
Research article

Synergizing intelligence and knowledge discovery: Hybrid black hole algorithm for optimizing discrete Hopfield neural network with negative based systematic satisfiability

Nur ‘Afifah Rusdi^{1,2}, Nur Ezlin Zamri³, Mohd Shareduwan Mohd Kasihmuiddin¹, Nurul Atiqah Romli¹, Gaeithry Manoharam¹, Suad Abdeen¹ and Mohd. Asyraf Mansor^{4,*}

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

² Institute of Engineering Mathematics, Universiti Malaysia Perlis, Kampus Pauh Putra, 02600 Arau, Perlis, Malaysia

³ Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

⁴ School of Distance Education, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

* **Correspondence:** Email: asyrafman@usm.my; Tel: +6046533935.

Abstract: The current systematic logical rules in the Discrete Hopfield Neural Network encounter significant challenges, including repetitive final neuron states that lead to the issue of overfitting. Furthermore, the systematic logical rules neglect the impact on the appearance of negative literals within the logical structure, and most recent efforts have primarily focused on improving the learning capabilities of the network, which could potentially limit its overall efficiency. To tackle the limitation, we introduced a Negative Based Higher Order Systematic Logic to the network, imposing restriction on the appearance of negative literals within the clauses. Additionally, a Hybrid Black Hole Algorithm was proposed in the retrieval phase to optimize the final neuron states. This ensured that the optimized states achieved maximum diversity and reach global minima solutions with the lowest similarity index, thereby enhancing the overall performance of the network. The results illustrated that the proposed model can achieve up to 10,000 diversified and global solutions with an average similarity index of 0.09. The findings indicated that the optimized final neuron states are in optimal configurations. Based on the findings, the development of the new systematic SAT and the implementation of the Hybrid Black Hole algorithm to optimize the retrieval capabilities of DHNN to achieve multi-objective

functions result in updated final neuron states with high diversity, high attainment of global minima solutions, and produces states with a low similarity index. Consequently, this proposed model could be extended for logic mining applications to tackle classification tasks. The optimized final neuron states will enhance the retrieval of high-quality induced logic, which is effective for classification and knowledge extraction.

Keywords: negative based higher order satisfiability; discrete Hopfield neural network; multi-objectives; hybrid black hole algorithm

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

1. Introduction

In the domain of computational intelligence, researchers have utilized the remarkable capabilities of Artificial Neural Networks ANNs to develop intelligent models capable of learning, reasoning and decision-making. The structure of ANNs is subject to variation, influencing the network's capabilities and efficiency in processing and learning from data [1]. As a flexible mathematical model, ANNs sparked a revolution in diverse fields. This transformative impact has led to significant progress in specific areas including image processing [2], speech recognition [3], time series analysis [4], traffic classification [5] and optimization of wireless networks [6]. Building upon the flexible capabilities of ANNs, one variant of ANNs is Discrete Hopfield Neural Network (DHNN) [7]. This network employed local field equation to update the neuron states, offering a potential solution to optimization problems. DHNN have gained considerable attention due to its straightforward structure, resembling of single-layer feedback neural networks [8]. These single-layer networks typically consist of input and output layers without hidden layers. The synaptic weights represent the connections between the neurons and their ability to learn and store information. Specifically, the simple structure of DHNN has proven effective in tackling various optimization challenges [9,10]. While acknowledging the capabilities and efficiency of DHNN in solving optimization problems, there is a very limited strategy to govern the structure of DHNN, primarily due to the predominant focus on achieving optimal neuron states. This often leaves DHNN functioning as a black box model. The term "black box model" implies that the inner workings of the network are not well-understood. This lack of interpretability can lead researchers into uncertainty, potentially optimizing the wrong aspects or facing challenges in determining what should be optimized. By refining these models, the capacity of the network to effectively store and retrieve patterns can be enhanced.

Abdullah [11] addressed this gap by further advancing the DHNN through the incorporation of the concept of satisfiability (SAT) ensuring proper neuron connectivity without compromising network behavior. This approach utilized the structure of Horn Satisfiability (HornSAT), where each clause has at most one positive literal to represent the neurons in DHNN. This milestone not only established a novel approach to neuron representation but also laid the foundation for the subsequent development of the Wan Abdullah (WA) method. This WA method involves comparing the cost function and Lyapunov energy function to find the synaptic weight values. This innovation sparked a new wave of research perspectives, which led to the recognition of SAT as systematic SAT and non-systematic SAT. Kasihmuddin et al. [12] introduced the incorporation of 2 Satisfiability (2SAT) into DHNN. This implementation resulted in a notable increase in the states that achieve global minima energy.

Moreover, Sathasivam et al. [13] introduced the first non-systematic SAT namely, Random 2 Satisfiability (RAN2SAT), which incorporates both first and second-order logics. The flexibility to represent the number of literals in each clause results in enhanced logical variation during the learning phase. However, in minimizing the cost function, the task of finding a satisfied interpretation becomes increasingly challenging as the number of neurons increases. This is due to the presence of more first order logic, which has a low probability of getting satisfied interpretation thereby contributing to more logical inconsistency. Expanding on this idea, Karim et al. [14] extended RAN2SAT to include third-order logic, known as Random 3 Satisfiability (RAN3SAT). The proposed RAN3SAT comprises three different logical combinations, $(k = 1, 3, k = 2, 3, \text{ and } k = 1, 2, 3)$. The simulation results indicated that the combination of second and third-order logics offers the most promising results such that this combination is more consistent in obtaining lower learning and testing errors. This finding led to the introduction of another new non-systematic logical rule, namely Major 2 Satisfiability (MAJ2SAT) in DHNN [15]. While the $(k = 2, 3)$ ratio remains consistent in RAN3SAT, the structure of MAJ2SAT diverged by incorporating a bias 2SAT clauses compared to 3SAT clauses. Simulated results demonstrated the successful incorporation of MAJ2SAT into DHNN, as the model exhibited the capability to generate global minima solutions and retrieve the optimal final neuron states. Conversely, Zamri et al. [16] proposed another non-systematic logic that represents a distinct perspective. In this work, Weighted Random k Satisfiability (r2SAT) was proposed with inclusion of a weighted ratio involving negative literals. r2SAT managed to obtain a good performance as compared to existing SAT which indicates that having dynamic distribution of negative literals will facilitate producing global minima solutions with diverse final neuron states.

