



Research article

Conditional random k satisfiability modeling for $k = 1, 2$ (CRAN2SAT) with non-monotonic Smish activation function in discrete Hopfield neural network

Nurshazneem Roslan^{1,2}, Saratha Sathasivam^{1,*} and Farah Liyana Azizan^{1,3}

¹ School of Mathematical Sciences, Universiti Sains Malaysia, USM, Penang 11800, Malaysia

² Institute of Engineering Mathematics, Universiti Malaysia Perlis, Arau, Perlis 02600, Malaysia

³ Centre for Pre-University Studies, Universiti Malaysia Sarawak, Kota Samarahan 94300, Sarawak, Malaysia

* **Correspondence:** Email: saratha@usm.my; Tel: +6046532428.

Abstract: The current development of logic satisfiability in discrete Hopfield neural networks (DHNN) has been segregated into systematic logic and non-systematic logic. Most of the research tends to improve non-systematic logical rules to various extents, such as introducing the ratio of a negative literal and a flexible hybrid logical structure that combines systematic and non-systematic structures. However, the existing non-systematic logical rule exhibited a drawback concerning the impact of negative literal within the logical structure. Therefore, this paper presented a novel class of non-systematic logic called conditional random k satisfiability for $k=1,2$ while intentionally disregarding both positive literals in second-order clauses. The proposed logic was embedded into the discrete Hopfield neural network with the ultimate goal of minimizing the cost function. Moreover, a novel non-monotonic Smish activation function has been introduced with the aim of enhancing the quality of the final neuronal state. The performance of the proposed logic with new activation function was compared with other state of the art logical rules in conjunction with five different types of activation functions. Based on the findings, the proposed logic has obtained a lower learning error, with the highest total neuron variation $TV=857$ and lowest average of Jaccard index, $JSI=0.5802$. On top of that, the Smish activation function highlights its capability in the DHNN based on the result ratio of improvement Zm and TV . The ratio of improvement for Smish is consistently the highest throughout all the types of activation function, showing that Smish outperforms other types of activation functions in terms of Zm and TV . This new development of logical rule with the non-monotonic Smish activation

function presents an alternative strategy to the logic mining technique. This finding will be of particular interest especially to the research areas of artificial neural network, logic satisfiability in DHNN and activation function.

Keywords: discrete Hopfield neural network (DHNN); conditional random two satisfiability; non-systematic logic; activation functions; Smish activation function; potential logic mining

Mathematics Subject Classification: 68N17, 68R07, 68T27

1. Introduction

The artificial neural network (ANN) is a computing framework that reflects the capability of the human brain to process information. The ANN's complex structure is due to the extensive interconnectivity of its neurons. By incorporating external input, the artificial neural network is trained to successfully execute a given task by utilizing all available neurons. Hopfield and Tank introduced the discrete Hopfield neural network (DHNN) in 1985 as one of the original subsets of ANN utilized in the solving of optimization issues. The DHNN has features such as fault tolerance, content addressable memory (CAM) and dynamical energy. The minimal energy of DHNN shows that the retrieved final neuron state is a global solution. Therefore, the outcome of the final energy function in DHNN is significantly influenced by the synaptic weight value gained by the network. Therefore, incorporating a symbolic rule is necessary to guarantee that the network consistently converges toward optimal solution. To address this problem, [1] is the primary work that incorporates satisfiability (SAT) logical rules into ANN. He introduced a basic principle of conducting logic programming within a DHNN through comparing the cost function and the Lyapunov energy function. Consequently, the connection between the logic (synaptic weight) is successfully computed by using the Abdullah method. This approach surpassed traditional learning methods like Hebbian learning [2]. Due to this success, [3] introduced Horn satisfiability (HORNSAT), a novel SAT concept in DHNN, incorporating a relaxation process during the retrieval phase. The result shows that HORNSAT can converge toward the absolute minimum energy effectively, and it is found that as the network becomes complex, the dynamic relaxation rate outperforms the constant relaxation rate. However, the implications of various logical rules within DHNNs remain unclear due to the constraints imposed by the HORNSAT structure. Therefore, a unique logical framework is needed, which allows more freedom with each clause not being limited to just one positive literal.

The authors of [4,5] extended the research to include other forms of structured logical rules, namely two satisfiability (2SAT) and three satisfiability (3SAT), respectively. The distinction between these two logical structures is that 2SAT contains two literals in each clause, whereas every clause within the 3SAT logic rule specifically includes three literals. The proposed 2SAT in DHNN demonstrates a high global minima ratio within a reasonable computational time. Even as the storage capacity of DHNN with 3SAT increases exponentially, the investigation into 3SAT has yielded an ideal value for the global minima ratio. Taking an alternative approach, [6] presented the integration of systematic 2SAT into the radial basis function neural network. This integration involves determining the center and width parameter values, which in turn enables the generation of output weights aligned with the 2SAT logic. Additionally, [7] proposed different evolutionary algorithms in the training phase of radial basis function neural network corresponding to 2SAT logic. Based on the result, the evolutionary programming algorithm is effective during the training phase and achieves an optimal

output weight. Although systematic logical structures have been successfully integrated into DHNN, they suffer from a lack of diversity in terms of the logical structure. This leads to an overfitting of synaptic weights due to the similarity between the clauses. As a solution, the scope of the SAT domain in DHNN has expanded to the non-systematic logical structures. Non-systematic logical structures are distinguished by having a specific number of literals in each clause. This unique representation enables the incorporation of first-order logic into the current systematic logical structure. The types of logical rule have been expanded to the non-systematic logical structures. Initially, [8] proposed random k satisfiability by combining first-order and second-order logic for $k=1,2$. Random 2SAT(RAN2SAT) offers the flexibility of having a dynamic number of literals within the clause. However, experimental results showed that first-order clauses introduced more logical inconsistencies than second-order clauses. As a result, the DHNN was unable to finish the learning phase effectively due to an increased number of first-order clauses. This ultimately led to a less effective retrieval phase. Therefore, there are various perspectives that have been explored in the development of non-systematic logical structure. In contrast, [9] has proposed Y-type random 2SAT (YRAN2SAT), a flexible hybrid logical structure that combines systematic and non-systematic structures. It offers random enumeration of first-order, second-order or both types of clauses. The study implemented five possible pathways of YRAN2SAT that show an improved solution capacity. Despite the successful flexibility of YRAN2SAT in DHNN, the non-systematic structure of YRAN2SAT has limited capacity to retrieve diversified solution. [10] introduced weighted random k satisfiability (r2SAT) for $k=1,2$ as a new class of non-systematic logic with weighted ratios of negative literals. They also integrated a logical phase for generating the non-systematic structure with the designated quantity of negative literals. As a result, r2SAT performs well in producing diverse neuron states and global minima solutions. Additionally, [11] proposed an alternative approach where the logic phase of r2SAT is modified by integrating a binary artificial bee colony algorithm to control the distribution of negative literals. This approach has increased the global minima ratio. As a consequence of this discovery, using a dynamic allocation of negative literals will aid in generating a higher number of global minimum solutions with diverse final neuron states. However, the importance of having negative literal in the logical structure remains unclear and requires more analysis.

Furthermore, the existing logical structure for the above logical rule is fully utilized c possible combination of clauses ($c=2^k$), where k is the order of the clause. For example, the first order clause will have two possible combinations of clauses, while the second order clause will have four possible combinations of clauses and the third order clause will have eight possible combinations of clauses, which will be randomly generated. As stated earlier, r2SAT regulates the proportion of negative literals within a clause. However, there is still a chance that all possible combinations will be generated. Thus, instead of evaluating the significance of including negative literals in the logical framework, it is equally crucial to assess the efficiency achieved by eliminating clauses with only positive literals (for the second order clause). Another interesting study in logic satisfiability is concerned about the quality of the final neuron state that is retrieved by the DHNN. Activation function is a crucial parameter within a DHNN. Activation functions are used to transform the weighted sum of inputs into a neuron's output signal. Therefore, the activation function plays a crucial role in retrieving the final neuron state within DHNN. An optimal activation function can provide greater variations and enhanced diversity in the final neuron states. For many years, a lot of activation functions have been explored in the neural network. [12] has conducted a survey that presents the developments of activation functions in neural networks. This survey has examined the performance of 18 activation functions across different types of networks. This survey mentioned that the hyperbolic tangent activation function is suffering from the gradient vanish problem. In another review, [13] conducted a survey into trainable activation

functions within the neural network field. In this paper, a comprehensive survey of trainable activation functions has been presented. Based on the review, the neural networks utilizing trainable activation functions can reduce computational complexity as they involve fewer parameters to manage. The traditional activation function used in neural networks is hyperbolic tangent activation function. [14] has explored the two-dimensional parameter space of the system by analyzing hyperbolic tangent and piecewise-linear activation functions. While [15] uses the classical hyperbolic tangent activation function with a proposed simple Rectified Linear Unit (ReLU) activation function to develop circuit's implementation of the Hopfield neural network, the traditional activation function used in logic satisfiability DHNN is hyperbolic tangent activation function (HTAF). [16] has proposed HTAF and the Elliot symmetric activation function in doing 3SAT in DHNN. In this work, various activation functions have been employed as dynamic post-optimization techniques to convert the activation level of a unit (neuron) into an output signal. In terms of global solutions, global hamming distance and central processing unit or computational time, the HTAF is noted to exhibit superior performance compared to the Elliot symmetric activation function and the McCulloch-Pitts function. Despite the acceptable performance of the HTAF, there remains questions about the interpretability of error analysis and the quality of final neuron states in the retrieval phase of DHNN, indicating a need for further analysis. Hence, through the application of an optimal activation function, the efficiency of the updating rule in DHNN can be enhanced. This results in a greater variety and a higher degree of diversity in the final neuron states.

Therefore, the present study addresses these gaps by introducing a new type of non-systematic random k satisfiability, for $k=1,2$ that utilizes first and second order logic with a specific condition. The suggested logic will eliminate positive literals from the second-order clauses while not imposing any constraints on the first-order clauses. This highlights the significance of negative literals in the logical structure, and further examination can be conducted on synaptic weight management for this proposed logic. On the other hand, by implementing an optimal activation function, the update rule in the DHNN can be made more effective, providing a broader variation and increased diversity in the final states of the neurons. [10] represents the nearest effort in recognizing the potential impact of negative literals within a logic satisfiability DHNN, while [16] is the most relevant study that analyzes various activation functions in a logic satisfiability DHNN. Thus, the contribution of this paper as follows:

- 1) To formulate a new non-systematic logical rule, namely, conditional random k satisfiability, where $k=1,2$ and uses first order and second order logic without including both positive literal in the second order clauses. The proposed conditional RAN2SAT will demonstrate the role of negative literal in the second order clause.
- 2) To incorporate the proposed conditional RAN2SAT into DHNNs by reducing the logical inconsistency of the logical rule corresponding to the zero-cost function. The behavior of the DHNNs will be influenced by the cost function derived in line with the proposed logic.
- 3) To introduce a novel non-monotonic Smish activation function intended to improve the efficacy of the updating rule within DHNNs, which is expected to contribute to greater variety and enhance diversity in the final neuron states. The performance of the Smish activation function will be evaluated with different types of activation functions during the retrieval phase of DHNNs.
- 4) To investigate the performance of conditional RAN2SAT in DHNNs. In addition, the capability of non-monotonic Smish activation function with different logical rules will be analyzed. Various performance metrics, such as learning error, testing error, energy profile and similarity

metric, are presented to validate the performance of the proposed conditional random k satisfiability, which utilizes the Smish activation function.

The article is structured as follows: Section two discusses the motivation behind the research. Followed by this, section three introduces the basic formulation of conditional random k satisfiability for $k=1,2$, which is also abbreviated as conditional random k satisfiability (CRAN2SAT). Section four elaborates the detailed presentation of the proposed model, CRAN2SAT, incorporating with DHNN. Furthermore, section five provides a concise discussion of the proposed activation function used during the retrieval phase of the DHNN model. Section six introduces the experimental configurations and the metrics used to evaluate performance in this study. Section seven explores the results and related discussions about the performance of the proposed DHNN-CRAN2SAT logic. This also illustrates the influence of activation functions on optimizing the final neuron state of the DHNN model, based on the selected metrics. The findings of this study are summarized in section eight.

2. Motivation

2.1. The exploration of satisfiability representation in DHNN

The current development of satisfiability representation is significant in the field of computer science and mathematics where it offers an alternative method of representing the information of any datasets. Therefore, the purpose of satisfiability representation integrated with DHNN is to enable the user to interpret the output from the network. In simpler terms, satisfiability representation can be understood as a logical guideline that depicts the output generated by the network. Even though there are various types of satisfiability representations integrated with DHNN, there is still an empty room for exploration regarding the structure, behavior and their potential application. The logical rule that has been proposed by other researchers [8–10,17–19] shows that each of the logical structures has their own characteristic, whereby all the logical rule has been successfully embedded in DHNN. Thus, this motivates this study to explore the satisfiability representation in other perspectives, because different satisfiability representation will demonstrate a different performance analysis. Consequently, various types of logical rules can help practitioners or engineers to choose the best logic that suits their problem. Satisfiability representation finds application in various domains including quantum chemistry [20], classification methods [21], chaos computing [22] and in logic mining methodologies [23]. Therefore, by exploring other types of logical rule in DHNN, it is expected to be a guidance for the outsider researcher when choosing the best logic according to their problem or datasets.

2.2. The effect of negative literal in second order clause

The logical combination of SAT is highly influenced by the order of clauses in non-systematic SAT [8–10]. For example, k SAT generates c possible combinations of clauses ($c=2^k$), where k represents the clause order. A new class of non-systematic logic called weighted random k satisfiability was introduced by Zamri et al. [10]. It incorporates weighted ratios of negative literals for different values of $k=1,2$. They also included a logic phase to generate the desired number of negative literals in the non-systematic structure. Based on the successful performance of r2SAT in generating diverse neuron states and global minima solutions, it is evident that a dynamic distribution of negative literals contributes to producing diversified final neuron states. However, the significance and rationale behind using these negative literals within the logical rule remains unclear. Therefore, this study aims to analyze the impact of restricting the possible combinations in solving logic satisfiability using DHNN.

We hypothesize that imposing this restriction will result in higher global solutions compared to the existing non-systematic logical rules. Thus, this study aims to investigate the impact of including at least one negative literal in each of the second-order clauses while no restrictions will be imposed on first-order clauses. By exploring this aspect, we can gain a better understanding of the importance of negative literals within the logical rule. Further exploration in this area could provide valuable insights for developing more efficient logical structures within logic satisfiability in DHNN.

2.3. Inadequate metrics for evaluating the effectiveness of activation functions in DHNN

The selection of the activation function plays a crucial role in neural networks as it has a significant impact on the network's ability to learn complex patterns and make accurate predictions [24]. Over the years, numerous activation functions have been investigated in the field of neural networks. Dubey et al. [12] conducted a survey that presents advancements in activation functions within neural networks. In DHNN logic satisfiability, an activation function acts as a mathematical operation that determines a final neuron state based on its initial input. Obtaining an optimal retrieved final neuron state is significant since the quality of the final neuron state reflects the nature of the logical rule in DHNN. The study conducted by Mansor and Sathasivam [16] demonstrates the successful performance of the Hyperbolic tangent activation function in DHNN. However, there is limited information on how errors are interpreted and how the final neuron state influences the overall performance in DHNN. This highlights a potential research opportunity to enhance the updating rule in DHNN by implementing an optimal activation function. Such an enhancement could lead to increased diversity and variations in the final neuron states, ultimately improving both effectiveness and efficiency of this neural network model in retrieving diversified final neuron states. According to [17], expanding the diversity of the final neuron state improves the ability of DHNN to identify more global solutions in various solution spaces. This urges the study to propose an optimal activation function to enhance the updating rule in the DHNN, with the aim of guaranteeing that the DHNN retrieves a more varied and diversified range of final neuron states. Therefore, this study aims to emphasize the importance of utilizing an optimal activation function to improve the effectiveness of updating rule in DHNN.

3. CRAN2SAT representation

CRAN2SAT is a logical representation that consists of first order and second order logic with a specific condition. The proposed logic will exclude both positive literals in the second-order clauses and does not impose any restrictions on the first-order clauses. It's worth noting that the symbol c within CRAN2SAT is used as a versatile representation of "conditional", while also serving the purpose of distinguishing it from the conventional RAN2SAT with the newly introduced logical rule. The basic components of proposed CRAN2SAT are as follows:

(a) A set of m second order clauses represent as $C_1^{(2)*}, C_2^{(2)*}, \dots, C_m^{(2)*}$, where $C_m^{(2)*} = (B_i^* \vee D_i^*)$, $m \in \mathbb{Z}^+$.

(b) A set of n first order clauses represent as $C_1^{(1)}, C_2^{(1)}, \dots, C_n^{(1)}$, where $C_n^{(1)} = (A_i)$, $n \in \mathbb{Z}^+$.

