figure 1 shows the details of generating w3SAT. The most important initial condition is to ensure that equation (9) is satisfied,

$$|rn - \alpha| = 0. \tag{9}$$

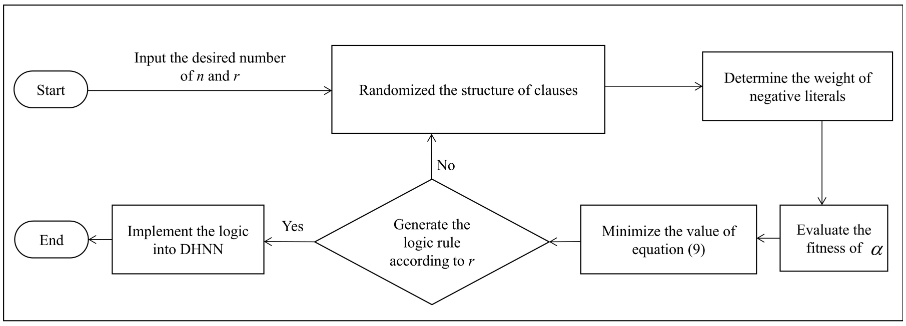

Fig. 1. Flow chart of generating w3SAT.

3. Weighted 3-satisfiability in discrete Hopfield neural network

Discrete Hopfield Neural Network (DHNN) is a recurrent variant of ANNs without a hidden layer. DHNN is valuable for applications requiring robust pattern recognition, optimization, and error correction due to their ability to handle discrete states and perform associative memory tasks effectively. DHNN is employed to recognize and retrieve patterns from noisy or incomplete data by leveraging their associative memory capabilities. The states of DHNN consist of bipolar values which are represented by $-1, 1$. The general neuron updating rule in DHNN is given by:

$$S_i = \begin{cases} 1, & \sum_{j,k}^{n} W_{ijk} S_j S_k \geqslant \delta_i, \\ -1, & \text{otherwise.} \end{cases} \tag{10}$$

Hence, S_i denotes the state of neuron i after update. W_{ijk} denotes the synaptic weight that refers to the strength or efficacy of the connection between neuron i, j and k. δ_i is the threshold value of neuron i. There are two distinctive features about the synaptic weight. First, the synaptic weight has no self-connection. Therefore, $W_{ii} = W_{jj} = W_{kk} = 0$ and $W_{iii} = W_{jjj} = W_{kkk} = 0$. Second, synaptic weight plays a symmetrical role in DHNN. The weight satisfies the condition $W_{ij} = W_{ji}$ and $W_{ijk} = W_{jik} = W_{kji}$. The predetermined threshold $\delta_i = 0$ is always used to make the network's behavior more straightforward and predictable. In this case, w3SAT only considers 3 neurons in each clause. The primary goal of implementing G_{w3SAT} into DHNN is to minimize the cost function. As a result, the network can achieve the lowest possible energy state during the training phase. Overall, the cost function can be given by the following formula:

$$E_{G_{w3SAT}} = \sum_{i=1}^{m} \left(\prod_{j=1}^{l} Q_{ij} \right). \tag{11}$$

Where m is the number of third-order clauses and l is the number of variations in G_{w3SAT}. The inconsistency of G_{w3SAT} is determined as in equation (12):

$$Q_{ij} = \begin{cases} \frac{1}{2}(1 + S_{A_i}), & \text{if } A_i \\ \frac{1}{2}(1 - S_{A_i}), & \text{if} \neg A_i \end{cases} \tag{12}$$

In order to make sure the cost function to be zero, at least one interpretation must be satisfied. The literals in each clause are connected by DNF so that the clause will be satisfied when at least one neuron state is satisfied. On the contrary, G_{w3SAT} cannot be satisfied when $E_{G_{w3SAT}}$ cannot be equal to zero. In this case, the synaptic weights must be initialized randomly. In this study, the weights for DHNN-w3SAT are obtained by using the Wan Abdullah method depends on identifying logical inconsistencies [13]. Each neuron is assigned a truth value, the objective is to minimize the cost function by maximizing the number of satisfied clauses. The local field plays an important role in suppressing the retrieval output before finding the final neuron state. The equation of local field can be used to modify the neurons state. Equation (13) indicates the formula of local field h_i,

$$h_i = \sum_{k=1,k\neq i,j}^{n} \sum_{j=1,j\neq i}^{n} W_{ijk}^{(3)} S_j S_k + \sum_{j=1,j\neq i}^{n} W_{ij}^{(2)} S_j + W_i^{(1)}. \tag{13}$$