However, the role of negative literals in systematic SAT remain unclear as current works based on systematic SAT failed to emphasize the influence of negative literals within the clauses. For example, Mansor et al. [17] capitalizes 3 Satisfiability (3SAT) structure to represent the neurons in DHNN. Despite the great performance of higher-order systematic logic, this model neglects the negative links in the neuron connections. The formulation of this logical rule only considers the random distribution of positive and negative literals within e clauses. No attention has been given to investigate the impact of negative synaptic weight distribution on the retrieval of final neuron states with global energy. Negative literals are often neglected due to their association with faults or errors. It is crucial to mention that variations in synaptic weights in terms of magnitude are essential for accurately representing real-life classification problems. Therefore, introducing a systematic SAT that promotes the appearance of negative literals results in diverse final neuron states produced, making DHNN a more effective computational system for any optimization problem.

In enhancing the overall performance of DHNN, various initiatives have been proposed by recent researchers to optimize the learning phase of DHNN [12,15,17–20]. However, there has been limited attention among researchers towards optimizing the retrieval phase. The most recent work that focused on enhancing the retrieval capabilities of DHNN was proposed by Kasihmuddin et al. [21]. In this work, an Estimation of Distribution Algorithm (EDA) was employed to optimize the retrieval phase. Specifically, a univariate marginal Gaussian distribution probability model was used to introduce minor perturbations to the neurons, which in turns will reduce the possible local minima solutions. However, the diversification of the optimized final neuron states in terms of negativity remains unclear and the proposed model failed to guarantee all the optimized final neuron states attained global minima energy. Additionally, the dissimilarity of the optimized final neuron states, particularly focusing on the negative states, also remains uncertain.

These gaps can be addressed by introducing a new metaheuristic approach to optimize the

retrieved final neuron states and tackle the multi-objective functions. In response to this, one simple algorithm that has gained attention from many researchers is known as the Black Hole algorithm (BHA). BHA is a natural phenomenon-based algorithm that draws inspiration from the dynamics of black holes in space. It efficiently solves complex optimization problems by simulating how a black hole attracts nearby stars to find optimal solutions. As introduced by Hatamlou [22], BHA has been successfully applied to data clustering challenges using six different benchmark datasets with varying levels of complexity. When compared to other optimization algorithms, BHA outperformed them by generating high-quality solutions with low standard deviation. This success paves the way for implementing BHA into applications such as machine learning [23], image processing, aircraft systems [24] and network applications [25]. In another development, Pashaei & Aydin [26] introduced the Binary Black Hole Algorithm (BBHA) as a streamlined solution for feature selection problems in biological datasets. BBHA addresses the complexities associated with conventional methods by minimizing parameter requirements, offering remarkable computational speed and straightforward implementation. BBHA was used as part of a method for feature selection. The research discovered that the utilization of the hyperbolic tangent function enabled BBHA to effectively overcome challenges associated with feature selection in text, image and biomedical data. This approach outperformed other algorithms considered in the study.

Despite BHA showing promise in various contexts, the incorporation of BBHA into DHNN for optimization problems remains uncertain. This gap presents an interesting opportunity for further investigation, focusing on examining the effectiveness of incorporating BBHA to optimize the retrieval phase of DHNN. It is worth noting that in terms of solving logic satisfiability in DHNN, this work represents the first attempt to optimize the retrieval phase of DHNN in achieving multi-objective functions, which consequently improves the quality of retrieved final neuron states. Thus, this paper presents a new logical rule referred to as Negative Based Higher Order Systematic logic. This logical rule emphasizes the restriction in the appearance of negative literals within the clauses of 3SAT. By incorporating this new logical rule, we can effectively model the neuron in DHNN, leading to enhanced performance of higher order systematic logic. An effective model will result in a higher satisfied interpretation and ultimately optimize the learning phase. Consequently, an optimal synaptic weight will contribute to an improved retrieval phase. By implementing the Hybrid Binary Black Hole Algorithm during the retrieval phase of DHNN, this study aims to enhance the diversification of the states in terms of negativity while attaining global minima solutions and minimizing the similarity index. Therefore, the contribution of this paper is as follows:

1. To formulate a new higher order systematic logical rule namely Negative Based Higher Order Systematic Logic as a symbolic neuron representation in Discrete Hopfield Neural Network. By incorporating the proposed logical rule, the neuron in Discrete Hopfield Neural Network is effectively modelled leading to improved performance of higher order systematic logic.
2. To propose Hybrid Binary Black Hole Algorithm in the retrieval phase of Discrete Hopfield Neural Network by implementing Election Algorithm as a learning algorithm. In this context, the proposed metaheuristic algorithm will be utilized to optimize the final neuron states, aiming to enhance the diversification of the states in terms of negativity and attain global minima solutions while having lowest similarity index. Thus, implementation of this proposed algorithm will guarantee the optimized final neuron states will be beneficial in the perspective of logic mining.
3. To evaluate the performance of Negative Based Higher Order Systematic Logic in Discrete

Hopfield Neural Networks and analyze the effectiveness of Hybrid Binary Black Hole Algorithm in optimizing the final neuron states obtained by the network. The performance analysis will be divided into two parts. First, the performance of the proposed logic with different logical rules will be evaluated using various metrics such as learning error, testing error and similarity analysis. Second, we examine how well Hybrid Black Hole Algorithm performs in optimizing the retrieval capabilities of Discrete Hopfield Neural Network.