The $C_m^{(2)*}$ and $C_n^{(1)}$ represents the second and first order clauses, where a set of literals can be either positive or negative literal such that

$$B_i^* \in \{B_i^*, -B_i^*\}, D_i^* \in \{D_i^*, -D_i^*\} \text{ and } A_i \in \{A_i, -A_i\}.$$

Notably, i denotes the number of independent literals within the clauses. It's worth noting that the primary distinction between CRAN2SAT and RAN2SAT lies within the components (a). The logical structure of RAN2SAT [8], will be fully utilized c possible combination of clauses ($c=2^k$), where k is the order of the clause. The first order clause, $C_i^{(1)}$ will have two possible combinations of clauses as in Eq (1). While the second order clause, $C_i^{(2)}$ will have four possible combinations of clauses as in Eq (2). However, the logical structure of CRAN2SAT will exclude both positive literals in the second-order clauses. In other words there are only three possible combinations of second order clauses. Due to this condition, the second order clause of CRAN2SAT is structured based on $C_i^{(2)*}$ as in Eq (3)

$$C_i^{(1)} = \{A_i, \neg A_i\}, \quad (1)$$

$$C_i^{(2)} \in \{(B_i \vee D_i), (\neg B_i \vee D_i), (B_i \vee \neg D_i), (\neg B_i \vee \neg D_i)\}, \quad (2)$$

$$C_i^{(2)*} \in \{(\neg B_i^* \vee D_i^*), (B_i^* \vee \neg D_i^*), (\neg B_i^* \vee \neg D_i^*)\}, \quad B_i^* \vee D_i^* \notin C_i^{(2)*}. \quad (3)$$

All the clauses are connected by logical AND (\wedge) and literals within each clause are connected by logical OR (\vee). Therefore, by using the components in (a)–(d), the general formulation of CRAN2SAT or $P_{CRAN2SAT}$ and the definition of the clause in $P_{CRAN2SAT}$ is in Eq (5), where m represents the total count of clauses with two literals, while the variable n represents the total count of clauses with one literal

$$P_{CRAN2SAT} = \wedge_{i=1}^m C_i^{(2)*} \wedge_{i=1}^n C_i^{(1)}, \quad (4)$$

$$C_i^{(k)} = \begin{cases} (B_i^* \vee D_i^*), & k=2, \\ A_i, & k=1. \end{cases} \quad (5)$$

From Table 1, there will be only 3 possible combinations for the second order clause (refer Eq (3)), which means the combination of the second order clauses can be appear repeatedly as the number of the second order clause m increases (maybe the clauses will appear consistently same). The main criteria that makes the proposed CRAN2SAT differ from r2SAT is that even though r2SAT controls the proportion of the negative literal in the clause, there is still a possibility of all the combinations in Eq (2) being generated. The bipolar value that holds value of one and negative one represents TRUE and FALSE, respectively. If all the clauses in $P_{CRAN2SAT}$ are satisfied, the logical rule of $P_{CRAN2SAT} = 1$. In other words, the clauses in $P_{CRAN2SAT}$ are not satisfied and the logical rule of $P_{CRAN2SAT} = -1$. In this paper, a random distribution of $C_i^{(1)}$ and $C_i^{(2)*}$ will be embedded in DHNN. According to Eqs (4) and (5), the possible structure of $P_{CRAN2SAT}$ can be represented as in Table 1.

Table 1. Possible structure of $P_{CRAN2SAT}$.

NN	$m=n$	Possible $P_{CRAN2SAT}$
3	1	$P_{CRAN2SAT} = (B_1 \vee \neg D_1) \wedge \neg A_1$
		$P_{CRAN2SAT} = (\neg B_1 \vee \neg D_1) \wedge A_1$
		$P_{CRAN2SAT} = (B_1 \vee \neg D_1) \wedge (\neg B_2 \vee \neg D_2) \wedge \neg A_1 \wedge A_2$
6	2	$P_{CRAN2SAT} = (\neg B_1 \vee D_1) \wedge (\neg B_2 \vee D_2) \wedge A_1 \wedge A_2$
		$P_{CRAN2SAT} = (\neg B_1 \vee \neg D_1) \wedge (\neg B_2 \vee D_2) \wedge \neg A_1 \wedge \neg A_2$
		$P_{CRAN2SAT} = (\neg B_1 \vee \neg D_1) \wedge (\neg B_2 \vee D_2) \wedge (\neg B_3 \vee \neg D_3) \wedge \neg A_1 \wedge A_2 \wedge A_3$
9	3	$P_{CRAN2SAT} = (\neg B_1 \vee D_1) \wedge (\neg B_2 \vee \neg D_2) \wedge (B_3 \vee \neg D_3) \wedge A_1 \wedge \neg A_2 \wedge \neg A_3$
		$P_{CRAN2SAT} = (\neg B_1 \vee \neg D_1) \wedge (B_2 \vee \neg D_2) \wedge (\neg B_3 \vee D_3) \wedge \neg A_1 \wedge \neg A_2 \wedge \neg A_3$

4. CRAN2SAT in DHNN

The DHNN is a subset of ANN. It comprises N interconnected neurons with no hidden layers. One notable feature of DHNN is its CAM, which allows it to store patterns related to the problem. DHNN contains a limited number of neurons in bipolar form, denoted as $S_i = \{-1, 1\}$ for $i \in \mathbb{N}$, which are interpreted as true and false respectively. The general formulation of updating the neuron state with predefined threshold value is as follows:

$$S_i = \begin{cases} -1, & \text{if } \sum_j^N W_{ij} S_j \geq \theta, \\ 1, & \text{otherwise.} \end{cases} \quad (6)$$

Where S_i and S_j are states of the i th and j th neuron, respectively, W_{ij} is the synaptic weight between i th and j th neuron, with q as the predefined threshold value that is set to $\theta = 0$. This is to ensure the energy of DHNN is monotonically decreases [3]. There are two characteristics for the synaptic weight in DHNN. Initially, in the DHNN there is an absence of self-connection among the neurons suggesting that the diagonal of synaptic weight is zero, where $W_{ii} = W_{jj} = 0$. Second, the synaptic weight within DHNN consistently exhibits symmetry, denoted by $W_{ij} = W_{ji}$. In this research paper, the logic of $P_{CRAN2SAT}$ is incorporated into DHNN by associating each variable of $P_{CRAN2SAT}$ with the corresponding neuron state.

The process of logic satisfiability in DHNN encompasses two parts, generally referred to as the learning phase and the retrieval phase. During the learning phase, the primary goal is to reduce the inconsistency of $P_{CRAN2SAT}$, which corresponds to the minimized cost function, $E_{P_{CRAN2SAT}}$ defined as:

$$E_{P_{CRAN2SAT}} = \sum_{i=1}^{NC} \prod_{j=1}^{m+n} Q_{ij}, \quad (7)$$

where NC and $m+n$ refer to the number of clauses and number of variables in $P_{CRAN2SAT}$, respectively. The inconsistency of $P_{CRAN2SAT}$ is defined by taking the negation of $P_{CRAN2SAT}$, denoted as Q_{ij} , which is defined as

$$Q_{ij} = \begin{cases} \frac{1}{2}(1 - S_X), & \text{if } \neg X, \\ \frac{1}{2}(1 + S_X), & \text{if } X. \end{cases} \quad (8)$$

S_X denotes the neuron state in the clause with X and $\neg X$ as a positive and negative literal respectively. X consists of arbitrary literals of $\{A_i, B_i, C_i\}$. In the general equation of the logical structure $P_{CRAN2SAT}$ represented by Eqs (4) and (5), the conjunction (\vee) signifies multiplication while the disjunction (\wedge) represents addition in the cost function. The minimized cost function is identified by the utmost number of clauses that have been satisfied. The value of $E_{P_{CRAN2SAT}}$ has a proportional relationship with the number of clauses in $P_{CRAN2SAT}$ that are not satisfied. When the number of unsatisfied clauses increase, it will result in a corresponding increase in the value of $E_{P_{CRAN2SAT}}$. Based on Eq (7), if $E_{P_{CRAN2SAT}} = 0$, this indicates that the logical inconsistency of $P_{CRAN2SAT}$ has been minimized. In other words, $E_{P_{CRAN2SAT}} = 0$ shows that all the clauses in $P_{CRAN2SAT}$ are satisfied.

According to Abdullah's method [1], the value of synaptic weight W_{ij} is determined by comparing the coefficient of the cost function, $E_{P_{CRAN2SAT}}$ with the Lyapunov energy function denoted as $H_{P_{CRAN2SAT}}$. The Lyapunov energy function is defined as

$$H_{P_{CRAN2SAT}} = -\frac{1}{2} \sum_i^N \sum_j^N W_{ij} S_i S_j - \sum_i^N W_i S_i. \quad (9)$$

The Lyapunov energy function in DHNN is formulated to describe the energy value of the network. When this energy reaches its global minimum, the network achieves a stable state [25]. The DHNN model aims to ensure that solutions move towards the global minimum energy, which corresponds to the stable state of the neurons. However, it is worth mentioning that finding the global minimum energy depends on the efficiency of both learning and retrieval phases in DHNN. If these phases are inefficient, DHNN may end up trapped at a local minimum energy or suboptimal solution. Then, all the value of W_{ij} will be stored in a matrix form as a CAM. During the retrieval phase, the final neuron state generated by the DHNN is assessed to determine whether the solution achieved is global or local. During this phase, the retrieved neuron state is asynchronously updated by applying the W_{ij} values stored in the CAM. Consequently, through the utilization of the stored W_{ij} in CAM, the DHNN's local field is computed in the subsequent manner, where local field is denoted as h_i ,

$$h_i = \sum_{j \neq i}^N W_{ij} S_j + W_i. \quad (10)$$

Once the value of h_i is obtained, the final neuron state is retrieved by squashing the value of h_i by using an activation function. The final neuron state $S_i(t)$ in DHNN will be updated by

$$S_i(t) = \begin{cases} 1, & \text{if } g(h_i) \geq 0, \\ -1, & \text{if } g(h_i) < 0, \end{cases} \quad (11)$$

where $g(h_i)$ is obtained according to the types of the activation function. Currently, the HTAF is used to transform the value of h_i , to be either one or negative one. However, this paper proposes a new non-monotonic Smish activation function to enhance the efficiency of the updating rule within DHNN. Additionally, the DHNN-CRAN2SAT with proposed Smish activation function will be compared with other types of activation functions, namely, McCulloch-Pitts, piecewise linear activation function, Elliot activation function, HTAF and Swish activation function. A detailed explanation regarding the proposed Smish activation functions is discussed in section five.

Finally, the quality of the retrieved neuron state is assessed using the equation provided below:

$$\left| H_{P_{CRAN2SAT}} - H_{P_{CRAN2SAT}}^{\min} \right| \leq Tol, \quad (12)$$

where $Tol=0.001$ is a predetermined tolerance value [3]. The $H_{P_{CRAN2SAT}}$ is the final energy associated with the final neuron state $S_i(t)$, and $H_{P_{CRAN2SAT}}^{\min}$ are two absolute minimum energies computed by

$$H_{P_{CRAN2SAT}}^{\min} = -\left(\frac{m+2n}{4} \right), \quad (13)$$

where m and n represent the number of second and first order clauses existing in $P_{CRAN2SAT}$, respectively. The final energy $H_{P_{CRAN2SAT}}$ is computed by using Eq (9). According to [25], the Lyapunov energy function plays a crucial role in determining the level of convergence within the network. The energy value obtained can be categorized as either the global minimum or a local minimum.

Hence, if Eq (12) is fulfilled, the resulting neuron state will be regarded as a global minimum energy. Otherwise, the retrieved neuron state will be confined to a local minimum energy.

5. Proposed non-monotonic Smish activation function in DHNN

This section will introduce the non-monotonic Smish activation function in solving logic satisfiability in DHNN. This, its characteristics, and properties will be discussed further. The activation function works as a transfer function to transform the output into the bipolar form. The final neuron state that is retrieved by DHNN will demonstrate the nature of the model. The purpose of introducing the non-monotonic Smish activation function in DHNN is to enhance the effectiveness of DHNN in retrieving a diversified final neuron state. Notably, this is the first attempt of applying the Smish activation function in solving logic satisfiability in DHNN. Smish activation function is a new type of nonlinear activation function and was proposed by [26]. Figure 1 illustrates the figure for the Smish activation function and its derivative.

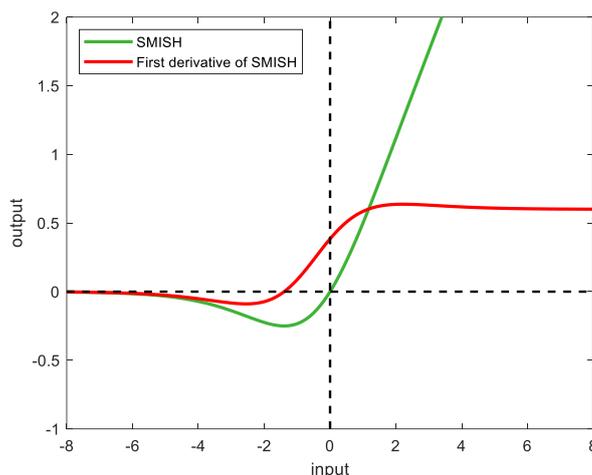


Figure 1. Illustration of Smish activation function and its derivative.

The Smish activation function is expressed as follows

$$g(h_i) = \alpha h_i \cdot \tanh \left[\ln \left(1 + \text{sigmoid}(\beta h_i) \right) \right] \quad (14)$$

with

$$\text{sigmoid}(\beta h_i) = \frac{1}{1 + e^{-\beta h_i}}. \quad (15)$$

The Smish activation function is a nonlinear activation function that exhibits a graph that is not linear. The output of the Smish activation function ranges between $[-0.25, \infty)$. In terms of its output range, the Smish activation function has a lower bound but no upper bound. In Figure 1, it can be observed that the Smish activation function is continuous without any discontinuity in its output range. The Smish activation function is considered as a learning based adaptive activation function because it contains learnable parameters value α and β . If compared to the other types of activation functions, there is no doubt that the Smish activation function is more complex in terms of the function itself. According to [26], the $\text{sigmoid}(\beta h_i)$ in Eq (14) is used to reduce the range of values local field, h_i . Furthermore, the logarithmic operation is incorporated to achieve a smoothly transitioned curve and a consistent trend. In the perspective of solving logic satisfiability in DHNN, Smish then multiplies its tanh operation by the value of the local field, h_i simultaneously. Therefore, this demonstrates the ability to regularize negative output. Due to these properties, the Smish activation function can maintain partial sparsity and a regularization effect for negative inputs. As a result, the positive values will serve as a straightforward linear representation. Since the Smish activation function is reported as a smooth and non-monotonic function, it will help to improve the ability of the network convergences. This makes the Smish activation function compatible with the Lyapunov energy function in DHNN, as the Lyapunov energy function plays a pivotal role in monitoring the convergence of DHNN [27].

According to [28], an ideal activation function is anticipated to possess certain characteristics. The function should exhibit continuity and differentiability across all points, with the derivative remaining unsaturated to achieve optimal performance. The issue of vanishing gradients can occur when using a saturated activation function, which can impact inputs that are highly positive or highly negative. According to [12], the issue of a vanishing gradient occurs when the gradient of an objective

function with respect to a parameter approaches zero, which results in minimal parameter updates during network training. Thus, the vanishing gradient problem will cause the output value to have a tendency to move toward zero. As a result, extremely small derivatives have the potential to interrupt the learning process. Therefore, to suit with the expected features, the Smish activation function is a non-zero derivative at zero (refer to Figure 1). This eventually will help to overcome the problem of vanishing gradients because it is partially saturated toward, $h_i \rightarrow (-\infty)$. The properties of continuously differentiable are preferable in good activation functions. The undefined gradient at the midpoint of a non-continuously differentiable activation function can negatively impact training performance [29]. Maintaining a differentiable property helps avoid singularities [30] and impacts the convergence speed of the network [31].

6. Experimental setup

This section outlines the experimentation process that has been developed to assess the efficacy of the proposed model. It includes an explanation of the simulation platform, what is the existing and benchmark of the activation function, how parameters and activation functions are assigned, the performance metrics that will be utilized and the baseline methods.

6.1. Simulation platform

The proposed model DHNN-CRAN2SAT with Smish activation function is implemented in DEV C++, version 5.11 with a specification of Intel Core i7 processor with 8 GB RAM in the Windows 10 operating system. These simulations were running out by using the same device to avoid biases. The simulated datasets are generated by randomly assigning bipolar representations $\{-1,1\}$ to the neuron state. In this study, a simulated dataset will be used to analyze the performance of different approaches. First, the proposed DHNN-CRAN2SAT is compared with the existing non-systematic logical rule (DHNN-CRAN2SAT, DHNN-r2SAT and DHNN-YRAN2SAT). Then, DHNN-CRAN2SAT with a proposed non-monotonic Smish activation function is compared with other types of activation functions, such as McCulloch-Pitts, Piecewise Linear activation function, Elliot symmetric activation function, HTAF and Swish activation function. Finally, the capabilities of Smish activation functions will be conducted within the existing non-systematic logic in DHNN. Figure 2 explains each configuration of DHNN-CRAN2SAT.