DHNN employs the Hyperbolic Tangent Activation Function (HTAF) due to its non-linear properties, which facilitate the convergence of the final neuron states [25]. The local field influences the effectiveness of the final neuron states generated by the DHNN. Subsequently, these final states are evaluated to determine whether the solution is overfitting. The updating equation is determined by equation (14) as below:

$$S_i(t) = \begin{cases} 1, & \tanh(h_i) \geqslant 0, \\ -1, & \text{otherwise.} \end{cases} \tag{14}$$

$$\tanh(h_i) = \frac{e^{h_i} - e^{-h_i}}{e^{h_i} + e^{-h_i}}. \tag{15}$$

Equation (15) is for HTAF. It is worth mentioning that the updated neuron states would be the same as the initial neuron states when $h_i = 0$. In the learning phase, the cost function will be compared with the Lyapunov energy function as shown in equation (16), and the minimum value of the function can be calculated by equation (17):

$$L_G = -\frac{1}{3}\sum_{i=1}^{n} \sum_{k=1,k\neq i,j}^{n} \sum_{j=1,j\neq i}^{n} W_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2}\sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} W_{ij}^{(2)} S_i S_j - \sum_{i=1}^{n} W_i^{(1)}. \tag{16}$$

$$L_G^{\min} = -\frac{m}{8}. \tag{17}$$

The quality of the final neuron states can be evaluated through equation (18). This equation can also be used to make distinction between the local minimum solutions and the global minimum solutions,

$$\left| L_G - L_G^{\min} \right| \leqslant \text{Tol}. \tag{18}$$

Where Tol is a tolerance value which is determined previously. Figure 2 illustrates the schematic diagram showing the implementation of DHNN-w3SAT. The diagram is divided into two primary phases: the learning phase and the retrieval phase. During the learning phase, the clauses for G_{w3SAT}