To effectively fulfil all the objectives, we begin by discussing its motivation in Section 2. In Section 3, we describe the proposed logical representation and discuss the implementation of the logical structure for DHNN in Section 4. Then, we explain the proposed multi-objective functions in the retrieval phase of DHNN in Section 5. In Section 6, we explain the proposed Hybrid Black Hole Algorithm in achieving the multi-objective functions. Next, the experimental framework will be explained in Section 7, and we focus on the findings and analysis in Section 8. Finally, the conclusions drawn from the study and suggestions for future research exploration are provided in Section 9.

2. Motivation

2.1. Ineffective structure of logical representation

Logical rule is important to guarantee that the knowledge can be represented effectively without losing any information [27]. Note that each structural component on logical rule must be represented correctly before being encoded as symbolic rules into any computational systems. However, much uncertainty exists about the generalizability characteristics of logical rule formulation to be formed in a specific manner or condition. In the context of systematic Satisfiability (SAT) in the DHNN models, existing works such as 2SAT by Kasihmuddin et al. [12] and 3SAT by Mansor et al. [17] disregarded the distribution of negative literals throughout respective SAT logical rules. This implied that the distribution of negative literals is set randomly. When the SAT model operates with randomized positive and negative activated neuron connections, the interpretability quality of DHNN is questioned. This is because, with no demand and unclearness on the distribution of either negative or positive literals, the SAT model provided unclear information on what logical relations are significant to be directed towards optimal production of the final neuron states. Consequently, the practicality of embedding SAT as logical rules into DHNN declines which depicts the importance for the SAT to have some level of negation control to ensure the direction of the retrieval phase can be strengthened. Unfortunately, little attention was given in the development of formulating logic with specific conditions before being encoded as an actual symbolic language. Therefore, we introduced Negative Based Higher Order Systematic logic or NR3SAT as a logical rule to control the appearance of negative literals within the clauses. Note that this is the first approach on introducing higher-order systematic logical rule that emphasize the restriction on the appearance of negative literals in the formulation of SAT.

2.2. Limited diversification of states concerning negativity

The final neuron states retrieved during the retrieval phase of DHNN are significant as the states represent the network's pattern and act as an indicator of successful pattern retrieval. Evaluating the quality of the retrieved final neuron states often involves assessing the diversity of the solutions. Previous studies in the literature typically assessed solution diversification through similarity index

analysis. In this context, a low similarity index indicates high dissimilarity between the retrieved and benchmark states, contributing to diversified solutions. For instance, Karim et al. [14], employed measures such as the Jaccard index, Kulczynski Measure and Ochiai coefficient to evaluate the quality of the retrieved solutions. Additionally, Roslan et al. [28] categorizes a solution as diverse from the benchmark if it surpasses the specified diversity tolerance value $tol_d = 0.1$. Another study by Alway et al. [29] defined diversified solutions in terms of dissimilarity compared to benchmark states, categorizing retrieved final neuron states as diversified if there is at least a 10% dissimilarity to the benchmark states. However, existing works have predominantly overlooked a crucial aspect which is diversification concerning solution string. While each solution string may differ from the benchmark states, the diversification of the solution string in terms of negativity remains uncertain. This oversight can lead to high neuron overfitting due to the low impact of diversity. Therefore, this paper proposes the incorporation of the Hybrid Black Hole Algorithm to optimize the retrieved final neuron states, ensuring that the optimized states achieve the desired proportion of negative states in each solution string while maintaining achieve maximum global solutions.

2.3. Inefficient discrete Hopfield neural network

According to Gharehchopogh et al. [30], optimization refers to the process of determining the most favorable values for decision variables to achieve either the minimum or maximum value of a given objective function. The primary aim of optimization techniques is to thoroughly explore and analyze the search space, ultimately finding an optimal solution for a specific problem [31]. In the context of logic satisfiability within the DHNN framework, current research predominantly directs its efforts toward improving the learning phase of the network. Numerous researchers employ metaheuristic algorithms to bolster the DHNN framework, resulting in optimal synaptic weight being retrieved during the learning phase [18,20,32,33]. If the proposed learning algorithm fails to attain the satisfied interpretation, synaptic weights will be generated randomly. However, despite the achievement of optimal synaptic weights leading to the retrieval of global solutions, uncertainties persist regarding the quality of the retrieved final neuron states, particularly within the solution strings. This can be supported by the work on systematic SAT, which encounters an overfitting issue, even though the solutions obtained are global [34]. In another development, Kasihmuddin et al. [21] utilized an Estimation of Distribution Algorithm (EDA) to optimize the retrieval phase of DHNN but encountered suboptimal results in the learning phase due to relying on Exhaustive Search (ES) as a learning algorithm. Previous studies have focused on improving either the learning or retrieval phases of DHNN individually, but little attention has been given to enhancing both simultaneously. This provides motivation to enhance the efficiency of the DHNN framework by optimizing both learning and retrieval phase. Thus, Election Algorithm inspired by Sathasivam et al. [34] will be employed during the learning phase to enhance the learning capabilities of DHNN in minimizing the cost function. Moreover, Hybrid Black Hole Algorithm will be proposed to optimize the retrieval phase of DHNN in achieving the multi-objectives functions. The aim of introducing these objectives is to ensure the optimized final neurons state are in optimal configurations. The optimization will focus on generating optimal final neuron states by considering the diversification of the solutions string, the attainment of global solutions and the lowest similarity index.