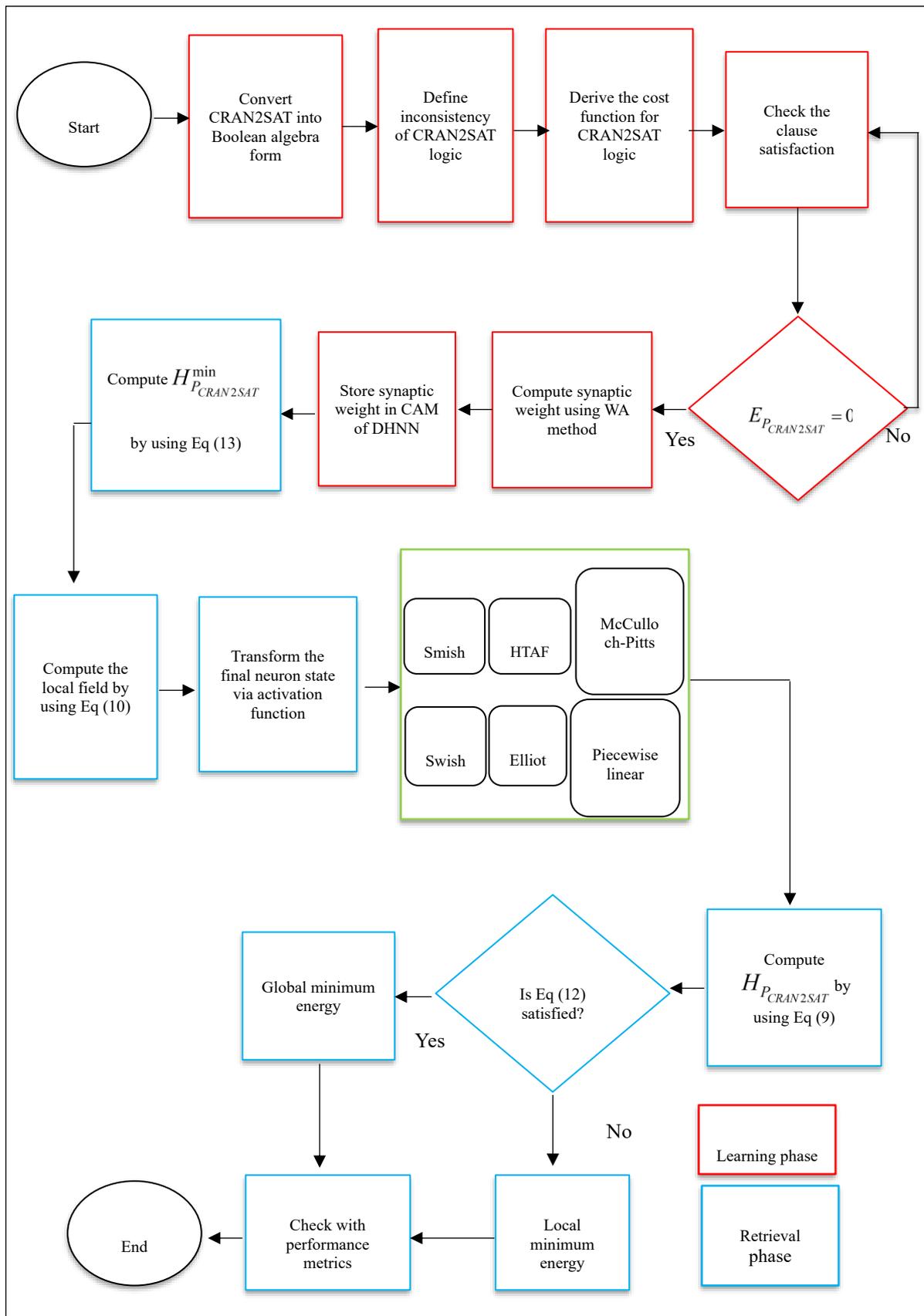


Figure 2. Flowchart of DHNN-CRAN2SAT with various activation functions.

6.2. Existing and benchmark activation functions

The final neuron state $S_i(t)$ in DHNN will be updated by using Eq (11), where $g(h_i)$ is obtained according to the types of the activation function. In this study, there are five benchmark activation functions that will be compared with the proposed Smish activation function. Currently, the HTAF is used to transform the value of the local field h_i to bipolar form of $\{-1,1\}$ [19,32,33]. The choice of the benchmark activation function is chosen based on two criterions. The first criterion is based on the existing activation function that has been applied in solving logic satisfiability in DHNN, such as McCulloch-Pitts [1], Elliot symmetric and hyperbolic tangent [16]. The second criteria considers the activation function that has never been applied in solving logic satisfiability in DHNN, such as Piecewise linear and Swish activation function. The second criterion is specifically chosen based on the characteristic of the activation function. The Swish activation function has the same characteristics and non-monotonic activation function as Smish, while the piecewise linear function is composed of multiple linear segments, each defined over a specific interval. Activation functions play an important role in the neural network, where it is a mathematical function that determines the final neuron state for the given input neuron. The quality of a final neuron state that is retrieved by DHNN will demonstrate the nature of the model. The existing activation function that has been previously applied in DHNN has several issues. The details of the existing and benchmark activation are discussed in the rest of this section.

6.2.1. McCulloch-Pitts

The McCulloch-Pitts function is the conventional platform that has been used in logic satisfiability in DHNN. This function was popularized by [1] and it is reported that the solution will trap in the local minimum of the energy [34]. The output of the McCulloch-Pitts does not have limitations (unbounded) because the McCulloch-Pitts function is fully based on the value of local field, h_i . Therefore, the updating neuron state will be directly transformed by using Eq (11)

$$g(h_i) = h_i. \quad (16)$$

6.2.2. Elliot symmetric activation function

The Elliot symmetric activation function is a computationally efficient approximation of the hyperbolic tangent, as it maps the output to a range between $(-1,1)$ [35]. The Elliot symmetric activation function is expressed below:

$$g(h_i) = \frac{h_i}{1 + |h_i|}. \quad (17)$$

The Elliot symmetric activation function has been applied in doing logic programming in DHNN for 3SAT clauses [16]. However, the hyperbolic tangent activation function demonstrates superior performance when compared to the Elliot symmetric activation function. In addition, it is also used for extreme learning machines for industrial drying process regressions [34], and the result shows that Elliot obtained the best performances in most cases.

6.2.3. HTAF

HTAF produces values in the range of $(-1,1)$ and it is symmetric about the origin. The formulation of HTAF is given as follows

$$g(h_i) = \tanh(h_i) = \frac{e^{h_i} - e^{-h_i}}{e^{h_i} + e^{-h_i}}. \quad (18)$$

The properties of HTAF are

$$\tanh(0) = 0, \quad \tanh(h_i) \in (-1,1), \quad \forall h_i \in \mathbb{R} \quad \text{and} \quad \tanh(-h_i) = -\tanh(h_i).$$

Furthermore, $\tanh(h_i) \rightarrow 1$ as $h_i \rightarrow \infty$ and $\tanh(h_i) \rightarrow -1$ as $h_i \rightarrow -\infty$ and it is strictly increasing on \mathbb{R} . Due to $\tanh(h_i) \in (-1,1)$, this indicates the output range of HTAF is between $(-1,1)$. A general problem with the HTAF is that this function is saturated and suffers from the vanishing gradient problem because of its bounded limits [36]. Based on the above properties, if the value of the local field h_i becomes larger, it will approximate the tangent function to be one while small values of the local field h_i will approximate the tangent function to be negative one. Moreover, the functions are only sensitive to changes around the midpoint of their input, such as when the input is zero. Once saturated, it becomes challenging for the learning algorithm to continue to adapt the weights to improve the performance of the model [37]. During the retrieval phase, the final neuron state is updated by using Eq (11). Based on this equation, the final neuron state will be transformed to one if $g(h_i) \geq 0$. Due to this reason, the tangent function will contribute to the low diversified final neuron states. This occurs since the tangent function is only responsive to changes around the midpoint of a zero input. Beyond this point, it typically approaches either one or negative one.

6.2.4. Piecewise linear activation function

A piecewise linear function is a type of function that comprises a limited number of linear segments, each characterized over an equally sized interval. This is an example of a piecewise linear activation function:

$$g(h_i) = \begin{cases} -1, & \text{if } h_i \leq -1, \\ h_i, & \text{if } -1 < h_i \leq 1, \\ 1, & \text{if } h_i > 1. \end{cases} \quad (19)$$

The key feature of piecewise linear function is the lack of curvature within each interval defined by its breakpoint, which results in a constant first-order derivative. The piecewise linear function has a lack in terms of the transition points, which might introduce non-smoothness in the function. Therefore, it will affect the lack of differentiability at the transition points [28].

6.2.5. Swish activation function

The Swish activation function proposed by [38] is obtained by multiplying the input and sigmoid function. The hybrid activation function is defined as follows

$$g(h_i) = h_i \cdot \text{sigmoid}(\beta h_i) = \frac{h_i}{1 + e^{-\beta h_i}}. \quad (20)$$

Swish is a smooth, non-monotonic activation function that lacks and doesn't have any upper bound. It is bounded below and unbounded above. Swish is partially saturated toward, $h_i \rightarrow (-\infty)$ and, thus, Swish does not suffer vanishing gradient problems. The distinguishing characteristic of the Swish function from other activation functions is its non-monotonic properties. The Swish activation function with $\beta=1$ was introduced by [39]. Based on the parameter value of β , the shape of the Swish activation function is adjusted between the linear and ReLU functions. The smaller and higher values of β lead toward the linear and ReLU functions, respectively [12]. As mentioned in the previous section, a good activation function should exhibit continuity and differentiability across its entire range, with the additional condition that its derivative remains unsaturated. The vanishing gradient problem will cause the output value to have a tendency to move toward zero [12]. Out of the five-benchmark activation function, only Swish does not have a vanishing gradient problem. Figure 3 illustrates the graph of the benchmark activation function and its derivative.

Table 2 summarizes properties of all activation functions and Table 3 lists all the functions for the benchmark of activation functions.

Table 2. Summary of the properties of activation functions.

Types of activation functions	Monotonic	Bounded	Diminish gradient	Output range
McCulloch-Pitts	Yes	Yes	Yes	$[-\infty, +\infty)$
Piecewise linear	Yes	Yes	Yes	$(-1, 1)$
Eliot symmetric	Yes	Yes	Yes	$(-1, 1)$
HTAF	Yes	Yes	Yes	$(-1, 1)$
Swish	No	No	No	$[-\infty, +\infty)$
Smish	No	No	No	$[-0.25, \infty)$

Table 3. List of activation function in DHNN.

Activation function	Function
McCulloch-Pitts	$g(h_i) = h_i$
Piecewise linear	$g(h_i) = \begin{cases} -1, & \text{if } h_i \leq -1, \\ h_i, & \text{if } -1 < h_i \leq 1, \\ 1, & \text{if } h_i > 1. \end{cases}$
Elliot symmetric	$g(h_i) = \frac{h_i}{1 + h_i }$
Hyperbolic tangent	$g(h_i) = \tanh(h_i) = \frac{e^{h_i} - e^{-h_i}}{e^{h_i} + e^{-h_i}}$
Swish	$g(h_i) = h_i \cdot \text{sigmoid}(\beta h_i) = \frac{h_i}{1 + e^{-\beta h_i}}$

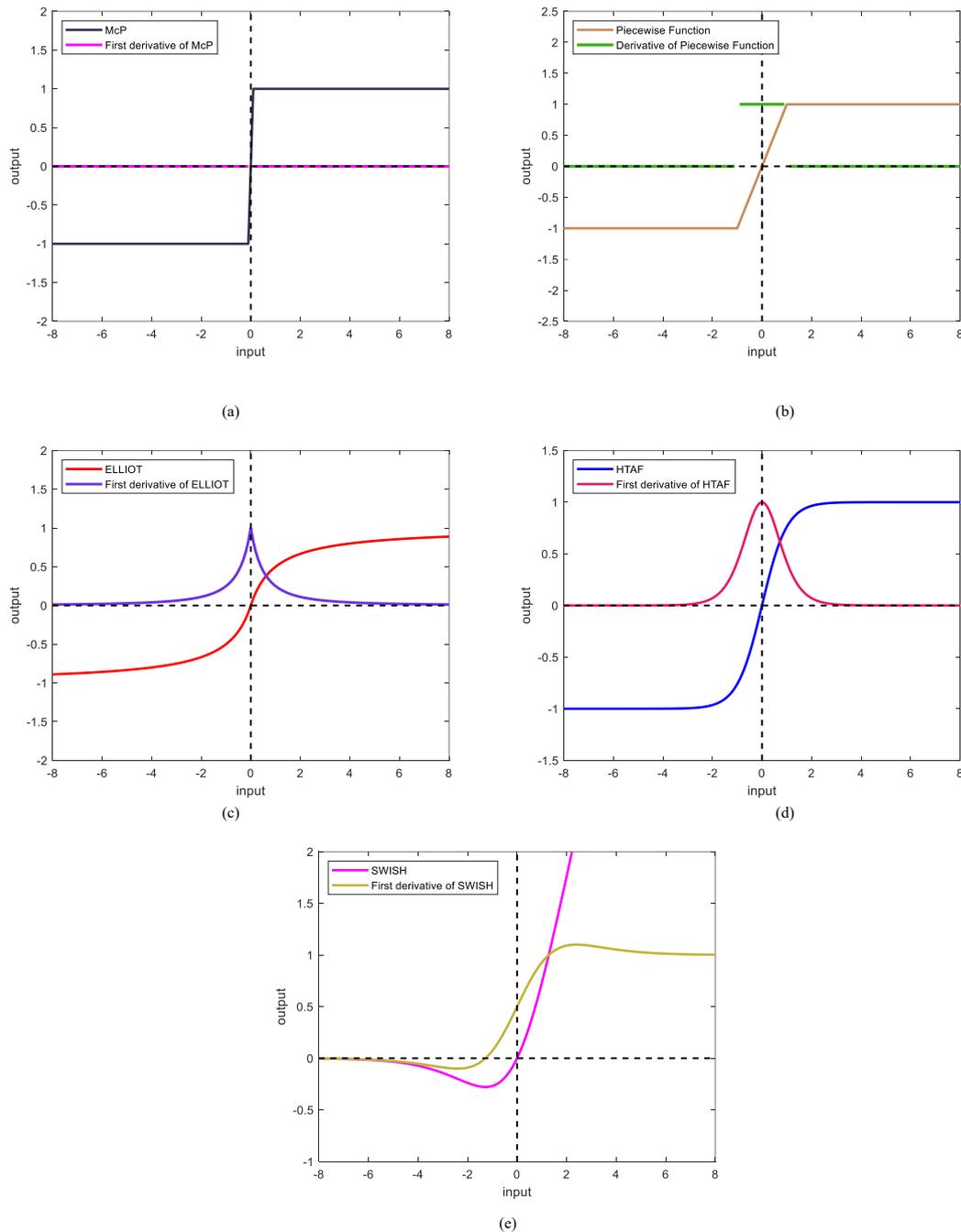


Figure 3. Illustration of activation functions and its derivative: (a) McCulloch-Pitts; (b) piecewise linear activation function; (c) Elliot symmetric activation function; (d) HTAF; (e) Swish activation function.

6.3. Parameter and activation function setup in DHNN

The evaluation of the proposed DHNN-CRAN2SAT model will be conducted based on its performance during the learning and retrieval phases. The exhaustive search (ES) technique that is employed during the learning phase is to validate clause satisfaction. The characteristic of ES is based on a “trial and error” method. Therefore, the learning will be terminated upon achieving the target of

clause satisfaction or upon reaching the maximum count of learning iterations denoted as NH . Followed by the implementation of the proposed non-monotonic Smish activation function due to access of the quality of the retrieved neuron state during both the learning and retrieval phases, the initialization of neuron states is carried out through random generalization. This approach guarantees an equal and unbiased representation of the generated solutions. The parameter value involved in the Swish and Smish activation function is chosen based on the current research by [26,38], respectively. The parameters employed for the simulation were listed in Tables 4 and 5.

Table 4. General settings of CRAN2SAT with DHNN in all presented models.

Setting	Parameter value
Number of neurons (NN)	$3 \leq NN \leq 45$ [10]
Neuron combination (C)	100
Number of trials (NT)	100
Number of learning (NH)	100
Synaptic weight method	Abdullah method [1]
Threshold Central Processing Unit time	24 hours
Tolerance value, Tol	0.001 [3]
Initialization of neuron states	Random
Learning algorithm	Exhaustive search [3]
Activation function	McP, piecewise linear, Elliot, HTAF, Swish, Smish

Table 5. List of parameters of activation function in DHNN.

Activation function	Parameter value
Swish	$\beta = 1$ [38]
Smish	$\alpha = 1$ and $\beta = 1$ [26]

6.4. Performance metrics

This section will explain how the performance of the DHNN-CRAN2SAT model is evaluated with the other logical rule. The performance of the DHNN-CRAN2SAT model will be evaluated by using error analysis (learning phase and retrieval phase) and similarity analysis. Since we also proposed DHNN-CRAN2SAT with non-monotonic Smish activation function, the capability of the activation function is evaluated based on the testing error analysis and similarity analysis. Therefore, the purpose of the performance metrics is listed below:

- Learning phase: To minimize the cost function in Eq (7), by searching for a satisfying interpretation of DHNN-CRAN2SAT.
- Retrieval phase: To access the quality of the retrieved neuron states (determine the solutions are global minima solutions or local minima solutions); involves the computation of Eq (12).
- Similarity analysis: To examine the variation in the final neuron states generated by the DHNN model as compared to benchmark neuron states.

6.4.1. Learning phase

Throughout the learning process, the objective is to examine the efficiency of DHNN-CRAN2SAT in minimizing the cost function, as indicated in Eq (7). In the learning phase, the root mean square error (RMSE) and mean absolute percentage error (MAPE) are utilized to measure the capability of DHNN-CRAN2SAT in getting satisfied interpretations. This metric has been commonly employed in numerous studies, such as in [40,41]. Therefore, to evaluate the efficacy of DHNN-CRAN2SAT throughout the learning process, Eqs (21) and (22) are introduced as follows

$$RMSE_{Learning} = \sum_{i=1}^{NH} \sqrt{\frac{(f_{NC} - f_i)^2}{NH}}, \quad (21)$$

$$MAPE_{Learning} = \sum_{i=1}^{NH} \frac{100}{NH} \frac{|f_{NC} - f_i|}{|f_i|}, \quad (22)$$

where f_{NC} represents the maximum fitness and f_i signifies the current fitness. The best fitness is related to the number of clauses that are satisfied. Note that NC represents the count of clauses within the logical rule. Moreover, the DHNN-CRAN2SAT is considered optimal when the errors in Eqs (21) and (22) approach a zero value.