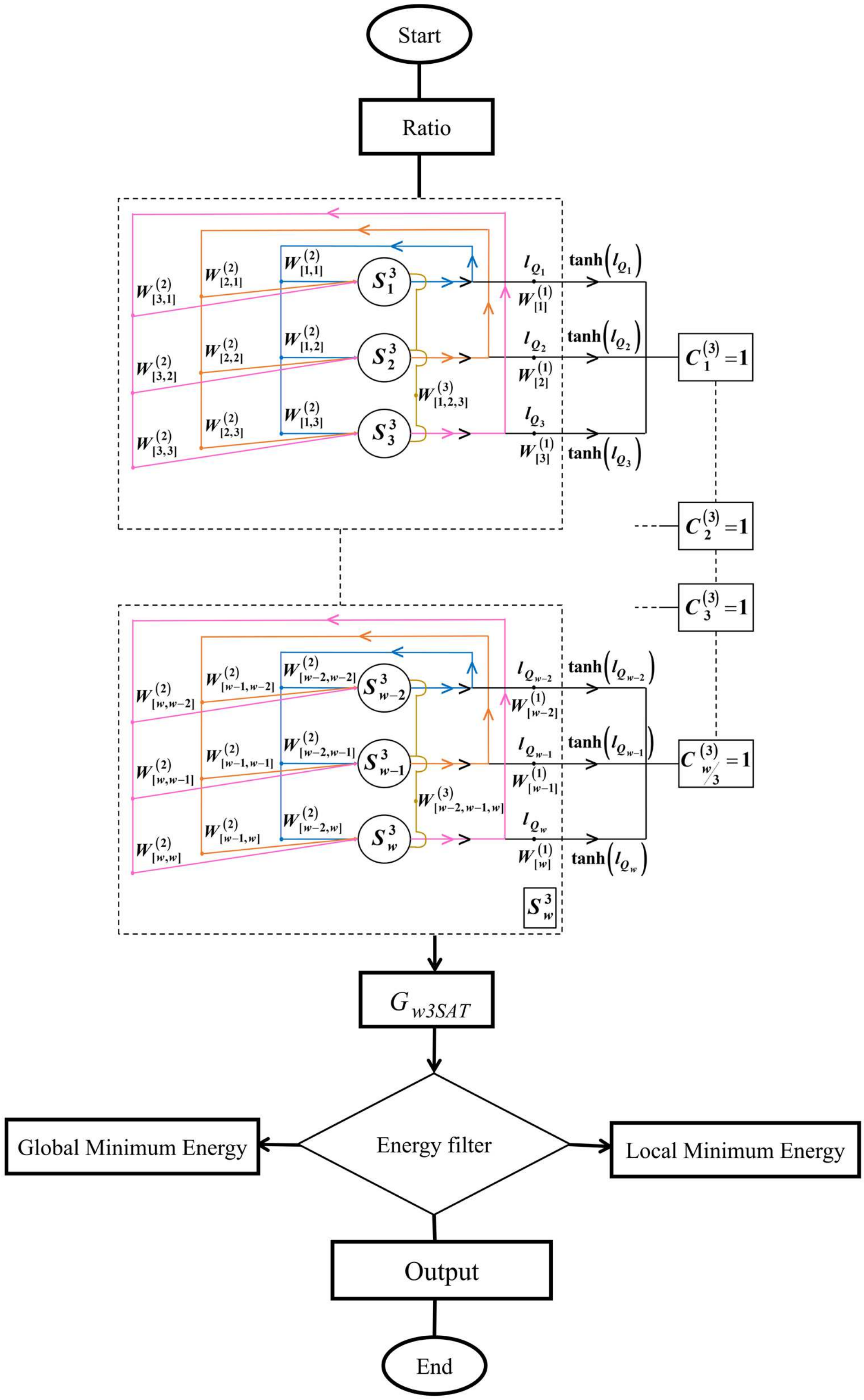

Fig. 2. Schematic diagram of DHNN-w3SAT.

are determined randomly and then translated into Boolean algebra representation. In this phase, the optimal synaptic weights can be obtained by achieving the optimal neuron. In the retrieval phase, the energy function was utilized to evaluate whether the solutions are local or global minimum.

4. Experimental settings and evaluation metrics

The proposed networks were conducted using Python 3 coding software with a specification of a 3.30 GHz Radeon Graphics processor with 16 GB RAM, and a 64-bit Windows 11 operating system. All the experiments have been done based on the same computer with the same compiler in order to ensure unbiased interpretation of the findings. In this section, the proposed logic rule implemented in DHNN will be evaluated by various performance metrics. This model will undergo three parts: the logic phase, the learning phase and the retrieval phase. The details of each phase are performed as below:

(a) Different number of literals which leads to different number of clauses.

(b) Different values of ratio. Different ratio determines different number of negative literals.

(c) Various metrics. The value of metrics are used to evaluate the effectiveness of the models proposed.

In the logic phase, rounding function was used to obtain the correct number of negative literals. The neuron population is set to 100 in order to avoid the early convergence. Higher number of neurons can have a higher capacity of information storage. Due to the objective of gaining optimal global solutions, the number of neurons would not be settled too high. Table 1 provides the parameters involved in w3SAT. Table 2 indicates the symbols in each phase of the logic which are used to evaluate the value of the metrics.

Table 1. List of parameters for DHNN-w3SAT.

Parameter	Parameter Value
Number of neurons, N	$20 \leqslant N \leqslant 100$
Ratio of Weight, r	[0.1,0.9] [26]
Number of Negative Literals, α	$[rn]$
Step of Ratio, Δr	0.1 [23]
Number of combinations, N_C	100 [16]
Number of learning, N_l	100 [16]
Current number of learning, N_{cl}	$N_{cl} \leqslant N_l$
Number of iterations, N_i	$N_i \leqslant N_l$ [23]
Number of testing trials, N_t	100 [16]
Tolerance value, Tol	0.001 [20]
Activation function	HTAF [25]
Neuron states in learning phase	Random
Neuron states in retrieval phase	Random
Synaptic weight method	Wan Abdullah method (WA) [13]

Table 2. List of symbols for DHNN-w3SAT.

Symbol	Symbol meaning
N_{mf}	Max value of fitness
N_{cf}	Current number of fitness
N_{ls}	Number of local solutions
N_{gs}	Number of global solutions
N_{ts}	Number of total solutions

For each phase of the study, the performance metrics are employed for evaluating the efficiency of the proposed model DHNN-w3SAT in order to compare with other existing approaches. In this article, the performance assessment focus on two phases of DHNN: the learning phase and the retrieval phase. DHNN model focuses on reducing discrepancy, quantified by the cost function of networks during the learning phase. The model assessing proposed w3SAT evaluates the ultimate configuration achieving the lowest energy state, while also assessing the similarity among neuron states during the retrieval phase.