3. Negative based higher order systematic satisfiability

Negative Based Higher Order Systematic Satisfiability referred to as NR3SAT is a logical representation that consists of higher order logic 3SAT with a restriction on the appearance of negative literals within the clauses. The main features of the proposed NR3SAT are presented as follows:

- Set of m variables represent as $q_1, q_2, q_3, \dots, q_m$, which holds bipolar values of $q_i \in \{1, -1\}$.
- Set of literals, $q_i \in \{q_i, \neg q_i\}$ such that q_i and $\neg q_i$ represents positive and negative literals, respectively.
- Set of n definite third order clauses represent as $Q_1^*, Q_2^*, Q_3^*, \dots, Q_n^*$ such that each clause, Q_i^* consists of at least one negative literals in a clause and $n \in \mathbb{Z}^+$.
- Each clause of Q_i^* is restricted to only three literals such that all clauses are connected to logical AND (\wedge), and literals in each clause are connected to logical OR (\vee).

Using the information gathered in (a)–(d), the general formulation of NR3SAT is introduced as follows:

$$\ell_{NR3SAT} = \wedge_{i=1}^n Q_i^*, \quad (1)$$

where the possible combination of ℓ_{NR3SAT} is defined as in Eq (2) such that:

$$Q_i^* \in \{2^k - 1\}, \quad k = 3. \quad (2)$$

These combinations of clauses do not consider all positive literals in a clause such that $Q_i^* \notin (q_i \vee q_j \vee q_k)$. By considering both Eq (1) and Eq (2), the possible structure of ℓ_{NR3SAT} for the minimum number of neurons can be represented as follows:

$$\ell_{NR3SAT} = (q_1 \vee q_2 \vee \neg q_3) \wedge (q_4 \vee \neg q_5 \vee \neg q_6) \wedge (q_7 \vee \neg q_8 \vee q_9). \quad (3)$$

4. Negative based higher order systematic satisfiability in discrete Hopfield neural network

Incorporating NR3SAT into the learning phase of DHNN, often denoted as ℓ_{NR3SAT} serves as the central objective, aiming to minimize the cost function of the network corresponding to the logic. This minimization of the cost function can be obtained by reducing the logical inconsistency of ℓ_{NR3SAT} . Therefore, the cost function, $E_{\ell_{NR3SAT}}$ of ℓ_{NR3SAT} can be deduced as shown in Eq (4).

$$E_{\ell_{NR3SAT}} = \sum_{i=1}^n \prod_{j=1}^3 Z_{ij}. \quad (4)$$

Referring to Eq (4), n is the number of clause in ℓ_{NR3SAT} and Z_{ij} denoted as the inconsistency of ℓ_{NR3SAT} . In this context, the inconsistency of ℓ_{NR3SAT} can be derived by taking the negation of ℓ_{NR3SAT} and then expand it using Eq (5) such that $\neg q_i$ is the negation of the literal in ℓ_{NR3SAT} . It is also crucial to emphasize that by taking the negation of ℓ_{NR3SAT} , the logical AND operation will signify the multiplication of literals within the clauses, while the logical OR operation

signifies the addition of clauses to other clauses.

$$Z_{ij} = \begin{cases} \frac{1}{2}(1 - S_{q_i}), & \text{if } \neg q_i \\ \frac{1}{2}(1 + S_{q_i}), & \text{otherwise} \end{cases}. \quad (5)$$

Note that, in reducing the logical inconsistency, the value of $E_{\ell_{NR3SAT}}$ reflects the count of unsatisfied clauses. If the number of unsatisfied clauses increases, the value of $E_{\ell_{NR3SAT}}$ will also increase. Therefore, the minimization of the cost function only obtained when $E_{\ell_{NR3SAT}} = 0$ indicates that all clauses in ℓ_{NR3SAT} are satisfied. In this case, Exhaustive Search (ES) will be implemented to find a satisfied interpretation of ℓ_{NR3SAT} . ES is often referred to as the trial-and-error approach characterized by its comprehensive exploration of all possible combinations of subsets within a given set. The main motivation behind utilizing ES as a learning algorithm is it is easy to be implemented. Additionally, most researchers in the literature including recent works by [35–37] employed ES to assess the capabilities and the stability of newly proposed symbolic rule in DHNN. Thus, this makes ES a valuable and commonly used approach for evaluating and benchmarking the performance of novel symbolic rules in DHNN. Using the concept of generate and validate, this approach enumerates all potential solutions within the search space until it identifies the optimal solution. Therefore, ES will be employed to determine the maximum number of satisfied clauses as described in Eq (6) such that Q_i^* referred to the clause of ℓ_{NR3SAT} and NC is a total clause. A satisfied interpretation of Q_i^* clause can be defined as in Eq (7).

$$f_{NR3SAT(ES)} = \max \left[\sum_{i=0}^{NC} Q_i^* \right], \quad (6)$$

$$Q_i^* = \begin{cases} 1, & \text{Satisfied} \\ -1, & \text{Unsatisfied} \end{cases}. \quad (7)$$

In this context, ES plays a crucial role in locating a satisfied interpretation that optimizes the fitness function in Eq (6) which correspond to the minimization of the cost function. Then, by considering all connection of ℓ_{NR3SAT} , the Lyapunov energy function of DHNN can be written as outlined in Eq (8).