6.4.2. Retrieval phase

In the retrieval phase, the assessment of the solution's quality generated by DHNN-CRAN2SAT relies on the synaptic weights generated throughout the learning phase. Inspired by [18], we utilize RMSE and MAPE to evaluate the accuracy of the retrieved neuron state. A zero value for testing errors signifies that the retrieval phases are optimal:

$$RMSE_{Testing} = \sum_{i=1}^{NT \cdot C} \sqrt{\frac{(NT_{Solution} - NG_{Solution})^2}{NT \cdot C}}, \quad (23)$$

$$MAE_{Testing} = \sum_{i=1}^{NT \cdot C} \frac{1}{NT \cdot C} |NT_{Solution} - NG_{Solution}|. \quad (24)$$

Furthermore, the energy error analysis is employed to access the final neuron state that converges toward the minimum energy of DHNN-CRAN2SAT. Once the local field value h_i is obtained according to Eq (10), the final neuron state is obtained by applying an activation function to squash the h_i value, as depicted in Eq (11).

In this study, a new non-monotonic Smish activation function has been introduced to enhance efficiency of the updating rule within DHNN. Therefore, once the retrieved final neuron state is converted to the bipolar form of $\{-1,1\}$, the final energy is computed by using Eq (9). The assessment of energy error is evaluated by using RMSE and MAPE as described in Eqs (25) and (26), such as in the work [40,42].

$$RMSE_{Energy} = \sum_{i=1}^{NT \cdot C} \sqrt{\frac{(H_{min} - H_f)^2}{NT \cdot C}}, \quad (25)$$

$$MAE_{Energy} = \sum_{i=1}^{NT \cdot C} \frac{1}{NT \cdot C} |H_{\min} - H_f|. \quad (26)$$

Additionally, the global minima ratio Z_m will demonstrate the capability of our model work with DHNN [3]. Throughout the evaluation stage, the final state of the neuron will be analyzed to determine whether it converges toward a global or local solution by using Eq (12). If the equation is satisfied, the solution is considered as a global solution. Otherwise, it will be considered as a local solution. The global minima ratio is obtained by using the following formula:

$$Z_m = \frac{1}{NT \cdot C} \sum_{i=1}^{NT \cdot C} NG_{Solution}. \quad (27)$$

Notably, the retrieval phase is evaluated differently based on the performance of DHNN-CRAN2SAT and DHNN-CRAN2SAT with Smish activation function.

6.4.3. Similarity analysis

The quality of the retrieved neuron states generated by DHNN-CRAN2SAT can be analyzed by using similarity analysis and total neuron variation. This similarity analysis was inspired by [27], in which the similarity of the neuron state was assessed by comparing the retrieved final neuron state with the benchmark neuron state of the DHNN-CRAN2SAT model. The comparison will be evaluated based on the individual neuron state. The benchmark neuron state, which is also known as an ideal neuron state, is defined as

$$S_i^{\max} = \begin{cases} 1, & \text{for } A, \\ -1, & \text{for } \neg A. \end{cases} \quad (28)$$

Here, A and $\neg A$ denote positive and negative literals, respectively, within every clause of the DHNN-CRAN2SAT model, S_i^{\max} is the benchmark neuron state of $P_{CRAN2SAT}$ and S_i represents the obtained final neuron state through the DHNN process. Note that the similarity index only evaluates the final neuron state that fulfills Eq (12), signifying that the final neuron state must attain global minimum solution. The formulation of the Jaccard similarity index is as follows

$$JSI = \frac{pp}{pp + pn + np}. \quad (29)$$

Table 6 describes the parameter value involved in computing the value of the Jaccard similarity index and the list of symbols used in the retrieval phase summarized in Table 7.

Table 6. List of parameters incorporated within the similarity index.

Parameter	Parameter remarks
pp	Number of (S_i^{\max}, S_i) where $S_i^{\max} = 1$ and $S_i = 1$
pn	Number of (S_i^{\max}, S_i) where $S_i^{\max} = 1$ and $S_i = -1$
np	Number of (S_i^{\max}, S_i) where $S_i^{\max} = -1$ and $S_i = 1$
nn	Number of (S_i^{\max}, S_i) where $S_i^{\max} = -1$ and $S_i = -1$

Table 7. List of symbols utilized in the experimental setup.

Parameter	Parameter's name
NH	Number of learning
NT	Number of trials
C	Number of neuron combination
$NT_{Solution}$	Number of total solutions
$NG_{Solution}$	Number of global minimum solution
H_{min}	Minimum energy achieved in DHNN-CRAN2SAT
H_f	Final energy achieved in DHNN-CRAN2SAT
S_i^{max}	Benchmark neuron state
S_i	Final neuron state
TV	Total neuron variation

The range of Jaccard similarity index is between (0,1), which indicates the lowest value of Jaccard similarity index and shows high dissimilarity between the S_i and S_i^{max} . Furthermore, the total neuron variation is formulated as in Eqs (30) and (31)

$$TV = \sum_{i=0}^{\omega} F_i, \quad (30)$$

$$F_i = \begin{cases} 1, & S_i \neq S_i^{max} \\ 0, & S_i = S_i^{max} \end{cases}. \quad (31)$$

Therefore, ω signifies the total string of the final neuron state generated by DHNN, which aligns with the global minimum solutions. Meanwhile, F_i acts as a scoring mechanism to calculate the count of distinct neuron states ($S_i \neq S_i^{max}$) throughout the retrieval phase.

6.5. Comparative analysis and baseline methods

This research will carry out two assessments in relation to other foundational techniques, with a focus on experimenting in the areas of:

- (a) The investigation on the proposed logic CRAN2SAT in DHNN will be compared with the non-systematic logic employing first order and second order logic. Therefore, in terms of evaluating the performance of the proposed logic, CRAN2SAT will be compared with all non-systematic logical rules that contain first order and second order logic. Notably, this study does not compare with S-type random k satisfiability by [17], because in their study, the value of synaptic weight is not computed by using the Abdullah method. The characteristics of the existing non-systematic logic are outlined as follows:
 - (i) RAN2SAT proposed by [8]. This is the first non-systematic logical rule that merges 1SAT and 2SAT logic, enabling a flexible and dynamic number of literals per clause. However, experimental results showed that first-order clauses introduced more logical inconsistencies than second-order clauses.

- (ii) r2SAT introduced by [10] is a non-systematic logic with weighted ratios of negative literals. In addition, they also encompass a logic phase that creates a logical structure in line with the preferred count of negative literals. As a result, r2SAT performs well in producing diverse neuron states and global minima solutions. In this study, the logical structure of r2SAT has been set to have at least 50% of negative literal due to its efficacy within the logic phase of r2SAT.
- (iii) YRAN2SAT was introduced by [9] and possesses a flexible hybrid logical structure that is mixed together both systematic and non-systematic structures. It offers random enumeration of first-order, second-order or both types of clauses. The study has implemented five possible pathways of YRAN2SAT that show an improved solution capacity.

The main difference between the proposed CRAN2SAT with the existing logic is that the existing logic fully utilizes all the possible combinations of 2SAT clauses as in Eq (2). However, the proposed logic CRAN2SAT will exclude both positive literals in the second-order clauses as in Eq (3). It's worth noting that the quantity of neurons falls within a certain range, $3 \leq NV \leq 45$, applied for all baseline models.

- (b) The evaluation of DHNN-CRAN2SAT with the proposed non-monotonic Smish activation function will be compared with another five-activation function, namely, McCulloch-Pitts, piecewise linear activation function, Elliot symmetric activation function, hyperbolic tangent activation function (HTAF) and Swish activation function. Note that the piecewise linear and Swish activation functions are the first attempts that apply in solving logic satisfiability in DHNN.
 - (i) The McCulloch-Pitts function is the conventional platform that has been used in logic satisfiability in DHNN by [1], but the solution will trap in the local minimum of the energy [43]. The output of the McCulloch-Pitts does not have limitations because the McCulloch-Pitts function is fully based on the value of local field h_i .
 - (ii) The Elliot symmetric and hyperbolic tangent activation functions have been incorporated in the execution of logic satisfiability in DHNN for 3SAT clauses [16]. Subsequently, the hyperbolic tangent activation function demonstrated superior efficacy in comparison to the Elliot symmetric activation function.

7. Results and discussion

The result and discussion will be separated into three parts. Initially, we will examine the effectiveness of the DHNN-CRAN2SAT model during both the learning and retrieval phases. In the first part of discussion, the final neuron state that was retrieved by the DHNN-CRAN2SAT model was examined by using HTAF. In the second part, this paper aimed to evaluate the impact of various activation functions during the retrieval phase of DHNN. For this reason, the DHNN-CRAN2SAT will embed the proposed non-monotonic Smish activation function. Thus, the DHNN-CRAN2SAT with Smish activation function will be compared with various activation functions, namely, McCulloch-Pitts, piecewise linear activation function, Elliot symmetric activation function, HTAF and Swish activation function. Finally, the full capability of different activation functions will be investigated throughout existing SAT logic in DHNN.

In Section 7.1, the capability of DHNN-CRAN2SAT in minimizing the cost function and generating an optimal synaptic weight will be examined. The effectiveness of DHNN-CRAN2SAT in

minimizing logical inconsistency was assessed through the application of performance metrics, namely, the RMSE and MAPE. Then, the behavior of DHNN-CRAN2SAT according to the value of synaptic weight leading to a global minima solution or local minima solution will be investigated. Moreover, the importance of the presence of negative literals in second-order clauses will be discussed. Meanwhile in Section 7.2, the retrieval phase aimed to demonstrate the impact of different activation functions in optimizing the final neuron state of DHNN. In order to evaluate this, the quality of the retrieved neuron states was analyzed using metrics such as RMSE, MAPE, and the ratio of the global minimum solutions.

Consequently, the quality of the global solutions retrieved by DHNN was measured through the similarity index and total neuron variation. Each of these sections concluded the most effective approaches, which were then used for comparative analysis with the current non-systematic logic in DHNN, Section 7.3. Concisely, as the number of neurons (NN) increase, the learning error in terms of $RMSE_{Learning}$ and $MAPE_{Learning}$ will increase. Figures 4 and 5 demonstrate the error of $RMSE_{Learning}$ and $MAPE_{Learning}$ for different state of the art SAT, respectively.

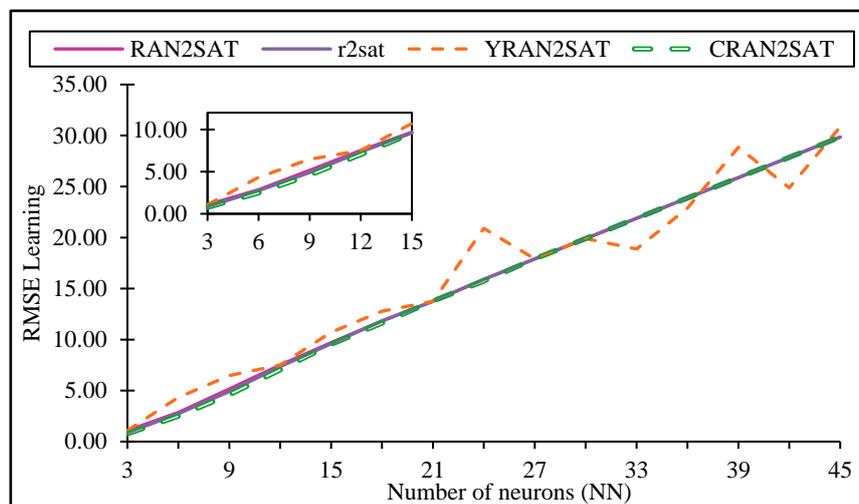


Figure 4. $RMSE_{Learning}$ for a different state of the art SAT.

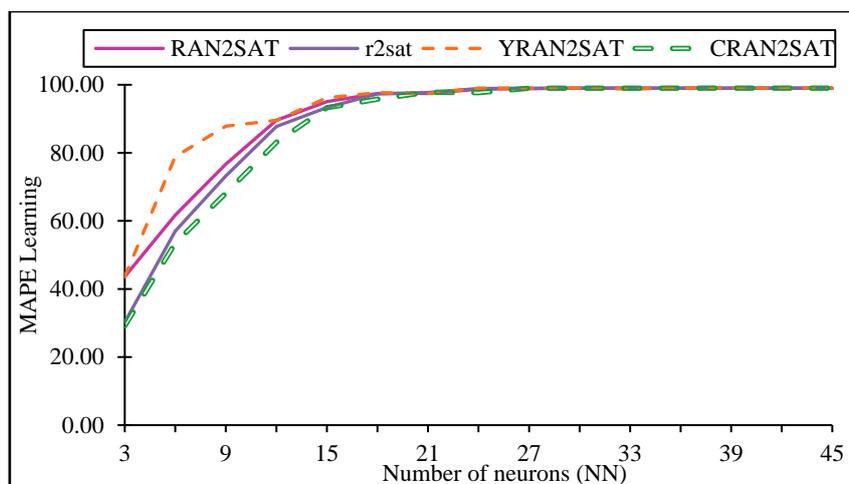


Figure 5. $MAPE_{Learning}$ for a different state of the art SAT.

Notably, all logical structure in the analysis contains the first order clause and second order clause, 1SAT and 2SAT, respectively. In this simulation, the logical structure of CRAN2SAT, RAN2SAT and r2SAT is emphasized based on the ratio of 1:1 for first order clause and second order clause. Additionally, the logical structure of r2SAT has been set to have at least 50% of negative literal [10]. Despite of having the same ratio for first order clause and second order clause, Figure 4 shows that the error of $RMSE_{Learning}$ for r2SAT is lower compared to the RAN2SAT. The logical structure of RAN2SAT differs from r2SAT since r2SAT's structure is regulated by a predetermined number of negative literals. Thus, by comparing $RMSE_{Learning}$ of RAN2SAT and r2SAT, we can infer that by having a higher proportion of negative literals in the logical structure, it will enhances the probability of obtaining a satisfied interpretation. This finding is fruitful for this research to address the significance of having negative literals in the logical structure.

As for the logical structure of YRAN2SAT, it has the capability to become both systematic and non-systematic logical rules. Therefore, the error of $RMSE_{Learning}$ for YRAN2SAT fluctuates because of the flexibility of the model that can be expressed as both systematic and non-systematic logical rules. From Figure 4, at $NN=33$ and $NN=42$, the error of $RMSE_{Learning}$ for YRAN2SAT will be at the lowest value because the structure of the YRAN2SAT is represented as systematic logic of 2SAT. However, at $NN=24$ and $NN=39$, the error of $RMSE_{Learning}$ for YRAN2SAT is higher because the logical rules are represented as systematic logic of 1SAT. One possible explanation for this outcome is that the DHNN encounters difficulty in completing its learning phase as the number of first order clauses increases. This can be attributed to a higher likelihood of obtaining unsatisfied clauses during this phase. Additionally, this situation will contribute to the randomly generated synaptic weight that can lead to a suboptimal retrieval phase. Supported by [42], as the quantity of first-order clauses enlarges, the DHNN encounters difficulty in completing the learning phase, subsequently leading to a suboptimal retrieval phase.

Based on result from the simulation, the proposed model CRAN2SAT has the smallest learning error of $RMSE_{Learning}$ and $MAPE_{Learning}$, compared to the other non-systematic SAT. The proposed CRAN2SAT model restricts the logical structure by excluding both positive literals in second-order clauses without imposing any restrictions on the first-order clauses. By incorporating this characteristic, our model will have a second order clause with at least one negative literal in each clause. Thus, the proposed CRAN2SAT model will have more negative literal compared to the r2SAT model. This is because of the percentage of negative literal for r2SAT is only 50% of the NN . In this simulation, the choice of $r = 0.5$ is made due to its effectiveness in the logic phase of the r2SAT [10]. Since the proposed model CRAN2SAT has the smallest learning error of $RMSE_{Learning}$ and $MAPE_{Learning}$, results in Figures 4 and 5 have successfully highlighted that having more negative literal in the second order clause will have a higher probability of getting satisfied interpretations. Therefore, the proposed CRAN2SAT model shows that the variety representation of non-systematic logical rule should emphasized on more negative literals of the second order clauses.

Even though CRAN2SAT demonstrates favorable performance, the incorporation of ES during the learning phase can impact the ability of DHNN-CRAN2SAT to achieve an ideal learning phase. Thus, it will lead to inconsistent learning interpretations. During this simulation, as $NN \geq 18$, the error of the logical structure will achieve maximum error, $MAPE_{Learning}=100$. In accordance with the discovery made by [41], it is possible to enhance the learning phase by integrating a learning algorithm into DHNN. This approach could aid in achieving an optimal satisfied interpretation for CRAN2SAT, consequently reducing the occurrence of logical inconsistencies. Regardless of comparing the proposed CRAN2SAT model with the existing non-systematic logical rule, we also make a comparison based on these different cases. This comparison is made to support our proposed CRAN2SAT model;

thus, our model incorporating the characteristic of excluding both positive literals in second-order clauses will result in an optimal learning phase. The CRAN2SAT model is compared with the logical structure that is comprise based on Cases 1–4 where:

Case 1: The logical structure excluding both negative literals in second-order clauses, which is denoted as $(\neg A \vee \neg B)$.

Case 2: The logical structure that comprises only positive literal in second-order clauses, which is denoted as $(A \vee B)$.

Case 3: The logical structure that comprises only negative literal in second-order clauses, which is denoted as $(\neg A \vee \neg B)$.

Case 4: The logical structure that comprises at most one positive literal per clause, which is also known as HORN2SAT.