Due to the maximum of fitness in the logic phase related to the structure of w3SAT, the better the structure, the lower the cost function. In this part of work, mean absolute error (MAE), root means square error (RMSE), and mean absolute percentage error (MAPE) were used to measure the effect of the performance of the models and systems. MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. RMSE represents the square root of the average of squared differences between prediction and actual observation. MAPE expresses the error as a percentage relative to the actual values. MAE and RMSE are in the same unit as the target variable, making them directly interpretable. MAPE provides a percentage error, which can be easier to interpret in some contexts [27]. MAE is less sensitive to outliers compared to RMSE, which squares the errors. MAPE can be problematic when actual values are close to zero, leading to undefined or infinite values. Equations (19)–(21) show the details about these metrics in the logic phases,

$$\mathrm{MAE}_l = \sum_{cf=1}^{N_l} \frac{|N_{mf} - N_{cf}|}{N_{cl}}, \tag{19}$$

$$\mathrm{RMSE}_l = \sum_{cf=1}^{N_l} \sqrt{\frac{(N_{mf} - N_{cf})^2}{N_{cl}}}, \tag{20}$$

$$\mathrm{MAPE}_l = \sum_{cf=1}^{N_l} \frac{|N_{mf} - N_{cf}|}{N_{mf} \cdot N_{cl}}. \tag{21}$$

In the learning phase, the best fitness will be obtained when the condition mentioned in equation (9) is satisfied. The better the logic model is, the lower the value of metrics mentioned above. When assessing the efficacy of the logic phase using equations (19)–(21), we evaluate both the fitness of neuron states and the clauses that produce optimal synaptic weights. These metrics collectively assess the performance of the training phase and analyze errors in synaptic weight management. The formula of testing error and energy error will also be discussed during the retrieval phase. Global minimum solutions represent the optimal outcomes in optimization problems, playing a crucial role in achieving the best possible results across diverse domains of application [22]. Locating the global minimum can be challenging, especially in complex and high-dimensional spaces where multiple local minima may exist. The ratio of global minimum solutions (r_G) plays an important role in the present metrics. This metric must be considered during the retrieval phase of the logic model. Equations (22)–(24) show the value of MAE, MAPE in the retrieval phase and the ratio of global minimum r_G which are worth mentioning,

$$\mathrm{MAE}_r = \frac{1}{N_t \cdot N_{ts}} \sum_{i=1}^{N_t} N_{ls}, \tag{22}$$

$$\mathrm{MAPE}_r = \sum_{i=1}^{N_t} \frac{1}{100 \cdot N_{ts}} |N_{ts} - N_{gs}|, \tag{23}$$

$$r_G = \frac{1}{N_t \cdot N_{ts}} \sum_{i=1}^{N_t} N_{gs}. \tag{24}$$

The final neuron states retrieved by DHNN-w3SAT will also be analyzed by similarity metric. Jaccard similarity index (JSI) is a measure used to assess the similarity between two sets. It is defined as the size of the intersection of the sets divided by the size of the union of the sets [28]. In the work proposed in this article, JSI is used in the recommendation model comparing the similarity of documents based on the characteristics. This index is defined in equation (25),

Table 3. Variables for similarity index JSI.

Variable	$S_i^{\max}$	S_i
o	1	1
p	1	−1
q	−1	1

$$J\left(S_i^{\max}, S_i\right) = \frac{o}{o+p+q}, \tag{25}$$

where each symbol is defined in Table 3.

Jaccard's index of similarity provides a simple yet effective way to quantify similarity between sets, making it a valuable tool in various fields for comparing and analyzing relationships between data points or entities based on their attributes or characteristics.

5. Results and discussion

In this section, the proposed logical output is measured by the utilization of evaluation metrics. The objective is to clarify the efficacy integrating a w3SAT structure which only consists of third-order clauses. In other words, w3SAT maintains a special logic rule with strictly 3 literals per clause which is called 3SAT extended by adding a ratio. The characteristic of this kind of logic is they may have a high quality of obtaining high ratio of global minimum solutions. Although 3SAT has broad theoretical applications, there is currently no efficient polynomial-time algorithm available to solve large and complex instances. Therefore, in practical applications, heuristic algorithms or approximation algorithms are often relied upon. This necessitates specific problem analysis in different domains when applying 3SAT, in order to choose appropriate solving strategies and tools.

For instance, we compared the novel proposed logic with existing random 3-satisfiability to better illustrate the differences. After adding the ratio, the values of the metrics are lower. A low MAE and RMSE value not only reflects the model's accuracy in fitting the training data but also indicates its ability to avoid overfitting. Achieving a low MAE or RMSE value is crucial as it helps in obtaining optimal synaptic weights in the learning phase. Figures 3–5 show the trend of MAE, RMSE and also MAPE in the learning phase of w3SAT.