$$H_{\ell_{NR3SAT}} = -\frac{1}{3} \sum_{\substack{i=1, \\ i \neq j \\ i \neq k}}^N \sum_{\substack{j=1, \\ j \neq i \\ j \neq k}}^N \sum_{\substack{k=1, \\ k \neq i \\ k \neq j}}^N W_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2} \sum_{\substack{i=1, \\ i \neq j}}^N \sum_{\substack{j=1, \\ j \neq i}}^N W_{ij}^{(2)} S_i S_j - \sum_{i=1}^N W_i S_i. \quad (8)$$

Hence, the values of the corresponding synaptic weights W_{ij} , $W_{ij}^{(2)}$ and $W_{ijk}^{(3)}$ can be obtained using the Wan Abdullah (WA) method through a direct comparison between the cost function presented in Eq (4) and the Lyapunov energy function of DHNN, as outlined in Eq (8) [11]. These resultant values can then be stored in a Content-Addressable Memory (CAM). This weight holds the properties of always being symmetrical ($W_{ij} = W_{ji}$) and there are no self-connections

among them. Hence, the anticipated global minimum energy can be achieved by substituting the synaptic weight values stored in CAM into Eq (8), by utilizing a satisfied interpretation. As an illustration, one such satisfied interpretation is provided below:

$$S_{q_1} = S_{q_2} = S_{q_3} = S_{q_4} = S_{q_5} = S_{q_6} = S_{q_7} = S_{q_8} = S_{q_9} = 1. \quad (9)$$

By considering the neuron states as in Eq (9) into Eq (8), the expected global minimum energy of ℓ_{NR3SAT} is $H_{\ell_{NR3SAT}}^{\min} = -0.375$. This value is defined as the lowest attainable energy state that the network can reach and will serve as a benchmark to evaluate the final energy value [38]. During the retrieval phase of DHNN, the synaptic weight values are used to update the local field of the network. When dealing with higher order neurons connections, the local field denoted as h_i can be defined as in Eq (10).

$$h_i = \sum_{\substack{j=1, k=1, \\ j \neq k, k \neq j}}^N \sum_{j=1}^N W_{ijk}^{(3)} S_j S_k + \sum_{j=1}^N W_{ij}^{(2)} S_j + W_i. \quad (10)$$

In this case, optimal synaptic weights acquired during the learning phase play a crucial role in ensuring that the network converges to optimal final neuron states during retrieval phase. Then, Hyperbolic Tangent Activation Function (HTAF), as detailed in Eq (11) serves as a squashing mechanism, transforming the values obtained in Eq (10) into bipolar states, typically either 1 or -1. Thus, the final neuron states, S_i^f are updated as demonstrated in Eq (12).

$$\tanh(h_i) = \frac{e^{h_i} - e^{-h_i}}{e^{h_i} + e^{-h_i}}, \quad (11)$$

$$S_i^f = \begin{cases} 1, & \text{if } \tanh(h_i) \geq 0 \\ -1, & \text{otherwise} \end{cases}. \quad (12)$$

Finally, final energy states were obtained using Eq (8). Then, the difference between the final energy and minimum energy will be evaluated based on Eq (13) as follows:

$$\left| H_{\ell_{NR3SAT}} - H_{\ell_{NR3SAT}}^{\min} \right| \leq Tol. \quad (13)$$

If the difference between global minimum energy and final energy falls within the range defined by Tol such that $Tol = 0.001$, ℓ_{NR3SAT} achieve global minima energy confirming it is successfully be embedded in DHNN. In this case, Tol is set to a specific value as recommended by [35–37] to reduce the statistical errors within the solutions. The pseudocode of $\ell_{NR3SAT(ES)}$ is presented in Algorithm 1, and the schematic diagram of DHNN is illustrated as in Figure 1.

5. Multi-objective functions in the retrieval phase of DHNN

We introduce a set of objective functions designed to ensure that the final neuron states obtained

are in their optimal configurations. Therefore, our aim of the first objective is to obtain diversified final neuron states. Now, one may wonder, why is diversity essential? To address this question, the need for diversified final neuron states arises from the necessity to create more dynamic induced logic. This involves introducing variations particularly in terms of negativity to effectively tackle real-life classification problems. Diverse solutions provide a wider range of responses, enhancing adaptability, and robustness in problem-solving scenarios. Second, it is crucial to ensure that all the diversified solutions generated are in a global state.

Algorithm 1. Pseudo-code of the general process in DHNN

```

1  Begin
2      {Learning Phase: Exhaustive Search}
3      Generate  $\ell_{NR3SAT}$  initial neuron states randomly;
4      do
5          Minimize cost function,  $E_{\ell_{NR3SAT}}$  ;
6          while  $(\varphi \leq NH)$  or  $f_i = f_{\max}$  do
7              for  $i=0$  to  $N-1$ 
8                  for  $j=i+1$  to  $N$  do
9                      Learn  $\ell_{NR3SAT}$  ;
10                     if  $f_{i+1} > f_i$  then
11                          $f_{i+1} = f_{i+1}$ ;
12                     else
13                          $f_{i+1} = f_i$ ;
14                     end if
15                 end for
16             end for
17         end while
18     do
19         Calculate synaptic weight and store in CAM;
20         Calculate expected global minimum energy,  $H_{\ell_{NR3SAT}}^{\min}$  ;
21     {Retrieval Phase}
22     Initialize random neuron states;
23     do
24         Calculate local field and update final neuron states using HTAF;
25         Calculate final neuron energy,  $H_{\ell_{NR3SAT}}$  ;
26         Verify global or local minimum energy;
27         if  $|H_{\ell_{NR3SAT}} - H_{\ell_{NR3SAT}}^{\min}| \leq Tol$  then
28             Global minimum energy;
29         else
30             Local minimum energy;
31 End