Results in Figure 6 illustrate the error of MAPE by comparing the logical structure based on CRAN2SAT, Cases 1–4. Based on this result, the logical structure of Case 3 and proposed method CRAN2SAT obtain the lowest MAPE. Thus, this result supports our proposed logical structure, which excludes both positive literals in second-order clauses. This is due to the result obtained by Figure 6, which has appropriately highlighted the significance of having more negative literal in second order clauses and how it can lead to a high probability of getting satisfied interpretation. It is also preserved that logical structure that comprises only positive literal in second order (Case 2) generates high learning error. Despite having at most one positive literal per clause, the logical structure of Case 4 is still unable to minimize the logical inconsistencies. Hence, it is significant to say that negative literals play a crucial role in achieving satisfied clauses. Therefore, throughout this analysis, the proportion of negative literal in the second order clause is significant because it can help in getting higher satisfied interpretation.

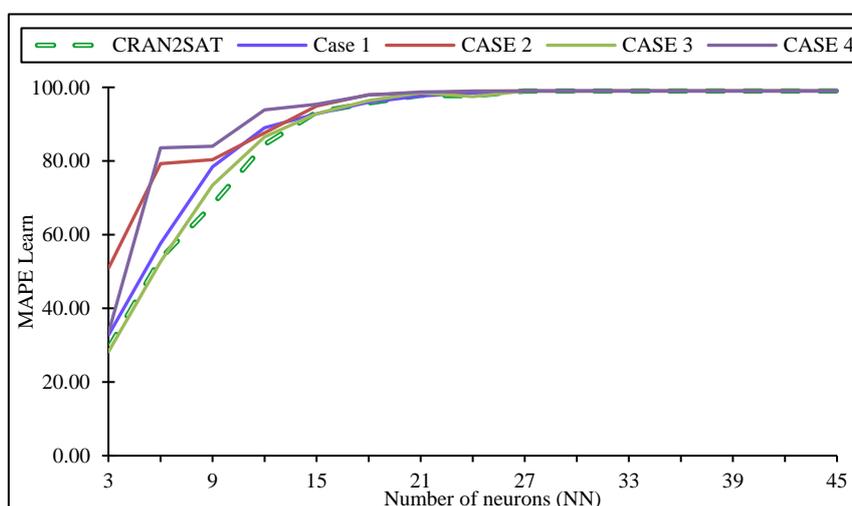


Figure 6. Comparison of $MAPE_{Learning}$ for between cases.

7.1. Retrieval phase

In this section, we will investigate the behavior of logic satisfiability in DHNN according to the value of synaptic weight, which leads to a global minima solution or local minima solution. In the learning process, an optimal synaptic weight is achieved that allows the DHNN-CRAN2SAT to reduce

logical inconsistencies. This section is divided into three segments: Analysis of testing errors, assessment of the final neuron state quality (including total neuron variation and similarity analysis) and analysis of synaptic weights.

7.1.1. Testing error analysis

Upon the completion of clause satisfaction checking (cost function minimization) during the learning phase of DHNN-CRAN2SAT, the synaptic weights are then generated using the Abdullah method [1]. The cost function that attains to a value of zero will lead to the retrieval of optimal synaptic weights during the testing phase, yielding a solution at the global minimum. Based on the results obtained during the learning process, the proposed CRAN2SAT model is able to generate an optimal synaptic weight during $3 \leq NN \leq 9$, leading a zero error for $RMSE_{Test}$ and $MAPE_{Test}$. During this interval, DHNN-CRAN2SAT is in optimal retrieval phase and demonstrates a strong capability in producing global minimum solutions. However, as NN increases, the testing error in terms of $RMSE_{Test}$ and $MAPE_{Test}$ will increase.

Figures 7 and 8 show the result of $RMSE_{Test}$ and $MAPE_{Test}$ for a different state of the art SAT, respectively. The result in the retrieval phase is interrelated with the achievement of the minimization logical inconsistencies during the learning phase. Thus, by referring to Figures 7 and 8, DHNN-CRAN2SAT obtained the lowest error in the retrieval phase due to the lowest error during the learning phase. Meanwhile, the testing error for DHNN-YRAN2SAT is not stable due to the different structure of the logic YRAN2SAT. During $NN=24$, YRAN2SAT has the highest error because the structure of 1SAT is more than 2SAT, while the smallest error is at $NN=33$ because the structure of 2SAT is more than 1SAT.

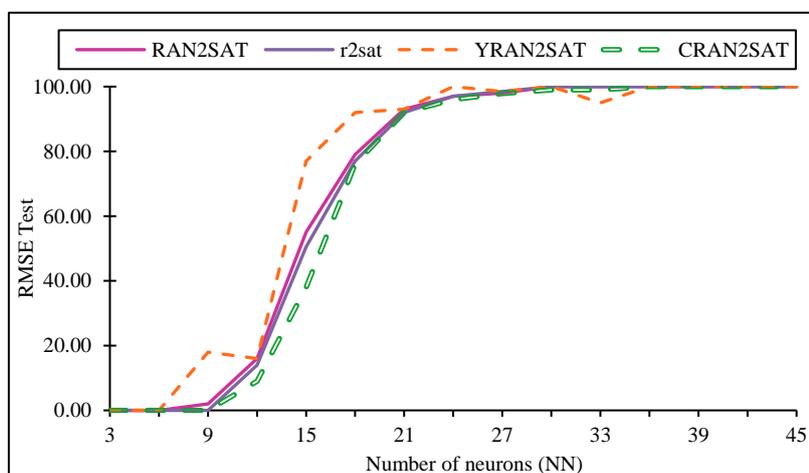


Figure 7. $RMSE_{Test}$ for a different state of the art SAT.

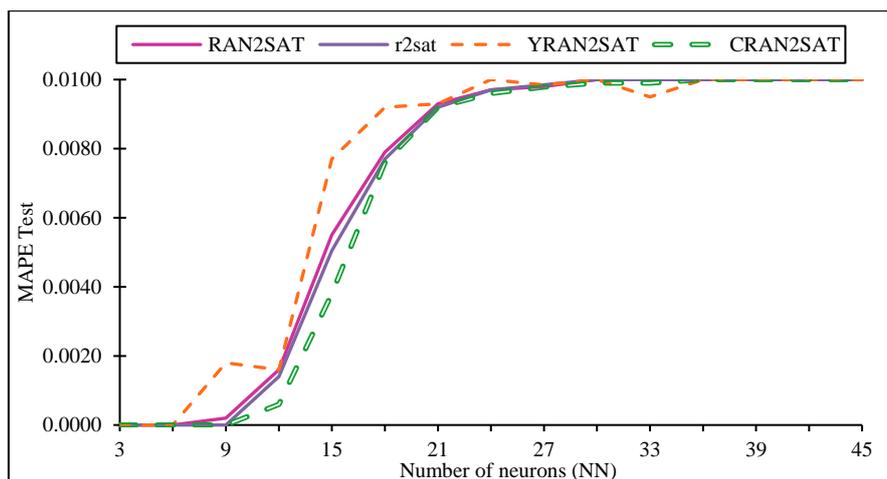


Figure 8. $MAPE_{Test}$ for a different state of the art SAT.

Conversely, DHNN-CRAN2SAT is in a suboptimal retrieval phase for $30 \leq NN \leq 45$, and during this interval, the proposed model achieved the highest testing error. Regardless of having local minima solution at that interval, DHNN-CRAN2SAT still managed to obtain the lowest testing error in terms of $RMSE_{Test}$ and $MAPE_{Test}$, compared to the other logical structure. As indicated by [18], the logical structure of the first-order clause has the potential to disrupt the process of retrieving correct synaptic weight, resulting in increased testing errors. Notably, our research employed ES during the learning phase, introducing a “trial and error” approach that influences the capability to minimize the cost function. If ES is unsuccessful in obtaining an optimal synaptic weight, the testing phase would be affected, potentially leading to the emergence of local minima solutions.

7.1.2. Energy analysis

In this segment, we will explore the energy profile and examine the results (either the solution is global solutions or local solutions) generated by DHNN-CRAN2SAT. The energy profile is important to indicate the convergence of the proposed model. The energy differences remained constant at zero, because DHNN-CRAN2SAT is able to achieve minimum energy in the testing phase. As NN increases, the convergence by the network will be slowly reduced. The energy profile is lower during the smallest value of NN , due to the high probability of getting satisfied interpretation, which leads to the minimization of energy. The energy function in DHNN acts as an indicator of whether the solutions produced by DHNN-CRAN2SAT are optimal or not. Results in Figures 9 and 10 show the RMSE and MAPE energy for a different state of the art SAT, respectively. From Figures 9 and 10, the proposed model DHNN-CRAN2SAT obtained the lowest difference in energy, which indicates that the logic able to achieve the lowest energy profile as compared to another logical rule. This result demonstrates that the number of global minimum energy decreases as the NN increases, as in Figure 9. Due to this, the DHNN-CRAN2SAT that was unable to satisfy the logical inconsistencies is considered as a local solution. During the interval of $3 \leq NN \leq 9$, DHNN-CRAN2SAT is able to retrieve a maximum number of global solutions, because of the ability of the model to minimize the cost function.

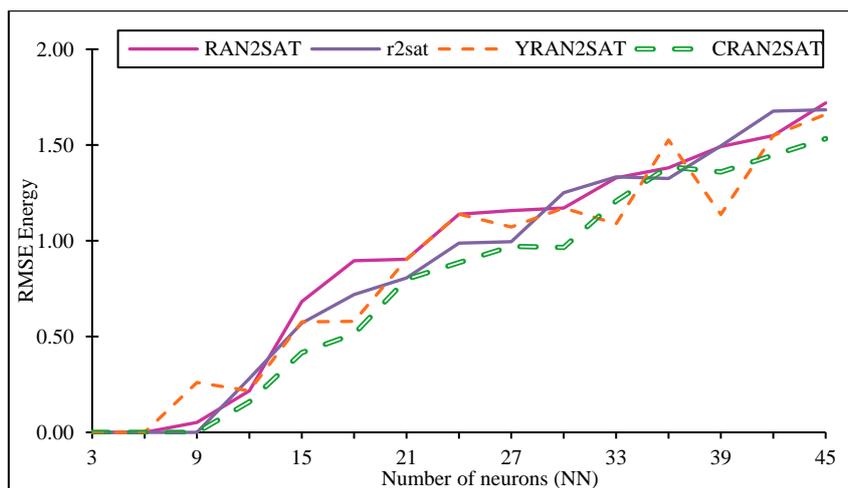


Figure 9. $RMSE_{Energy}$ for a different state of the art SAT.

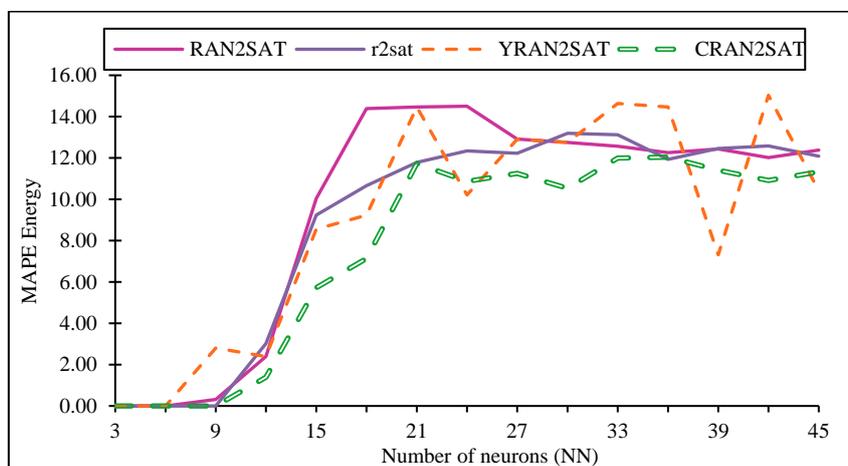


Figure 10. $MAPE_{Energy}$ for a different state of the art SAT.

The result of the global solution is represented as a global minimum ratio (Z_m). [10] stated that if a model can achieve the highest Z_m , it denotes that the proposed SAT has been successfully integrated into the DHNN. Therefore a number of neurons between the interval $3 \leq NN \leq 9$ is the best range to represent our model. During this interval, based on Figure 11, DHNN-CRAN2SAT can achieve maximum fitness of clause satisfaction. This contributes to an optimal synaptic weight and, thus is able to obtain a global minimum solution in the retrieval phase. Even though all the logical structure shows that Z_m gradually decreases, DHNN-CRAN2SAT manages to obtain the highest total value of Z_m compared to r2sat, YRAN2SAT and RAN2SAT. Notably, Figures 9 and 10 illustrate the structure of YRAN2SAT yielding to a waver energy profile and Z_m . It reveals that the minimum energy of various logic is dependent on the existence of different SAT clauses.

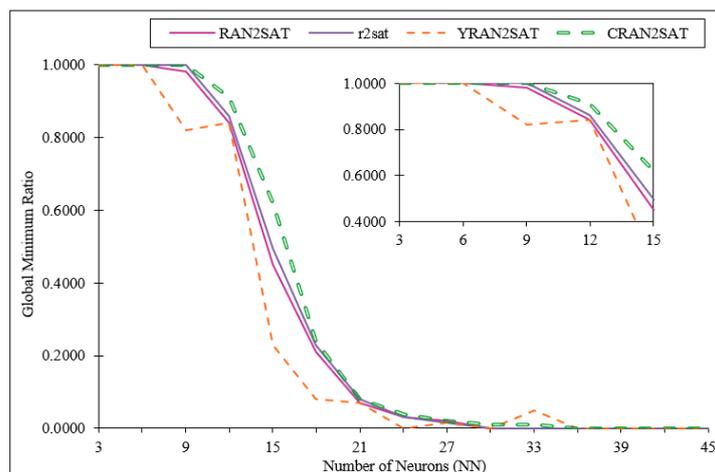


Figure 11. Z_m for a different state of the art SAT.

According to [27], the Lyapunov energy function is an important factor to access the convergence of DHNN. Based on the results in Figures 9 and 10, it is observed that the lower probability of getting satisfied interpretation during the learning phase will contribute to the higher energy profile. Due to this fact, the proposed model DHNN-CRAN2SAT successfully shows the relationship between the learning phase and the testing phase of DHNN. The capability of obtaining a quality of final neuron state depends on the Lyapunov energy function. The quality of a final neuron state in the testing phase is crucial because the retrieved final neuron state will demonstrate the nature of the model. According to [10], having a wider variety of final neuron states can be beneficial when undertaking real-world classification or forecasting tasks. For this purpose, a new activation function will be implemented in the retrieval phase to enhance the effectiveness of the updating rule within DHNN, which can provide more variations and increased diversity in the final neuron states. The analysis based on different activation functions will be discussed in the next section.

7.1.3. Similarity analysis

Within this segment, the assessment of the retrieved final neuron state is performed through the application of the total neuron variation (TV) and the Jaccard similarity index (JSI). It's important to highlight that the TV and JSI metrics are exclusively computed for the final neuron state that attains the global minimum energy. A higher TV suggests that the model has the ability to examine different solutions across varied areas in the search space. The JSI is utilized to assess the variety of final neuron states generated by DHNN-CRAN2SAT, by contrasting it with the benchmark neuron states.

A smaller JSI value indicates that the obtained final neuron states display a high degree of deviation from the benchmark states. Figure 12 displays the TV for different state of the art SAT. In Figure 12, it can be observed that RAN2SAT, r2sat and YRAN2SAT achieve an optimal TV during highest $NN=12$. However, CRAN2SAT extended to obtain the highest TV at $NN=15$. It is also notable that when $NN \geq 30$, the TV gradually decreases to a zero value. As for YRAN2SAT, during $NN = 33$, it still manages to obtain higher TV compared to the other logical rule due to the structure of 2SAT more than 1SAT.

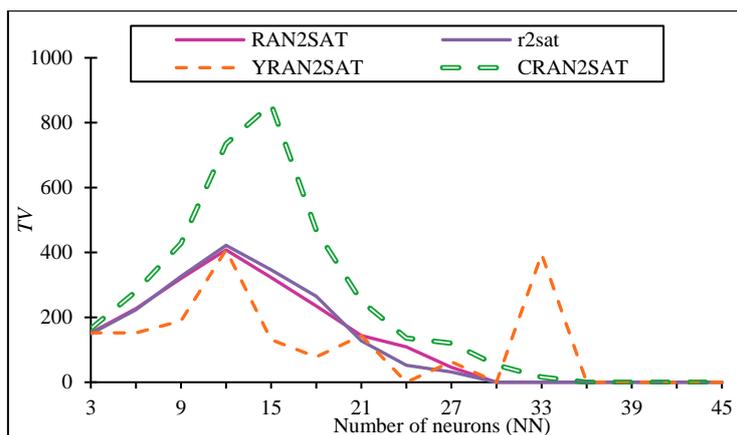


Figure 12. TV for a different state of the art SAT.

Since our study incorporated ES during the learning phase, as NN increases, the utilization of the “trial and error” approach will consequently impact the minimization of the cost function. If ES were unable to attain an optimal synaptic weight, this would influence the testing phase and consequently lead to a solution trapped in local minima. Thus, it will impact the evaluation of the retrieved final neuron state. This is because the TV and JSI will only measure the final neuron state that achieves global minimum energy. For this reason, as NN increases, the probability of the global minimum solution will decrease and, thus, the TV will also decrease.

The performance of the logical rule in terms of JSI is represented in Figure 13.

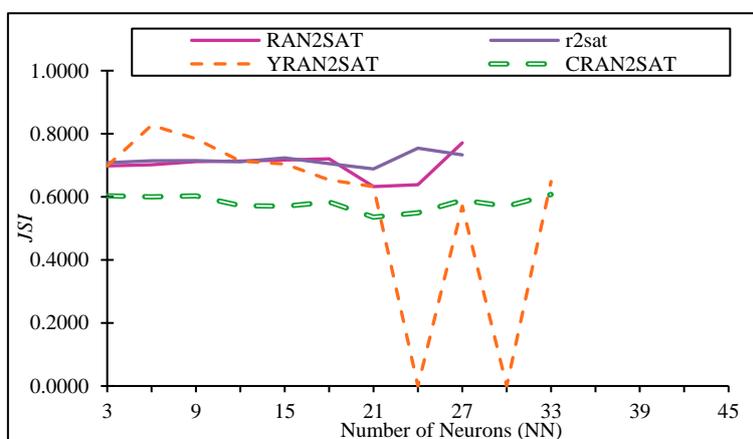


Figure 13. Jaccard similarity index for a different state of the art SAT.