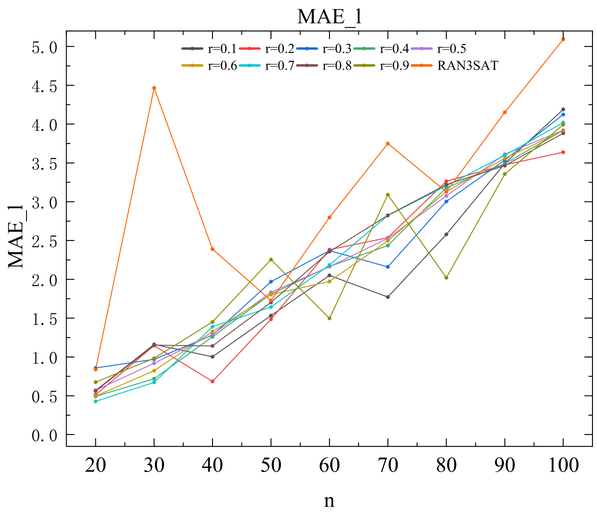

Fig. 3. MAE_l represents the trends of MAE in the learning phase.

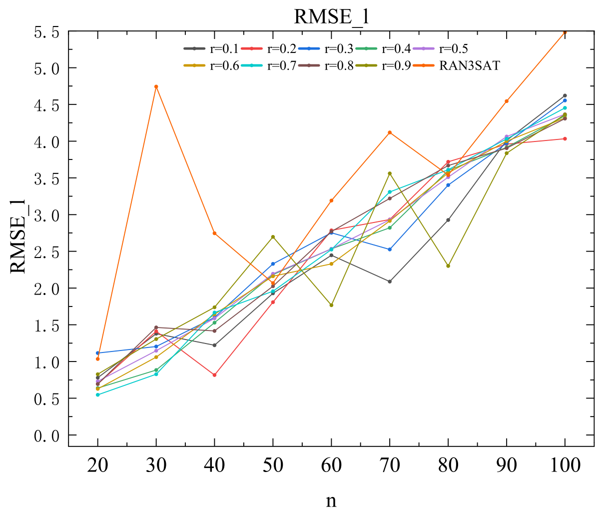

Fig. 4. $RMSE_l$ represents the trends of RMSE in the learning phase.

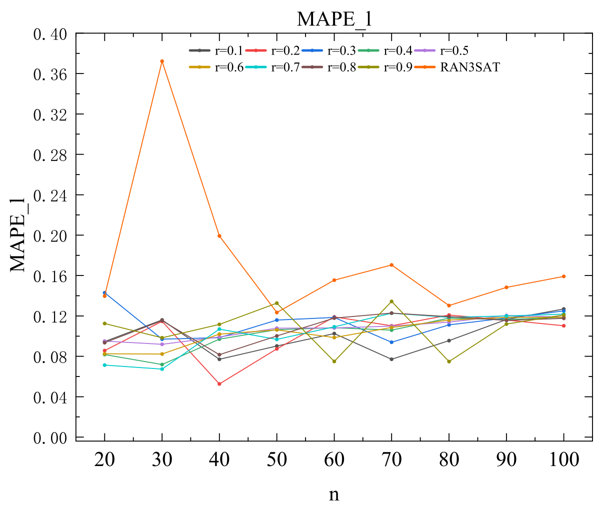

Fig. 5. $MAPE_l$ represents the trends of MAPE in the learning phase.

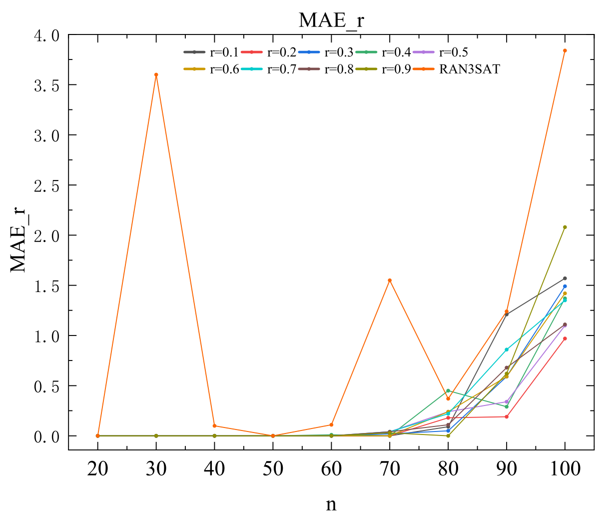

Fig. 6. MAE_r represents the trends of MAE in the retrieval phase.