```

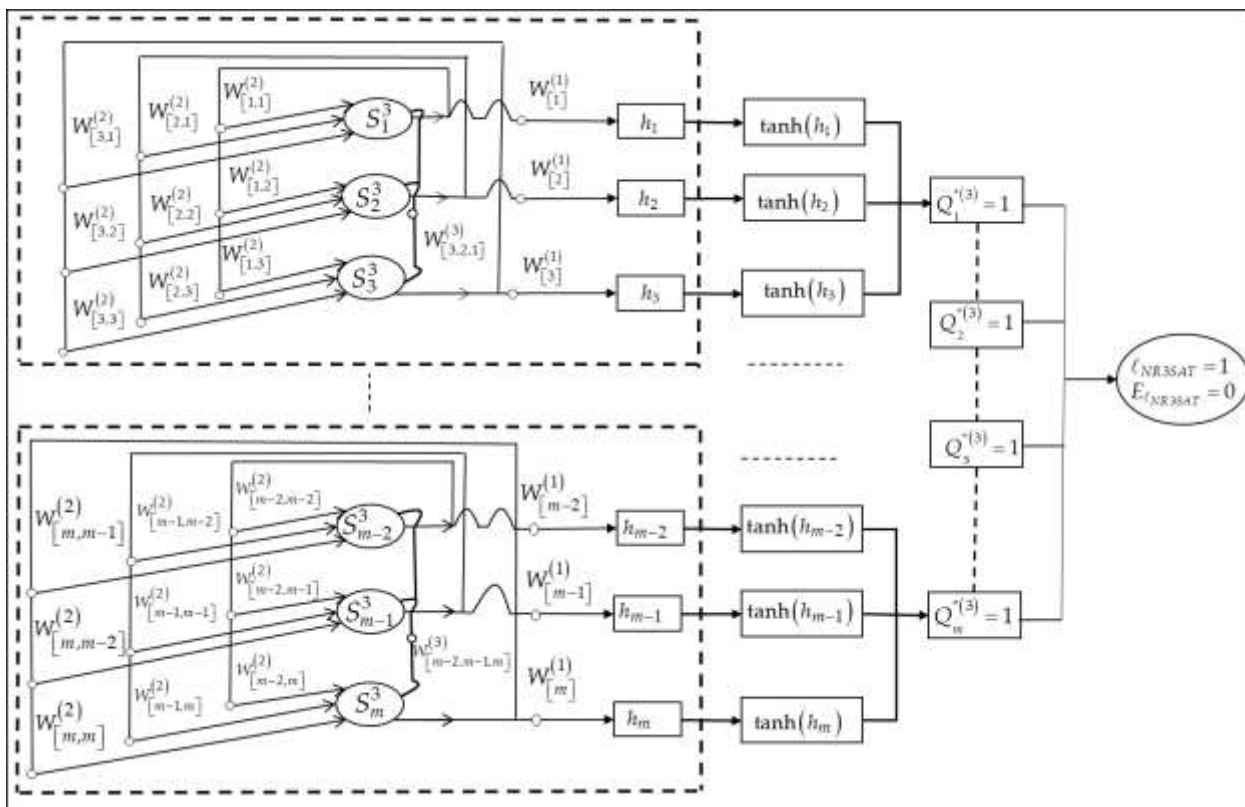


Figure 1. Schematic diagram of NR3SAT in DHNN.

Achieving global minima solutions signify that the proposed logic and algorithm are successfully implemented into the DHNN framework. Last, the third objective of this phase aims to ensure the lowest similarity index is obtained. Therefore, the novelty of this study lies in optimizing these three objectives: (i) Diversification in terms of negativity, (ii) attainment of a global minima solutions, and (iii) achieving a low similarity index. Mathematically, the proposed multi-objective functions can be generalized as follows:

$$F(f_D, f_Z, RT), \tag{14}$$

such that

$$F = \max(f_D), \tag{15}$$

$$F = \max(f_Z), \tag{16}$$

$$F = \min(RT). \tag{17}$$

To accomplish all these objectives, the implementation of a metaheuristic algorithm in the retrieval phase of DHNN is deemed essential. The upcoming section provides a comprehensive explanation of how the proposed metaheuristic algorithm can successfully fulfil all the objectives

outlined in Eqs (15)–(17). This algorithm plays a pivotal role in optimizing final neuron states, ensuring diversified and global solutions with low similarity indices, thus enhancing the overall performance and effectiveness of the DHNN framework.

6. The proposed hybrid black hole algorithm in optimizing retrieval phase of DHNN

In this section, we provide a comprehensive explanation of the optimization process in the retrieval phase of DHNN. To ensure the optimal utilization of synaptic weights during the retrieval phase, the Election Algorithm (EA) is employed during the learning phase to enhance the process of finding satisfied interpretation and consequently minimizing the cost function. EA is a valuable addition as it guarantees the achievement of optimal synaptic weights. Recent research by Abubakar & Danrimi [39], Roslan et al. [28], and Someetheram et al. [33] have demonstrated the effectiveness of EA as a learning algorithm to minimize the cost function within DHNN. The implementation of EA significantly enhances the convergence of the network towards minimizing the cost function. Then, an enhanced optimization algorithm called Hybrid Black Hole Algorithm (HBHA) will be employed to optimize the final neuron states. The primary goal was to ensure that the optimized final neuron states align with the proposed multi-objective function as discussed in the previous section. The HBHA process begins with the initialization of a population of stars, as outlined in Eq (18) as follows:

$$Star_i(S_i^f) = \begin{cases} 1, & \text{if } \tanh(h_i) \geq 0 \\ -1, & \text{otherwise} \end{cases} \quad (18)$$