Based on the results, CRAN2SAT shows the lowest JSI , while YRAN2SAT shows a fluctuate value of JSI . The structure of CRAN2SAT, which restricts the logical structure by excluding both positive literals in second-order clauses, successfully perceives the significance of negative literals in the second order clause. Notably, the structure of CRAN2SAT does not impose any restrictions for the first-order clauses. In this study, the number of negative literals for r2SAT is set to have 50% out of the NN . By comparing r2sat with RAN2SAT, there is a similar pattern of JSI , which means that during $3 \leq NN \leq 18$, it implies that there is low dissimilarity of the final neuron state with the benchmark states. However, JSI for the CRAN2SAT model is relatively small compared to the other logical structure, which indicates that the difference between the final neuron state and the benchmark state is

relatively large. From the above discussion, we provided a clear concept of cost minimization, synaptic weight, energy profile, and neuron variation, which describes the overall behavior of DHNN-CRAN2SAT. It can be concluded that the proposed logical rule, CRAN2SAT, successfully implemented in DHNN by lower learning error and diversified final neuron states.

The authors' desire to optimize the retrieved final neuron state during the testing phase was a stepping stone for the next research study. A new activation function will be implemented in the retrieval phase to improve the effectiveness of the updating rule in DHNN, which can provide more variations and increased diversity in the final neuron states. More variation in the final neuron states is very effective for logic mining, which can be imposed on numerous fields. Hence, this research introduces a novel non-monotonic Smish activation function designed to introduce greater variation and enrich the diversity in the final neuron states. The discussion on different activation functions will be covered in the next section.

7.1.4. Synaptic weight analysis

This section will discuss the management of synaptic weights in relation to the proposed CRAN2SAT model. The value of the synaptic weight such as W_{AB} , W_A and W_B are obtained by using the Abdullah method [1]. The proposed CRAN2SAT model restricts the logical structure by excluding both positive literals in second-order clauses without imposing any restrictions on the first-order clauses. By incorporating this characteristic, our model will have a second order clause with at least one negative literal in each clause. Therefore, by considering this criterion, we acknowledge that there are three possible combinations of 2SAT clauses, and two possible combinations of 1SAT clauses, while the existing non-systematic logic for $k=1,2$, such as RAN2SAT, will have four possible combinations of 2SAT clauses and 2 possible combinations of 1SAT clauses. By exhibiting to the value of synaptic weight of CRAN2SAT, there will be no value of synaptic weight for both positive literals in second-order clauses, $(A \vee B)$. This is because of the criterion of the CRAN2SAT model that excludes both positive literals in second-order clauses. [44] states that the Hopfield neural network is restricted to a symmetric connection network, which also impacts the limited storage capacity of the DHNN. Since we have reduced the $(A \vee B)$ clause from our proposed model, the storage complexity of CAM has been reduced. Furthermore, based on the individual contribution of the synaptic weight for both positive literals in second-order clauses $(A \vee B)$, the DHNN will retrieve a repetitive final neuron state. As mentioned in the previous section, the final neuron state will be transformed to one if $g(h_i) \geq 0$. The DHNN will retrieve a final neuron state based on the value obtained by the local field h_i . Due to this reason, the individual contribution of the synaptic weight for both positive literals in second-order clauses $(A \vee B)$ will contribute to a high probability of getting a repetitive final neuron state. Consequently, this will result to the low diversification between the final neuron state. This is supported by the performance of the proposed logic CRAN2SAT. The CRAN2SAT model, which constrains the logical structure by excluding positive literals in second-order clauses, has effectively demonstrated the significance of incorporating negative literals in generating a diverse range of final neuron states and achieving the highest global solution (refer Figures 11–13).

7.2. Different activation functions for DHNN-CRAN2SAT

This study has proposed a non-monotonic Smish activation function in DHNN-CRAN2SAT. In this section, the retrieval phase of DHNN-CRAN2SAT with the Smish activation function will be compared with other types of activation functions, namely, McCulloch-Pitts, piecewise linear activation function, Elliot symmetric activation function, HTAF and swish activation. As mentioned in the work [45], the analysis during the retrieval phase plays an important role as DHNN often tends to produce repetitive states instead of generating new final neuron states. According to [9], the initial neuron state that is generated during the initialization is always the same, which is only one type of neuron state. Thus, the result of the final neuron state only converges to only one type of solution. To overcome this issue, we proposed a non-monotonic Smish activation function to optimize the final neuron state retrieved by DHNN. Despite achieving a good performance in the hyperbolic tangent activation function by [16], the interpretability of error analysis and neuron variations during the retrieval phase in logic satisfiability DHNN remains unknown. Therefore, this paper will continue their work by analyzing six types of activation functions. Hence, the efficiency of different activation functions in retrieving a final neuron state will be compared. Additionally, the full capability of non-monotonic Smish activation function will be investigated throughout existing SAT logic in DHNN, such as RAN2SAT, r2AT and YRAN2SAT.

7.2.1. Analysis on DHNN-CRAN2SAT with proposed non-monotonic Smish activation

In this section, the effectiveness of the proposed non-monotonic Smish activation function in DHNN-CRAN2SAT with Smish activation function will be compared with the other types of activation functions. Results in Figures 14 and 15 show the RMSE and MAPE error for a six types of activation functions, respectively. Figures 14 and 15 show that during the interval of $3 \leq NN \leq 9$, the Smish activation function is able to generate an optimal synaptic weight that leading to a zero error for $RMSE_{Test}$ and $MAPE_{Test}$.

During this interval, DHNN-CRAN2SAT with Smish activation function is in optimal retrieval phase where it performs well in generating global minima solutions. Figures 14 and 15 show that is not only Smish activation able to retrieve an optimal synaptic weight during $3 \leq NN \leq 9$, but also the other five activation functions in optimal retrieval phase. This result is due to the capability of DHNN in minimizing the logical inconsistency during the learning phase. However, as NN increases, the testing error in terms of $RMSE_{Test}$ and $MAPE_{Test}$ will increase for all types of activation functions. Based on the results illustrated in Figures 14 and 15, the proposed non-monotonic Smish activation function in DHNN-CRAN2SAT obtained the lowest error compared to the other activation functions. Lower error values observed during the retrieval phase signify the effectiveness of the Smish activation function in identifying global minimum solutions over local minima solutions. The convergence property of the DHNN model can be validated by the existence of a high proportion of global minima solutions. Therefore, to assess the retrieved final neuron state, the behaviors of the Lyapunov energy function associated with the activation function utilized in the DHNN will be examined effectively. The convergence of the final neuron state can be observed by calculating the differences in Eq (12).

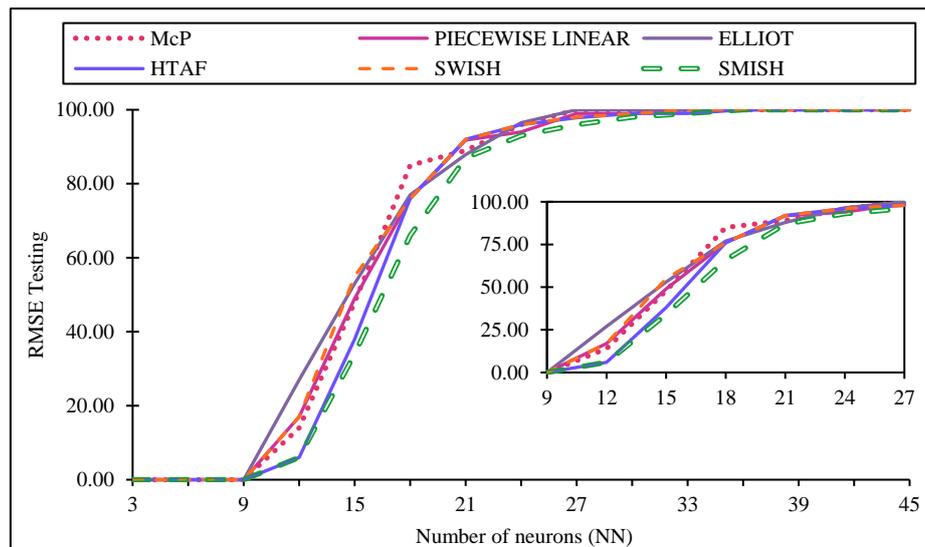


Figure 14. $RMSE_{Test}$ for DHNN-CRAN2SAT with various types of activation function.

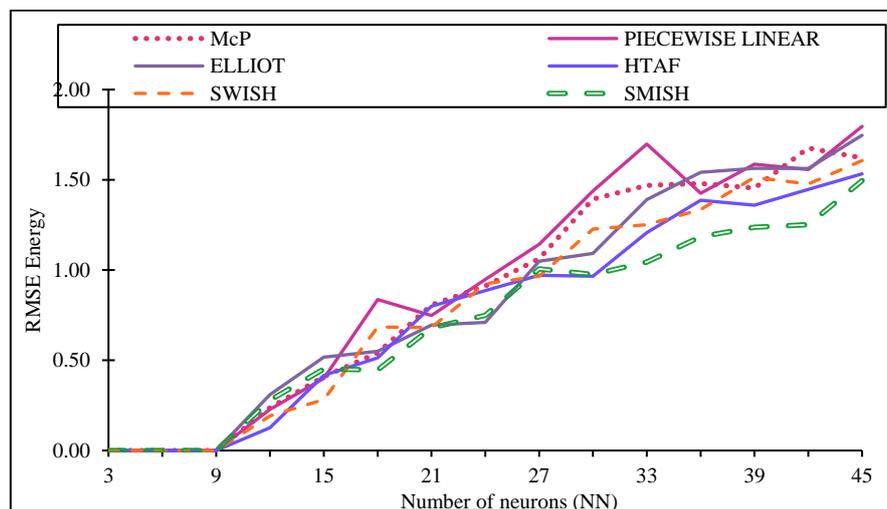


Figure 15. $RMSE_{Energy}$ for DHNN-CRAN2SAT with various types of activation function.

According to a study conducted by [8], it has been noted that the Lyapunov energy function consistently exhibits a monotonically decreasing trend, meaning that the energy derived from its logic always decreases over time. Due to the mentioned properties, it is believed that the non-monotonic activation function is the most compatible with the properties of Lyapunov energy function. This thought is verified based on the result demonstrated in Table 7, where the non-monotonic Smish activation function has the ability to contribute more global solution compared to the other types of activation functions. It shows that non-monotonic Smish activation functions can prevent the network from getting trapped in local minima during the retrieval phase. Thus, the Smish activation function can help DHNN in producing the highest value of global solutions, highest total neuron variation as well as more diversified final neuron state. The non-monotonic characteristic of the Smish activation function assures stable negative training, therefore improving the performance of expression. Due to the stability of negative training, this will influence the DHNN in the retrieved diversified final neuron state. Moreover, the non-monotonic nature of Smish within the range of $(-\infty, +\infty)$ can enhance the

network's ability for learning and gradient transformations [26]. If the activation function is not continuously differentiable, it will contribute to an undefined gradient at midpoint of the activation function, which will affect the training performance [29]. Thus, since the Smish activation function is a non-zero derivative at zero, this will help in the learning process, because a very small derivatives tends to stop the learning process. Notably, the parameter value in the Smish activation function where $\alpha = 1$ and $\beta = 1$ helps to maintain the gradient of the non-monotonic function, which makes the solution obtained not be fluctuate. Moreover, the characteristics of the Smish activation function enable neural networks to incorporate negative representations and achieve a diverse final neuron state. The Smish activation function is a smooth non-monotonic function with a lower bound but no upper bound. This can be validated by the results in Table 8, where the Smish activation function can provide more variations in the final neuron states.

The other five activation functions also have their own behavior and properties. The behavior of HTAF saturates when the modulus of the input tends toward infinity and the gradients decrease rapidly. As a consequence, this only permits the training of less complex networks [46]. Despite the desirable properties, such as centering its outputs around zero, HTAF can suffer from saturation and vanishing gradient issues. Especially as the input values become very large or very small, it will cause the output to reach its limits of +1 or -1. During the retrieval phase of DHNN, a zero output will result in a contribution of one to the final neuron state. Due to this issue, HTAF will contribute to the low diversified final neuron states. The McCulloch-Pitts function is the conventional platform that has been used in logic programming in DHNN. This function was popularized by [1] and it is reported the solution will trap in the local minimum of the energy [43]. In addition, the most common and simplistic activation function is the piecewise linear activation function [14]. The function consists of linear segments in different intervals of the input range. The transition points might introduce non-smoothness in the function which will affect the lack of differentiability at the transition points [26]. The Swish activation function involves multiplying the input function by its corresponding sigmoid function. The Swish activation function is also a non-monotonic with the output range of $(-\infty, +\infty)$. Based on the parameter value of β , the shape of the Swish activation function is adjusted between the linear and ReLU functions. The smaller and higher values of β lead toward the linear and ReLU functions, respectively [12]. The same goes to the Elliot symmetric activation function [47], which has similar characteristics to the Sigmoid function. The function is saturated for higher and lower inputs, which leads to vanishing gradient problem [12].

An appropriate activation function enables the network to reach a stable state of convergence and prevents it from oscillating between various states. The quality of the final neuron state in the retrieval phase is crucial because the retrieved final neuron state will demonstrate the nature of the model. Thus, this study claims that the most suitable activation function that is compatible with the Lyapunov energy function is the Smish activation function. The Smish non-monotonic activation functions allow for a wider range of possible solutions while the Lyapunov energy functions can accommodate this diversity by assessing the stability of the network. This is supported by [3], which states that the Lyapunov energy will reach its minimum point (equilibrium state) if DHNN remains stable without any oscillation.

Table 8. Z_m for DHNN-CRAN2SAT for all types of activation function. The bracket denotes the enhancement ratio attributed to the proposed Smish activation function. The bracket (*) indicates the proposed Smish activation outperformed other activation function since there is no value of Z_m for the respected activation function.

NN	Types of activation function					
	Smish	Swish	Hyperbolic tangent	Elliot	Piecewise linear	McCulloch-Pitts
3	1.00000	1.00000 (0.00000)	1.00000 (0.00000)	1.00000 (0.00000)	1.00000 (0.00000)	1.00000 (0.00000)
6	1.00000	1.00000 (0.00000)	1.00000 (0.00000)	1.00000 (0.00000)	1.00000 (0.00000)	1.00000 (0.00000)
9	1.00000	1.00000 (0.00000)	1.00000 (0.00000)	1.00000 (0.00000)	1.00000 (0.00000)	1.00000 (0.00000)
12	0.95500	0.83000 (0.15060)	0.94000 (0.01596)	0.73000 (0.30822)	0.83000 (0.15060)	0.86000 (0.11047)
15	0.66000	0.45000 (0.46667)	0.62000 (0.06452)	0.47000 (0.40426)	0.51000 (0.29412)	0.52000 (0.26923)
18	0.34000	0.24000 (0.41667)	0.24000 (0.41667)	0.23000 (0.47826)	0.24000 (0.41667)	0.15000 (1.26667)
21	0.13000	0.08000 (0.62500)	0.08000 (0.62500)	0.12120 (0.07261)	0.08210 (0.58343)	0.11000 (0.18182)
24	0.07000	0.04000 (0.75000)	0.04000 (0.75000)	0.03490 (1.00573)	0.06000 (0.16667)	0.04000(0.75000)
27	0.04000	0.02000 (1.00000)	0.02140 (0.86916)	0.00000 (*)	0.01000 (3.00000)	0.00000 (*)
30	0.02000	0.01100 (0.81818)	0.01000 (1.00000)	0.00000 (*)	0.01000 (1.00000)	0.00000 (*)
33	0.01850	0.00000 (*)	0.01000 (0.85000)	0.00000 (*)	0.00000 (*)	0.00000 (*)
36	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
39	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
42	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
45	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
(+/-/-)		(8/3/0)	(8/3/0)	(8/3/0)	(8/3/0)	(8/3/0)

By analyzing the result of Zm , the Smish activation function consistently outperformed the other five activation functions in terms of obtaining higher Zm . During the interval of $3 \leq NN \leq 9$, all the activation functions in DHNN were able to retrieve maximum of global solutions because during this interval, there is high probability of getting satisfied interpretation during the learning phase. Thus, it will lead to an optimal synaptic weight. Therefore, during this interval, the retrieved final neuron state is in a global solution. The enhancement ratio Zm for the Smish activation function in comparison to other activation functions was documented and analyzed, as shown in Table 8. The positive ratio of improvement for Smish activation function is consistently higher throughout all the types of activation functions indicating that Smish activation function outperforms other types of activation functions in terms of Zm . Based on the result in Table 8, the ratio of improvement Zm is measured for $3 \leq NN \leq 33$. It is because after $NN \geq 36$, DHNN is unable to retrieve a global solution due to the difficulty of minimizing the logical inconsistency during the learning phase.

Thus, in obtaining a higher value of Zm the Smish activation function wins eight between the range of $3 \leq NN \leq 33$. As shown in Table 8, Smish activation function attained the highest value of Zm between the number of neurons $12 \leq NN \leq 33$ compared to other types of activation functions. It has been mentioned in the previous section that Zm is an important metric in evaluating the performance of DHNN. Based on this result, it is proved that the Smish activation function can improve the effectiveness of the updating rule in DHNN, and incorporating the Smish activation function into the proposed DHNN-CRAN2SAT model can notably boost its performance, particularly in generating Zm .