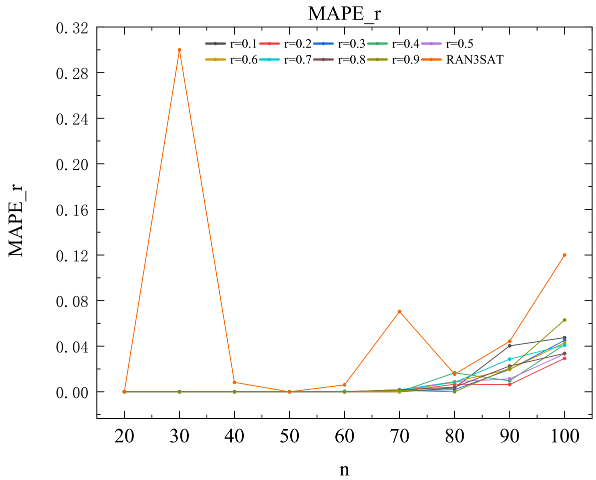

Fig. 7. MAPE_r represents the trends of MAPE in the retrieval phase.

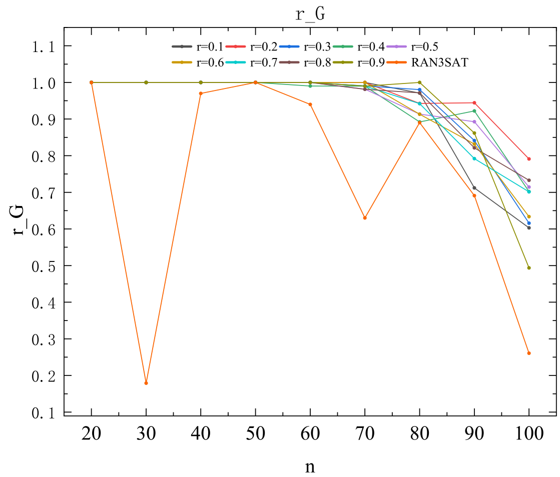

Fig. 8. r_G represents the global ratio for each various situations mentioned in the article.

The results of the metrics shown in Figures 3–5 implements that when the number of neurons increases, the errors increased also. The increase of neuron number can also lead a misconception of the information which is needed in the retrieval phase. Figures 6–7 represent the metrics of MAE and MAPE in the retrieval phase. Errors in the learning phase can significantly influence retrieval outcomes, and addressing these errors through feedback and reflection can improve both the learning and retrieval processes. Furthermore, the rate of global minimum solutions is also represented in Figure 8. As can be seen from the above figures, w3SAT, which adds weights, is better than RAN3SAT in all error analysis indicators. This also shows the positive impact of adding negative weights on logical rules. The Jaccard value, also known as the Jaccard similarity index, measures the similarity between two sets. This work compared all the sets of logical rules with different ratio and RAN3SAT. Figure 9 demonstrates the role of weight which can obviously improve the value of JSI compared with RAN3SAT. The JSI values with the added ratio of negative literals r generally show fluctuations across different values of r. Specifically, lower values of r (0.1 and 0.2) tend to yield lower JSI values, indicating less overlap between the sets compared to the baseline RAN3SAT JSI values. As r increases (from 0.3 to 0.9), the JSI values often increase, suggesting greater similarity between the sets relative to the benchmark states. Since a low Jaccard index helps identify dissimilar or distinct groups within a dataset, our further work needs to improve this proposed model to reach a lower value of the metrics of JSI.

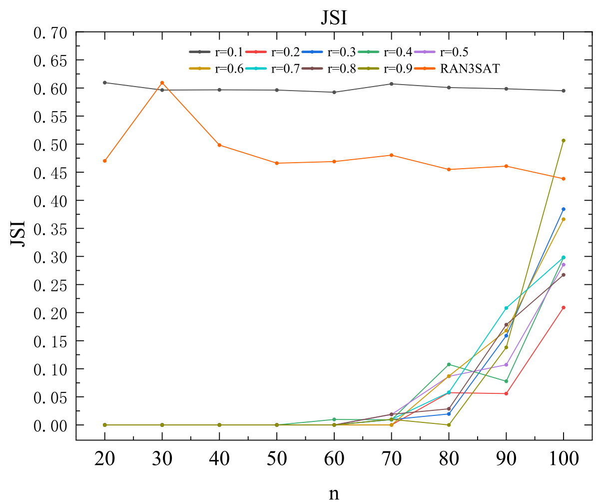

Fig. 9. JSI is the Jaccard similarity index value for each logic modelrepresents the trends of MAPE in the learning phase.

Overall, adding weight r influences the JSI by adjusting the overlap between sets, which can be advantageous for tasks where increased overlap or similarity is desired. However, the effectiveness of each weight r in achieving specific goals would depend on the context and objectives of the analysis or application. The overall trend shows that as the number of neurons increases, JSI values typically vary, indicating that as the problem size grows, the similarity between sets may also change.