Referring to Eq (18), the variable S_i^f represent the final neuron states to be optimized where $S_i^f \in [1, NN]$ and $Star_i \in [1, NT]$. In this context, NN referred to the total number of neurons defined and NT is referring to the number of trials declared in the process. These final neuron states were obtained by squashing the updated local field, h_i values using tangent hyperbolic function. Mathematically, the generalized initial star population, $P[Star_i(S_i^f)]$ can be derived as in Eq (19).

$$\begin{aligned} Star_1 : & S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad S_5^f \quad \dots \quad S_{NN}^f \\ Star_2 : & S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad S_5^f \quad \dots \quad S_{NN}^f \\ Star_3 : & S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad S_5^f \quad \dots \quad S_{NN}^f \quad . \\ & \vdots \\ Star_{NT} : & S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad S_5^f \quad \dots \quad S_{NN}^f \end{aligned} \quad (19)$$

After establishing the initial population, the HBHA utilizes the first two objective functions defined in Eqs (15) and (16) which involve diversifying the final neuron states in terms of negativity and achieving global minima solutions. In this context, f_D refers to the diversity fitness of ℓ_{NR3SAT} while f_Z denotes the fitness value associated with the attainment of global solutions. The diversification of final neuron states is a critical objective. It involves ensuring that the states of neurons in the network exhibit a wide range of patterns and behaviors rather than being uniform or repetitive. To achieve this objective, diversification is defined as the variation in terms of negativity, primarily derived from clauses. In essence, it focuses on diversifying the states of the literals within

the clauses, which is pivotal for enhancing the diversity of the logical rules within the network. This can be done by setting the percentage of negativity, $m\%$, that is needed and HBHA will be implemented to ensure the optimized final neuron states achieved at least $m\%$ of the clauses containing at least one negated state for each $Star_i$. Thus, the diversity fitness can be declared as follows:

$$f_D = m \left[\sum_{i=1}^n f_{D_i} (Q_i^*) \right], \quad (20)$$

such that

$$f_{D_i} (Q_i^*) = \begin{cases} 1, & \text{if } (S_i^f, S_i^f, S_i^f) \neq 1 \\ 0, & \text{otherwise} \end{cases}. \quad (21)$$

From the perspective of achieving global solutions, one alternative approach to determine whether the states are global or local is by computing the satisfied interpretation of each $Star_i$. In this context, each $Star_i$ is said to be global if the star can attain the maximum satisfied interpretation which is associated with the total number of clauses. The generalization of the fitness value associated with the attainment of a global solution can be defined in Eqs (22) and (23) as follows:

$$f_Z = \sum_{i=1}^n f_{Z_i} (Q_i^*), \quad (22)$$

such that

$$f_{Z_i} (Q_i^*) = \begin{cases} 1, & \text{if satisfied} \\ 0, & \text{otherwise} \end{cases}. \quad (23)$$

By defining the fitness function within the HBHA, fitness values can be calculated for each $Star_i$ in the population. Subsequently, based on these fitness values, the $Star_i$ with the highest fitness value which achieve both maximum diversity and global solution is selected as the Black Hole, $Star_i^{BH} (S_i^f)$, while the remaining $Star_i$ will be considered as normal stars. To enhance the exploration of the optimal solution within the search space, a global search operator called Star Replication is proposed in this research. This process draws inspiration from the cloning process in the Clonal Selection Algorithm, which is based on the natural processes of the immune system, explaining how the immune system generates antibodies to combat antigens. In the context of the HBHA, by replicating these stars, a balance is struck between exploring the solution space and exploiting promising solutions, a crucial aspect of effective optimization. The new population of stars after replicating those with the highest fitness can be formulated as follows:

$$Star_i^N (S_i^f) = \begin{cases} Star_i^R (S_i^f), & \text{if } Star_i^{BH} (S_i^f), \\ Star_i (S_i^f), & \text{otherwise} \end{cases}, \quad (24)$$

such that

$$n \left[Star_i^R(S_i^f) \right] = \beta * P \left[Star_i(S_i^f) \right]. \quad (25)$$

According to Eq (24), the $Star_i^{BH}(S_i^f)$ will be replicated to enhance the ability of HBHA to efficiently discover optimal solutions. The number of replicated star, $n \left[Star_i^R(S_i^f) \right]$ can be determined based on Eq (25) such that β represent the replicating rate defined within the range of 0 and 1. By considering the replicated stars denoted as $Star_i^R(S_i^f)$, the total number of population stars $P \left[Star_i(S_i^f) \right]$ will increase compared to the initial population of stars. In this new population, the $Star_i^R(S_i^f)$ are treated like other normal stars, $Star_i(S_i^f)$. After establishing the new population of stars, $Star_i^N(S_i^f)$, the strong gravitational pull of $Star_i^{BH}(S_i^f)$ causes all other stars to gradually move closer to the position of $Star_i^{BH}(S_i^f)$. Consequently, the positions of the stars are updated using Eq (26) as follows:

$$Star_i(S_i^f(t+1)) = Star_i(S_i^f) + \theta \left[Star_i^{BH}(S_i^f) - Star_i(S_i^f) \right]. \quad (26)$$