The value of TV and the ratio of improvement between Smish activation function and other types of activation functions that have been implemented in DHNN-CRAN2SAT were recorded and compared, as in Table 9. Due to the low capability of retrieving a global solution after $NN \geq 36$, for a fair comparison, the ratio of improvement TV is measured for $3 \leq NN \leq 33$. In this study, the Smish activation function achieved the maximum value of TV at $NN=15$ with the value of $TV=920$ has which outperformed other types of activation functions. Furthermore, Smish activation function also have the highest values of TV when compared one to one for each type of activation function. Although Smish activation function cannot compete with other activation functions in producing higher TV during $NN \geq 3$, and by comparing with HTAF, Smish activation function shows no ratio of improvement between $3 \leq NN \leq 6$, the Smish activation function shows the fruitful result by consistently winning nine out of eleven distribution number of neurons. Based on this result, it is shown that by implementing the Smish activation function in DHNN, the network can retrieve a diversified final neuron state even though the number of neurons have been increased. By referring to the figure of Smish activation function, it is shown that Smish activation function has a lower bound with no upper bound. This property has enabled the networks to benefit negative representation, which is compatible to solve problems in the retrieval phase of DHNN. In the DHNN model, it is crucial to introduce diversity in the final neuron state. This is because the DHNN tends to produce repetitive states instead of generating new ones. Moreover, having a greater range of distinct final neuron states would provide an advantage in practical classification or prediction tasks [10]. As demonstrated in Table 9, this research recognizes that the Smish activation function can contribute to a wider range of variations and boost diversity in the final states of the neurons.

Table 9. *TV* for DHNN-CRAN2SAT for all types of activation function. The positive bracket signifies the improvement ratio, while the negative ratio suggests that the other activation function surpassed the effectiveness of the proposed Smish activation function. The bracket (*) signifies the proposed Smish activation outperformed other activation function since there is no value of *TV* for the respected activation function.

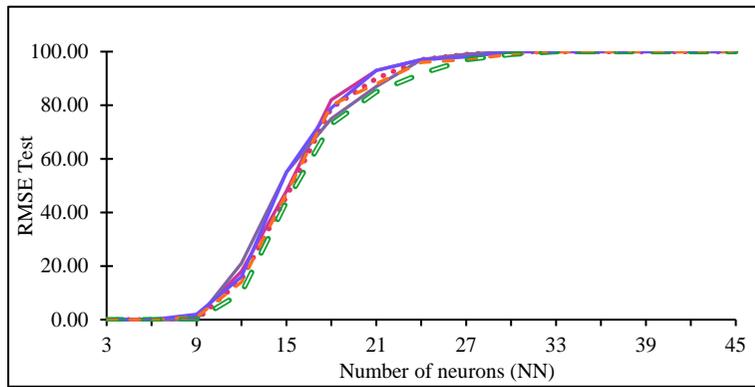
<i>NN</i>	Types of activation function					
	Smish	Swish	Hyperbolic tangent	Elliot	Piecewise linear	McCulloch-Pitts
3	165	166 (-0.0060)	165 (0.0000)	166 (-0.0060)	166 (-0.0060)	166 (-0.0060)
6	280	271 (0.0332)	280 (0.0000)	271 (0.0332)	271 (0.0332)	271 (0.0332)
9	465	465 (0.0000)	428 (0.0864)	465 (0.0000)	465 (0.0000)	465 (0.0000)
12	757	629 (0.2035)	737 (0.0271)	615 (0.2309)	635 (0.1921)	690 (0.0971)
15	920	674 (0.3650)	857 (0.0735)	639 (0.4397)	698 (0.3181)	769 (0.1964)
18	558	501 (0.1138)	468 (0.1923)	452 (0.2345)	453 (0.2318)	341 (0.6364)
21	382	256 (0.4922)	250 (0.5280)	299 (0.2776)	290 (0.3172)	246 (0.5528)
24	247	197 (0.2538)	136 (0.8162)	170 (0.4529)	196 (0.2602)	162 (0.5247)
27	157	151 (0.0397)	120 (0.3083)	0 (*)	16 (8.8125)	0 (*)
30	119	96 (0.2396)	53 (1.2453)	0 (*)	29 (3.1034)	0 (*)
33	86	0 (*)	1 (85.0000)	0 (*)	0 (*)	0 (*)
36	0	0	0	0	0	0
39	0	0	0	0	0	0
42	0	0	0	0	0	0
45	0	0	0	0	0	0
(+/-/-)		(9/1/1)	(9/2/0)	(9/1/1)	(9/1/1)	(9/1/1)

7.3. Comparison of six different types of activation function with existing SAT in DHNN

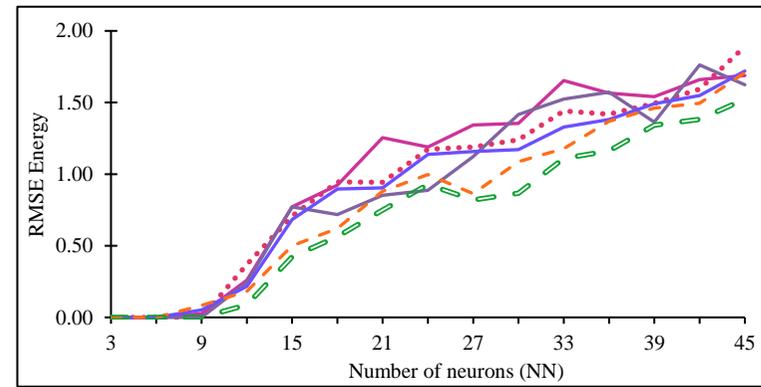
To support the effectiveness of the non-monotonic Smish activation function, the proposed activation function is embedded in the different types of non-systematic logical structure such as RAN2SAT, r2sat and YRAN2SAT. The performance of these logical structures will be evaluated with different types of activation functions, and the result is based on the $RMSE_{Test}$, $RMSE_{Energy}$, Zm and TV as shown in Figures 16–18, respectively. Based on the results in Figures 16–18, it shows that the non-monotonic Smish activation function is not only compatible in the DHNN-CRAN2SAT, but also compatible with other logical structures. Based on the results in Figures 16a–c and 17a–c, during the interval of $3 \leq NN \leq 9$, the logical structure of RAN2SAT and r2sat are able to generate an optimal synaptic weight, which leads to a zero error for $RMSE_{Test}$ and $RMSE_{Energy}$. However, as NN increases, there is a difficulty in minimizing the logical inconsistency during the learning phase, leading to suboptimal synaptic weight. Due to this reason, the value of Zm starts to decrease as $NN \geq 9$. In this context, the Smish activation function shows the ability of generating more Zm compared to the other types of activation functions. Although YRAN2SAT offers the flexibility of the logical structure, it does not limit the capability of Smish activation function in providing a good result in terms of $RMSE_{Test}$, $RMSE_{Energy}$ and Zm . The variability in the outcome observed in Figure 18 is attributed to the adaptability of both the number of first-order clauses and second-order clauses. Therefore, Smish activation functions can prevent the network from getting trapped in local minima during the retrieval phase, which helps in producing the highest value of global solutions with lowest testing error.

Based on the results in Figures 16d and 17d, there is a zero value of during due to the disability of DHNN to retrieve a global solution. It is because of the difficulty in minimizing the logical inconsistency during the learning process. Therefore, by analyzing the value of TV for each logical structure (refer Figures 16d and 17d), it is shown that the implementation of Smish activation function in DHNN has improved the capability of generating a diversified solution. The highest value of TV obtained by RAN2SAT, r2SAT and YRAN2SAT is $TV=496,481,463$, respectively. By referring to Figure 17d, during $NN=24$, the value of $TV=0$. It is because at $NN=24$, the structure of YRAN2SAT had a higher number of first order clauses compared to the second order clauses. Due to this reason, YRAN2SAT was unable to complete the learning phase due to the high percentage of obtaining unsatisfied clauses, which lead to a local solution. The highest result of Zm and TV highlights the effectiveness of the Smish activation function in DHNN. It shows the Smish activation function has capability to improve the effectiveness of the update of the final neuron state.

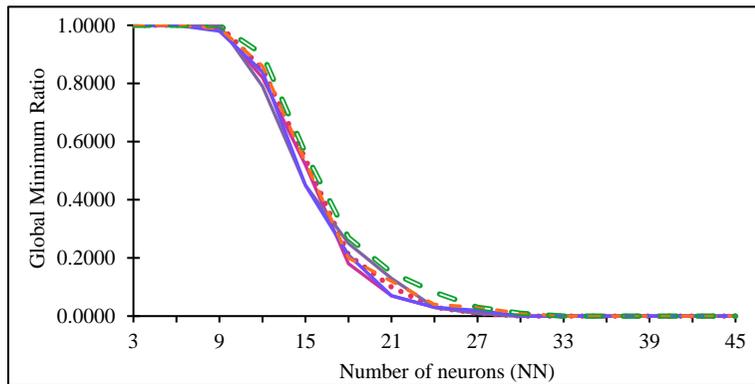
The other activation function has a problem in terms of saturated for higher and lower inputs, which leads to a vanishing gradient problem [12]. Due to this effect, the DHNN tends to retrieve a repetitive final neuron state. Additionally, the problem of a diminishing gradient will affect the convergence speed of the network [13] and affects the training performance [29]. Therefore, based on the above analysis, it is validated that the proposed non-monotonic Smish activation function has the capability to improve the effectiveness of the updating rule in DHNN, which can provide more variations and increase diversity in the final neuron states. The effectiveness of obtaining more variations in the final neuron states is beneficial in the logic mining, which can be imposed on numerous fields. Thus, this is another fruitful finding because the implementation of non-monotonic Smish activation function in DHNN has successfully provided more global solution with the highest diversification in the final neuron states.



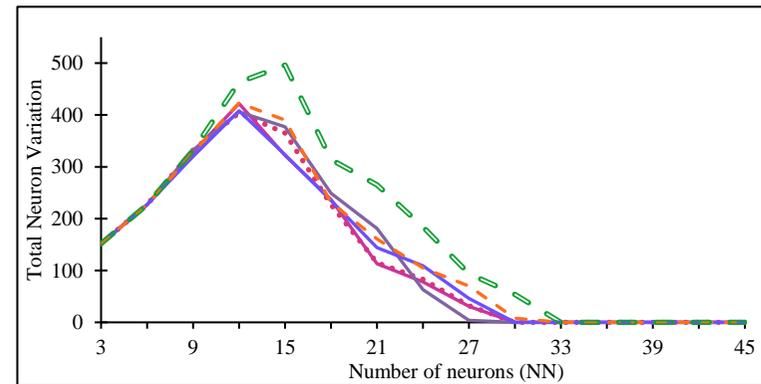
(a)



(b)



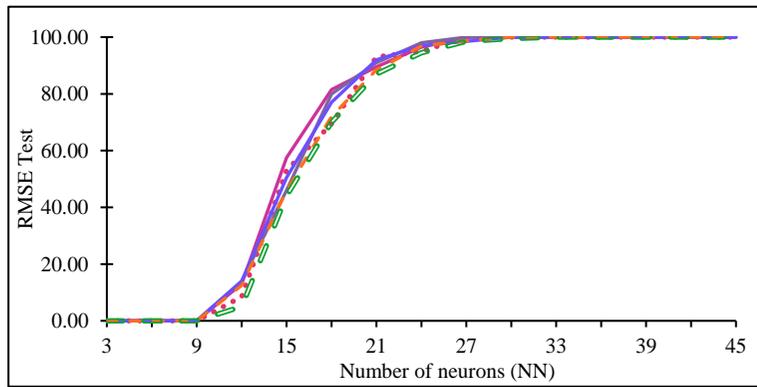
(c)



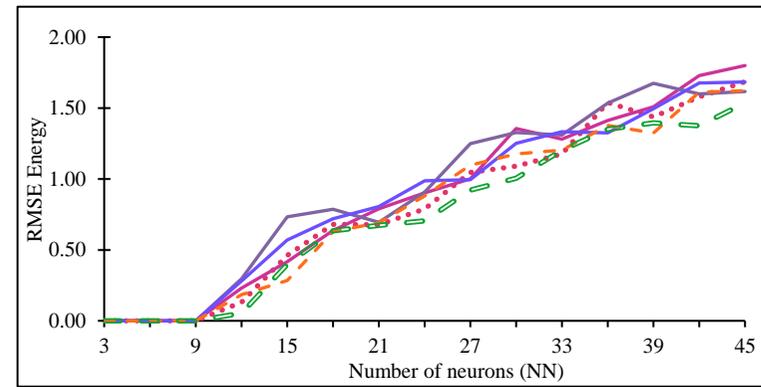
(d)



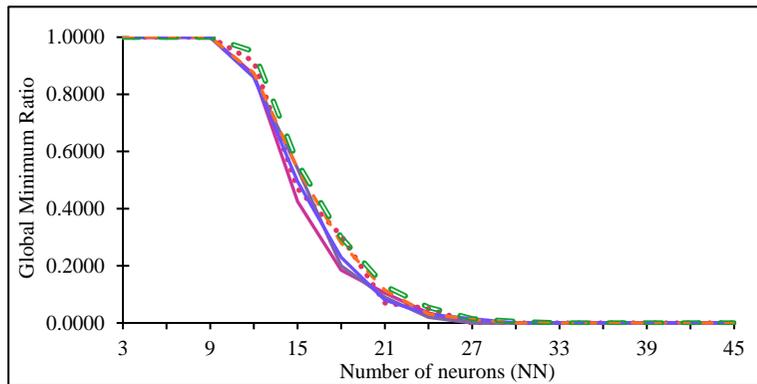
Figure 16. Performance of different activation functions in DHNN-RAN2SAT.



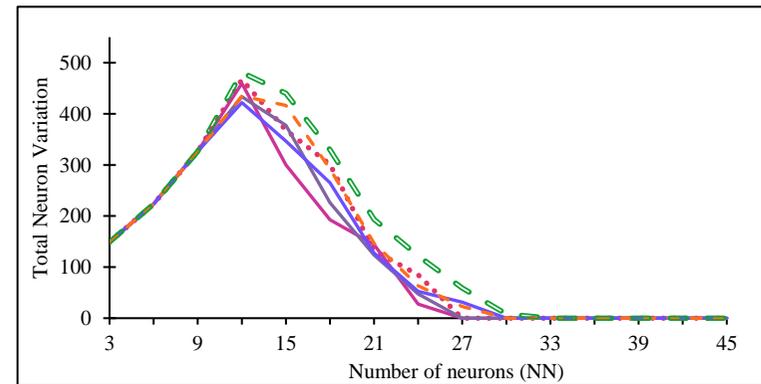
(a)



(b)



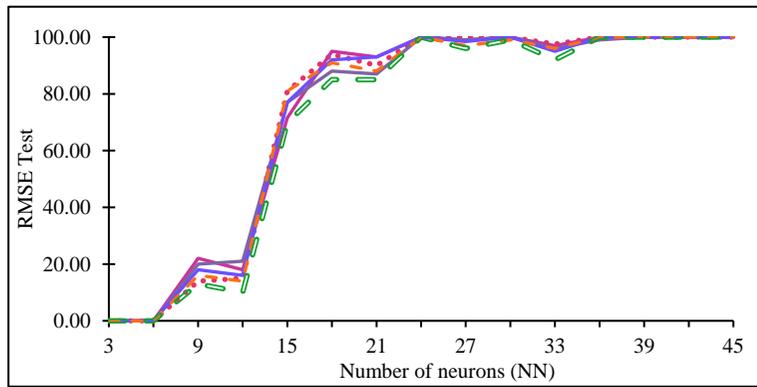
(c)



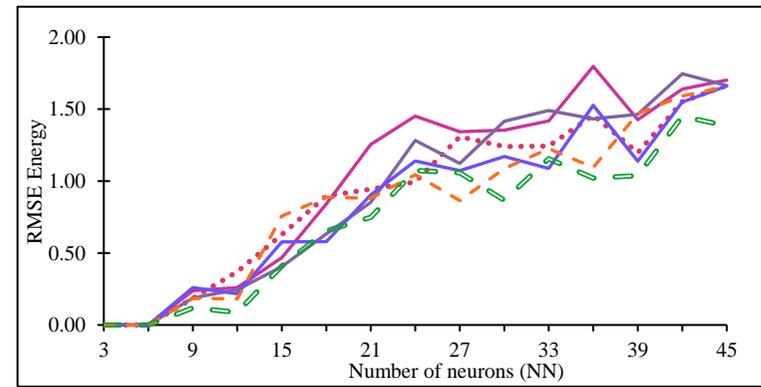
(d)



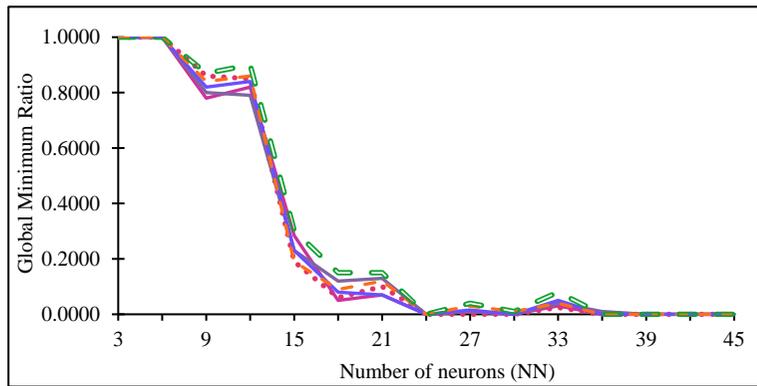
Figure 17. Performance of different activation functions in DHNN-r2SAT.