6. Conclusion

In this study, a novel systematic logic rule named w3SAT is implemented in DHNN. Depending on the great advantages of the structure of higher order logic, the proposed logic has numerous optimal aspects. First of all, w3SAT demonstrates high capacity for neuron variation when the ratio of third-order clauses increases. Furthermore, the DHNN-w3SAT approach demonstrates proficiency in retrieving precise synaptic weights, thereby achieving global minimum solutions. Finally, by comparing different ratios of w3SAT, the results show fewer errors with greater various of neuron states. The proposed logic also demonstrates great capacity for managing synaptic weights due to the minimization of energy. Additionally, high rate of global minimum optimization solutions is obtained when the number of neurons is set to 100. In conclusion, this research highlights the enhanced capabilities of the DHNN-w3SAT approach.

Notably, the robust architecture of artificial neural networks, combined with our innovative logic, forms a solid base for practical applications. In this framework, each neuron represents specific data attributes which are integrated into our logic-mining approach. This integration facilitates the development of induced logic with predictive and classification capabilities. For further work, our logic mining will emphasize the incorporation of control mechanisms to enhance the representation of complex dynamics and more accurately capture real-world behaviors within neural networks.

[1] Vaughan J. W. Making better use of the crowd: How crowdsourcing can advance machine learning research. Journal of Machine Learning Research. **18** (193), 1–46 (2018).

[2] Agatonovic-Kustrin S., Beresford R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. Journal of Pharmaceutical and Biomedical Analysis. **22** (5), 717–727 (2000).

[3] Da Costa N. L., De Lima M. D., Barbosa R. Analysis and improvements on feature selection methods based on artificial neural network weights. Applied Soft Computing. **127**, 109395 (2022).

[4] Huixian J. The analysis of plants image recognition based on deep learning and artificial neural network. IEEE Access. **8**, 68828–68841 (2020).

[5] Kim C.-G., Hwang Y.-J., Kamyod C. A study of profanity effect in sentiment analysis on natural language processing using ANN. Journal of web engineering. **21** (3), 751–766 (2022).

[6] Jalal F. E., Xu Y., Iqbal M., Javed M. F., Jamhiri B. Predictive modeling of swell-strength of expansive soils using artificial intelligence approaches: ANN, ANFIS and GEP. Journal of Environmental Management. **289**, 112420 (2021).

[7] Hopfield J. J., Tank D. W. "Neural" computation of decisions in optimization problems. Biological Cybernetics. **52** (3), 141–152 (1985).

[8] Shameer S., Baghalian K., Cheung C. M., Ratcliffe R. G., Sweetlove L. J. Computational analysis of the productivity potential of CAM. Nature Plants. **4** (3), 165–171 (2018).

[9] Joya G., Atencia M. A., Sandoval F. Hopfield neural networks for optimization: study of the different dynamics. Neurocomputing. **43** (1–4), 219–237 (2002).

[10] Swarup K. S., Simi P. V. Neural computation using discrete and continuous Hopfield networks for power system economic dispatch and unit commitment. Neurocomputing. **70** (1–3), 119–129 (2006).

[11] Yu Z., Abdulghani A. M., Zahid A., Heidari H., Imran M. A., Abbasi Q. H. An overview of neuromorphic computing for artificial intelligence enabled hardware-based hopfield neural network. IEEE Access. **8**, 67085–67099 (2020).

[12] Barco R., Lázaro P., Díez L., Wille V. Continuous versus discrete model in autodiagnosis systems for wireless networks. IEEE Transactions on Mobile Computing. **7** (6), 673–681 (2008).

[13] Abdullah W. A. T. W. Logic programming on a neural network. International Journal of Intelligent Systems. **7** (6), 513–519 (1992).

[14] Vizel Y., Weissenbacher G., Malik S. Boolean satisfiability solvers and their applications in model checking. Proceedings of the IEEE. **103** (11), 2021–2035 (2015).

[15] Mouhoub M., Sadaoui S. Systematic versus non-systematic methods for solving incremental satisfiability. Innovations in Applied Artificial Intelligence. 543–551 (2004).

[16] Mohd Kasihmuddin M. S., Mansor M. A., Md Basir M. F., Sathasivam S. Discrete mutation Hopfield neural network in propositional satisfiability. Mathematics. **7** (11), 1133 (2019).

[17] Mansor M. A., Sathasivam S. Optimal performance evaluation metrics for satisfiability logic representation in discrete hopfield neural network. International Journal of Mathematics and Computer Science. **16** (3), 963–976 (2021).

[18] Maneva E., Sinclair A. On the satisfiability threshold and clustering of solutions of random 3-SAT formulas. Theoretical Computer Science. **407** (1–3), 359–369 (2008).

[19] Sathasivam S., Mansor M. A., Ismail A. I. M., Jamaludin S. Z. M., Kasihmuddin M. S. M., Mamat M. Novel random k satisfiability for $k \leqslant 2$ in Hopfield neural network. Sains Malaysiana. **49** (11), 2847–2857 (2020).

[20] Karim S. A., Zamri N. E., Alway A., Kasihmuddin M. S. M., Ismail A. I. M., Mansor M. A., Hassan N. F. A. Random satisfiability: A higher-order logical approach in discrete Hopfield Neural Network. IEEE Access. **9**, 50831–50845 (2021).

[21] Guo Y., Kasihmuddin M. S. M., Gao Y., Mansor M. A., Wahab H. A., Zamri N. E., Chen J. YRAN2SAT: A novel flexible random satisfiability logical rule in discrete hopfield neural network. Advances in Engineering Software. **171**, 103169 (2022).

[22] Gao Y., Guo Y., Romli N. A., Kasihmuddin M. S. M., Chen W., Mansor M. A., Chen J. GRAN3SAT: Creating flexible higher-order logic satisfiability in the discrete hopfield neural network. Mathematics. **10** (11), 1899 (2022).

[23] Zamri N. E., Azhar S. A., Mansor M. A., Alway A., Kasihmuddin M. S. M. Weighted random k satisfiability for $k = 1, 2$ (r2SAT) in discrete Hopfield neural network. Applied Soft Computing. **126**, 109312 (2022).

[24] Karim S. A., Kasihmuddin M. S. M., Sathasivam S., Mansor M. A., Jamaludin S. Z. M., Amin M. R. A novel multi-objective hybrid election algorithm for higher-order random satisfiability in discrete hopfield neural network. Mathematics. **10** (12), 1963 (2022).

[25] Kılıçarslan S., Adem K., Çelik M. An overview of the activation functions used in deep learning algorithms. Journal of New Results in Science. **10** (3), 75–88 (2021).

[26] Muhammad Sidik S. S., Zamri N. E., Mohd Kasihmuddin M. S., Wahab H. A., Guo Y., Mansor M. A. Non-systematic weighted satisfiability in discrete hopfield neural network using binary artificial bee colony optimization. Mathematics. **10** (7), 1129 (2022).

[27] Chicco D., Warrens M. J., Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. Peerj Computer Science. **7**, e623 (2021).

[28] Real R., Vargas J. M. The probabilistic basis of Jaccard's index of similarity. Systematic Biology. **45** (3), 380–385 (1996).

Зважена випадкова 3-здійсненність у дискретній нейронній мережі Хопфілда

Лю Х.[1,2], Чан Ю.[2,3], Цзян Х.[2]

[1] *Загальноосвітня школа, Гуанчжоуський коледж технологій і бізнесу, 510850 Гуанчжоу, Китай*
[2] *Школа математичних наук, Університет Сайнс Малайзії, Пенанг 11800 USM, Малайзія*
[3] *Школа інформатики та інженерії, Хунаньський технологічний інститут, 421002 Хен'ян, Китай*

Завдання здійсненності (SAT) відіграє важливу роль в обчислювальній математиці, оскільки дає змогу представляти інформацію різних типів даних. Останні дослідження все частіше зосереджуються на моделюванні дискретної нейронної мережі Хопфілда (DHNN) на основі логічних SAT-формул. Попри широке використання SAT у DHNN, питання контролю розподілу позитивних і негативних літералів у логічних правилах досі залишалося майже не дослідженим. У цій статті запропоновано нове логічне правило — зважене 3-задоволення (w3SAT), що передбачає введення співвідношення негативних літералів у кластери формул 3SAT. Реалізація w3SAT у DHNN ґрунтується на мінімізації логічної неузгодженості через побудову відповідної цільової функції. Ефективність запропонованої логіки оцінено за низкою наперед визначених метрик. Результати показують зменшення похибок на всіх етапах при значеннях співвідношення негативних літералів понад 0,2. За низького значення $r = 0{,}1$ індекс подібності майже не відрізняється від показників для RAN3SAT. Загалом, запропоноване логічне правило продемонструвало кращі результати порівняно з відомими моделями, зокрема — вищу здатність до досягнення глобальних мінімумів при певних значеннях частки негативних літералів.

Ключові слова: *зважений; здійсненність; дискретна нейронна мережа Хопфілда; систематичність.*