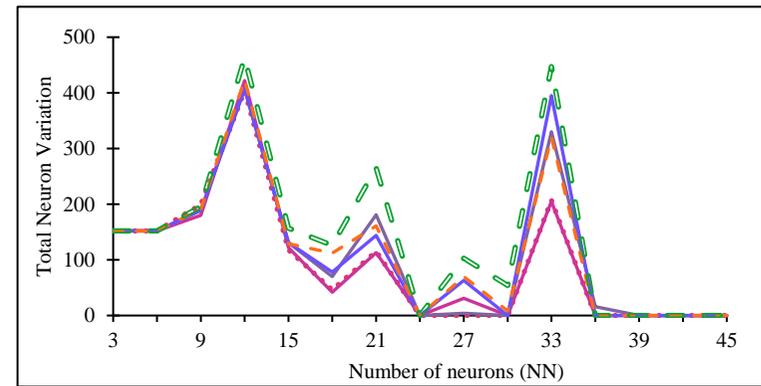
(a)



(b)



(c)



(d)



Figure 18. Performance of different activation functions in DHNN-YRAN2SAT.

8. Conclusions

This paper presented a novel non-systematic logical structure that uses first order and second order logic without including both positive literal in the second order clauses, namely, CRAN2SAT. The proposed logic CRAN2SAT was implemented into the DHNN by minimizing the cost function. By comparing the cost function with the Lyapunov energy function, the optimal synaptic weight was obtained. The performance of CRAN2SAT was compared with different logical rules (RAN2SAT, r2SAT and YRAN2SAT). The proposed logic CRAN2SAT has demonstrated an excellent performance in the learning and retrieval phase, as compared to the other state of the art logical rules. This result indicates that by having relatively high proportion of negative literals in the second order clauses can exhibit superior performance in producing global solution with a diversified final neuron state. Furthermore, the experimental simulations with different types of activation functions (McCulloch-Pitts, piecewise linear, Elliot symmetric, hyperbolic tangent and Swish) show that DHNN-CRAN2SAT with the proposed non-monotonic Smish activation function obtained the highest total neuron variations and provide more diversity in the final neuron states. The efficacy of the non-monotonic Smish activation function is further analyzed throughout different non-systematic logical structures. Based on the result, the Smish activation function manages to achieve a good performance for each of the logical structures. Thus, the Smish activation function highlights its capacity to retrieve a diversified final neuron state within the DHNN framework. Worth mentioning that, this is the first attempt to implement the Smish activation function in the DHNN. The proposed activation function can be integrated into various existing works, as demonstrated by previous studies [48–51]. In solving real-world classifications datasets, it is important to consider diverse solutions in order to improve a model's ability to adapt to various data structures. Having a wider range of final neuron states can be beneficial for solving real-world classification or forecasting tasks, as mentioned by [10]. Therefore, the proposed logical rule CRAN2SAT with Smish activation function in DHNNs successfully achieves the desired final neuron states in terms of high diversification between the neuron states. For future work, the learning phase can be optimized in various perspectives. For example, the whale optimization algorithm by [52] and black hole algorithm by [53] can be added to the learning phase to obtain an optimal synaptic weight. In addition, the proposed CRAN2SAT also can be implemented in the real-world dataset, as proposed by [33,54,55].

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The author is deeply thankful to the editor and the reviewers for their valuable suggestions to improve the quality of this manuscript. All the authors gratefully acknowledged the financial support by the Research University Grant (RUI) (1001/PMATHS/8011131) by Universiti Sains Malaysia.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. W. A. T. W. Abdullah, Logic programming on a neural network, *Int. J. Intell. Syst.*, **7** (1992), 513–519. <https://doi.org/10.1002/int.4550070604>
2. G. Detorakis, T. Bartley, E. Neftci, Contrastive Hebbian learning with random feedback weights, *Neural Networks*, **114** (2019), 1–14. <https://doi.org/10.1016/j.neunet.2019.01.008>
3. S. Sathasivam, Upgrading logic programming in Hopfield network, *Sains Malays.*, **39** (2010), 115–118.
4. M. S. M. Kasihmuddin, S. Sathasivam, M. A. Mansor, Hybrid genetic algorithm in the Hopfield network for logic satisfiability problem, *Pertanika J. Sci. Technol.*, 2017.
5. M. A. Mansor, M. S. M. Kasihmuddin, S. Sathasivam, Artificial immune system paradigm in the Hopfield network for 3-satisfiability problem, *Pertanika J. Sci. Technol.*, **25** (2017), 1173–1188.
6. S. Alzaeemi, M. A. Mansor, M. M. Kasihmuddin, S. Sathasivam, M. Mamat, Radial basis function neural network for 2 satisfiability programming, *Indonesian J. Electron. Eng. Comput. Sci.*, **18** (2020), 459–469. <https://doi.org/10.11591/ijeecs.v18.i1>
7. S. A. Alzaeemi, K. G. Tay, A. Huong, S. Sathasivam, M. K. M. Ali, Evolution performance of symbolic radial basis function neural network by using evolutionary algorithms, *Comput. Syst. Sci. Eng.*, **47** (2023), 1163–1184. <https://doi.org/10.32604/csse.2023.038912>
8. S. Sathasivam, M. A. Mansor, A. I. M. Ismail, S. Z. M. Jamaludin, M. S. M. Kasihmuddin, M. Mamat, Novel random k satisfiability for $k \leq 2$ in Hopfield neural network, *Sains Malays.*, **49** (2020), 2847–2857. <https://doi.org/10.17576/jsm-2020-4911-23>
9. Y. Guo, M. S. M. Kasihmuddin, Y. Gao, M. A. Mansor, H. A. Wahab, N. E. Zamri, et al., YRAN2SAT: a novel flexible random satisfiability logical rule in discrete Hopfield neural network, *Adv. Eng. Software*, **171** (2022), 103169. <https://doi.org/10.1016/j.advengsoft.2022.103169>
10. N. E. Zamri, S. A. Azhar, M. A. Mansor, A. Alway, M. S. M. Kasihmuddin, Weighted random k satisfiability for $k=1,2$ (r2SAT) in discrete Hopfield neural network, *Appl. Soft Comput.*, **126** (2022), 109312. <https://doi.org/10.1016/j.asoc.2022.109312>
11. S. S. M. Sidik, N. E. Zamri, M. S. M. Kasihmuddin, H. A. Wahab, Y. Guo, M. A. Mansor, Non-systematic weighted satisfiability in discrete Hopfield neural network using binary artificial bee colony optimization, *Mathematics*, **10** (2022), 1129. <https://doi.org/10.3390/math10071129>
12. S. R. Dubey, S. K. Singh, B. B. Chaudhuri, Activation functions in deep learning: a comprehensive survey and benchmark, *Neurocomputing*, **503** (2022), 92–108. <https://doi.org/10.1016/j.neucom.2022.06.111>
13. A. Apicella, F. Donnarumma, F. Isgrò, R. Prevete, A survey on modern trainable activation functions, *Neural Networks*, **138** (2021), 14–32. <https://doi.org/10.1016/j.neunet.2021.01.026>
14. A. C. Mathias, P. C. Rech, Hopfield neural network: the hyperbolic tangent and the piecewise-linear activation functions, *Neural Networks*, **34** (2012), 42–45. <https://doi.org/10.1016/j.neunet.2012.06.006>
15. C. Chen, F. Min, Y. Zhang, H. Bao, ReLU-type Hopfield neural network with analog hardware implementation, *Chaos Solitons Fract.*, **167** (2023), 113068. <https://doi.org/10.1016/j.chaos.2022.113068>
16. M. A. Mansor, S. Sathasivam, Accelerating activation function for 3-satisfiability logic programming, *Int. J. Intell. Syst. Appl.*, **8** (2016), 44–50. <https://doi.org/10.5815/ijisa.2016.10.05>

17. S. Abdeen, M. S. M. Kasihmuddin, N. E. Zamri, G. Manoharam, M. A. Mansor, N. Alshehri, S-type aandom k satisfiability logic in discrete Hopfield neural network using probability distribution: performance optimization and analysis, *Mathematics*, **11** (2023). <https://doi.org/10.3390/math11040984>
18. S. A. Karim, N. E. Zamri, A. Alway, M. S. M. Kasihmuddin, A. I. M. Ismail, M. A. Mansor, et al., Random satisfiability: a higher-order logical approach in discrete Hopfield neural network, *IEEE Access*, **9** (2021), 50831–50845. <https://doi.org/10.1109/ACCESS.2021.3068998>
19. V. Someetheram, M. F. Marsani, M. S. M. Kasihmuddin, N. E. Zamri, S. S. M. Sidik, S. Z. M. Jamaludin, Random maximum 2 satisfiability logic in discrete Hopfield neural network incorporating improved election algorithm, *Mathematics*, **10** (2022), 4734. <https://doi.org/10.3390/math10244734>
20. M. Soeken, G. Meuli, B. Schmitt, F. Mozafari, H. Riener, G. D. Micheli, Boolean satisfiability in quantum compilation, *Philos. Trans. R. Soc. A*, **378** (2020), 161. <https://doi.org/10.1098/rsta.2019.0161>
21. C. Hireche, H. Drias, H. Moulai, Grid based clustering for satisfiability solving, *Appl. Soft Comput.*, **88** (2020), 106069. <https://doi.org/10.1016/j.asoc.2020.106069>
22. H. Yamashita, K. Aihara, H. Suzuki, Timescales of Boolean satisfiability solver using continuous-time dynamical system, *Commun. Nonlinear Sci. Numer. Simul.*, **84** (2020), 105183. <https://doi.org/10.1016/j.cnsns.2020.105183>
23. L. C. Kho, M. S. M. Kasihmuddin, M. Mansor, S. Sathasivam, Logic mining in league of legends, *Pertanika J. Sci. Technol.*, **28** (2020), 211–225.
24. J. Feng, S. Lu, Performance analysis of various activation functions in artificial neural networks, *J. Phys.*, **1237** (2019), 022030. <https://doi.org/10.1088/1742-6596/1237/2/022030>
25. G. Wang, Z. Hao, B. Zhang, L. Jin, Convergence and robustness of bounded recurrent neural networks for solving dynamic Lyapunov equations, *Inf. Sci.*, **588** (2022), 106–123. <https://doi.org/10.1016/j.ins.2021.12.039>
26. X. Wang, H. Ren, A. Wang, Smish: a novel activation function for deep learning methods, *Electronics*, **11** (2022). 540. <https://doi.org/10.3390/electronics11040540>
27. M. S. M. Kasihmuddin, M. A. Mansor, M. F. M. Basir, S. Sathasivam, Discrete mutation Hopfield neural network in propositional satisfiability, *Mathematics*, **7** (2019), 1133. <https://doi.org/10.3390/math7111133>
28. Y. Koçak, G. Ü. Şiray, New activation functions for single layer feedforward neural network, *Expert Syst. Appl.*, **164** (2021), 113977. <https://doi.org/10.1016/j.eswa.2020.113977>
29. S. Kiliçarslan, C. Közkurt, S. Baş, A. Elen, Detection and classification of pneumonia using novel superior exponential (SupEx) activation function in convolutional neural networks, *Expert Syst. Appl.*, **217** (2023), 119503. <https://doi.org/10.1016/j.eswa.2023.119503>
30. D. Misra, Mish: a self regularized non-monotonic activation function, *ArXiv*, 2019. <https://doi.org/10.48550/arXiv.1908.08681>
31. J. Liu, Y. Liu, Q. Zhang, A weight initialization method based on neural network with asymmetric activation function, *Neurocomputing*, **483** (2022), 171–182. <https://doi.org/10.1016/j.neucom.2022.01.088>
32. F. L. Azizan, S. Sathasivam, M. K. M. Ali, Hybridised intelligent dynamic model of 3-satisfiability fuzzy logic Hopfield neural network, *Pertanika J. Sci. Technol.*, **31** (2023), 6. <https://doi.org/10.47836/pjst.31.4.06>

33. S. Z. M. Jamaludin, N. A. Romli, M. S. M. Kasihmuddin, A. Baharum, M. A. Mansor, M. F. Marsani, Novel logic mining incorporating log linear approach, *J. King Saud Univ.*, **34** (2022), 9011–9027. <https://doi.org/10.1016/j.jksuci.2022.08.026>
34. J. L. Salmeron, A. Ruiz-Celma, Elliot and symmetric Elliot extreme learning machines for Gaussian noisy industrial thermal modelling, *Energies*, **12** (2018), 90. <https://doi.org/10.3390/en12010090>
35. S. P. Ittiyavirah, S. A. Jones, P. Siddarth, Analysis of different activation functions using backpropagation neural networks, *J. Theor. Appl. Inf. Technol.*, **47** (2013), 1344–1348.
36. H. Abdel-Nabi, G. Al-Naymat, M. Z. Ali, A. Awajan, HcLSH: a novel non-linear monotonic activation function for deep learning methods, *IEEE Access*, **11** (2023), 47794–47815. <https://doi.org/10.1109/ACCESS.2023.3276298>
37. J. Brownlee, *Better deep learning: train faster, reduce overfitting, and make better predictions*, Machine Learning Mastery, 2018.
38. P. Ramachandran, B. Zoph, Q. V. Le, Searching for activation functions, *ArXiv*, 2017. <https://doi.org/10.48550/arXiv.1710.05941>
39. S. Elfving, E. Uchibe, K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, *Neural Networks*, **107** (2018), 3–11. <https://doi.org/10.1016/j.neunet.2017.12.012>
40. A. Alway, N. E. Zamri, S. A. Karim, M. A. Mansor, M. S. M. Kasihmuddin, M. M. Bazuhair, Major 2 satisfiability logic in discrete Hopfield neural network, *Int. J. Comput. Math.*, **99** (2022), 924–948. <https://doi.org/10.1080/00207160.2021.1939870>
41. F. L. Azizan, S. Sathasivam, M. K. M. Ali, N. Roslan, C. Feng, Hybridised network of fuzzy logic and a genetic algorithm in solving 3-satisfiability Hopfield neural networks, *Axioms*, **12** (2023), 250. <https://doi.org/10.3390/axioms12030250>
42. Y. Gao, Y. Guo, N. A. Romli, M. S. M. Kasihmuddin, W. Chen, M. A. Mansor, et al., GRAN3SAT: creating flexible higher-order logic satisfiability in the discrete Hopfield neural network, *Mathematics*, **10** (2022), 1899. <https://doi.org/10.3390/math10111899>
43. S. Sathasivam, W. A. T. W. Abdullah, Logic learning in Hopfield networks, *ArXiv*, 2008. <https://doi.org/10.48550/arXiv.0804.4075>
44. G. Pinkas, Symmetric neural networks and propositional logic satisfiability, *Neural Comput.*, **3** (1991), 282–291. <https://doi.org/10.1162/neco.1991.3.2.282>
45. P. Ong, Z. Zainuddin, Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction, *Appl. Soft Comput.*, **80** (2019), 374–386. <https://doi.org/10.1016/j.asoc.2019.04.016>
46. G. Maguolo, L. Nanni, S. Ghidoni, Ensemble of convolutional neural networks trained with different activation functions, *Expert Syst. Appl.*, **166** (2021), 114048. <https://doi.org/10.1016/j.eswa.2020.114048>
47. D. L. Elliott, A better activation function for artificial neural networks, *Unive. Md.*, 1993.
48. Y. Zhang, P. Li, C. Xu, X. Peng, R. Qiao, Investigating the effects of a fractional operator on the evolution of the ENSO model: bifurcations, stability and numerical analysis, *Fractal Fract.*, **7** (2023), 602. <https://doi.org/10.3390/fractalfract7080602>
49. C. Xu, Z. Liu, P. Li, J. Yan, L. Yao, Bifurcation mechanism for fractional-order three-triangle multi-delayed neural networks, *Neural Process. Lett.*, **55** (2023), 6125–6151. <https://doi.org/10.1007/s11063-022-11130-y>

50. P. Li, Y. Lu, C. Xu, J. Ren, Insight into Hopf bifurcation and control methods in fractional order BAM neural networks incorporating symmetric structure and delay, *Cogn. Comput.*, **15** (2023), 1825–1867. <https://doi.org/10.1007/s12559-023-10155-2>
51. P. Li, X. Peng, C. Xu, L. Han, S. Shi, Novel extended mixed controller design for bifurcation control of fractional-order Myc/E2F/miR-17-92 network model concerning delay, *Math. Methods Appl. Sci.*, **46** (2023), 18878–18898. <https://doi.org/10.1002/mma.9597>
52. A. G. Hussien, D. Oliva, E. H. Houssein, A. A. Juan, X. Yu, Binary whale optimization algorithm for dimensionality reduction, *Mathematics*, **8** (2020), 1821. <https://doi.org/10.3390/math8101821>
53. E. Pashaei, N. Aydin, Binary black hole algorithm for feature selection and classification on biological data, *Appl. Soft Comput.*, **56** (2017), 94–106. <https://doi.org/10.1016/j.asoc.2017.03.002>
54. N. A. Rusdi, M. S. M. Kasihmuddin, N. A. Romli, G. Manoharam, M. A. Mansor, Multi-unit discrete Hopfield neural network for higher order supervised learning through logic mining: optimal performance design and attribute selection, *J. King Saud Univ.*, **35** (2023), 101554. <https://doi.org/10.1016/j.jksuci.2023.101554>
55. S. A. Alzaeemi, S. Sathasivam, M. K. M. Ali, K. G. Tay, M. Velavan, Hybridized intelligent neural network optimization model for forecasting prices of rubber in Malaysia, *Comput. Syst. Sci. Eng.*, **47** (2023), 1471–1491. <https://doi.org/10.32604/csse.2023.037366>



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)