Referring to Eq (26), $Star_i(S_i^f(t+1))$ represents the updated positions of stars in the next iteration, where θ is a random number defined within the range of 0 and 1. $Star_i(S_i^f)$ and $Star_i^{BH}(S_i^f)$ refer to the current positions of the stars and BH respectively. During this phase, the positions of $Star_i(S_i^f(t+1))$ exist in continuous form. To convert these continuous values of star positions into bipolar values of 1 and -1, a transfer function is essential. This transfer function will determine the probability of changing the elements of the position. In this context, the Hyperbolic Tangent function is employed to modify the positions of the stars, as described in Eqs (27) and (28). It is worth noting that this function belongs to the group of V-shaped transfer functions and has been found to exhibit good performance compared to other transfer functions [40]. This function plays a crucial role in discretizing the positions of the stars, making them compatible with the requirements of the optimization algorithm.

$$Star_i(S_i^f(t+1)) = \left| \tanh(S_i^f(t+1)) \right|, \quad (27)$$

such that

$$S_i^f(t+1) = \begin{cases} 1, & \text{if } Star_i(S_i^f(t+1)) > rand \\ -1, & \text{otherwise} \end{cases}. \quad (28)$$

After updating the position of the stars, the fitness values will be computed. If $F \left[Star_i(S_i^f(t+1)) \right] > F \left[Star_i^{BH}(S_i^f) \right]$, the $Star_i(S_i^f(t+1))$ will be assigned as new

$Star_i^{BH}(S_i^f)$ or else $Star_i^{BH}(S_i^f)$ remain unchanged. It is essential to highlight that as the stars move towards the BH, there is a possibility of crossing the event horizon. If a star crosses the event horizon, it will be absorbed by the BH and disappears. The radius of the event horizon denoted as R is defined as outlined in Eq (29).

$$R = \frac{F[Star_i^{BH}(S_i^f)]}{\sum_{i=1}^{NT} F[Star_i(S_i^f)]}. \quad (29)$$

To ascertain whether the stars cross the event horizon or not, the distance between the star and BH denotes as D , is computed as described in Eq (30).

$$D = Star_i^{BH}(S_i^f) - Star_i(S_i^f). \quad (30)$$

If $D < R$, it indicates that the stars have crossed the event horizon and will disappear. In this case, new stars will be generated randomly to replace those that have crossed the event horizon. This process involves the exploitation of stars towards optimal solutions while simultaneously ensuring the continuous exploration of the search space, even as some stars may be absorbed by the BH during the optimization process. Then, the process in HBHA will be repeated until it meets the defined stopping criteria. This can be either when the process reaches the maximum specified number of generations or when the star population reaches the maximum fitness, as defined in Eqs (15) and (16). The optimize final neuron states can be generalized as follows:

$$\begin{aligned} Star_1 : S_1^{fa} \quad S_2^{fa} \quad S_3^{fa} \quad S_4^{fa} \quad S_5^{fa} \quad \dots \quad S_{NN}^{fa} \\ Star_2 : S_1^{fa} \quad S_2^{fa} \quad S_3^{fa} \quad S_4^{fa} \quad S_5^{fa} \quad \dots \quad S_{NN}^{fa} \\ Star_3 : S_1^{fa} \quad S_2^{fa} \quad S_3^{fa} \quad S_4^{fa} \quad S_5^{fa} \quad \dots \quad S_{NN}^{fa} \\ \vdots \\ Star_{NT} : S_1^{fa} \quad S_2^{fa} \quad S_3^{fa} \quad S_4^{fa} \quad S_5^{fa} \quad \dots \quad S_{NN}^{fa} \end{aligned} \quad (31)$$

Next, to fulfil the third objective function as in Eq (17), Rogers and Tanimoto similarity index (RT) will be employed in this phase to measure the similarity of states before (S_i^f) and after (S_i^{fa}) optimization takes place for each string of $Star_i$. This phase is essential to evaluate how HBHA helps to improve the current final neuron states. If there is high similarity between (S_i^f) and (S_i^{fa}) , the implementation of HBHA in optimizing the retrieval phase of DHNN is considered unsuccessful since the optimized solutions are overfitted. The RT value between the neuron of $Star_i$ can be obtained by comparing the states before the optimization took place as in Eq (19) and states after the optimization took place as in Eq (31). By considering τ as the similarity index coefficients for states before and after optimization took place, the formulation of RT can be generalized as in Eqs (32)–(37).

$$\tau = (S_i^f, S_i^{fa}), \quad (32)$$

$$RT = \frac{a+d}{a+d+2(b+c)}, \quad (33)$$

such that

$$a = \begin{cases} 1, & \text{if } \tau = (1 \ 1) \\ 0, & \text{otherwise} \end{cases}, \quad (34)$$

$$b = \begin{cases} 1, & \text{if } \tau = (1 \ -1) \\ 0, & \text{otherwise} \end{cases}, \quad (35)$$

$$c = \begin{cases} 1, & \text{if } \tau = (-1 \ 1) \\ 0, & \text{otherwise} \end{cases}, \quad (36)$$

$$d = \begin{cases} 1, & \text{if } \tau = (-1 \ -1) \\ 0, & \text{otherwise} \end{cases}. \quad (37)$$

Referring to Eq (33), the lower value of RT indicates high dissimilarity between the states before and after the optimization took place. Therefore, the optimized final neuron states will be considered as new final neuron states (S_i^{nf}) of ℓ_{NR3SAT} if the states after being optimized can successfully achieve all three multi objectives derived as in Eqs (15)–(17). This can be generalized as follows:

$$S_i^{nf} \rightarrow \text{Obj}(f_D, f_Z, RT). \quad (38)$$

The pseudocode of HBHA is presented in Algorithm 2. Additionally, the flowchart of the proposed HBHA can be summarized as in Figures 2 and Figure 3 illustrated the flowchart of the proposed model in optimizing the learning phase and retrieval phase of DHNN.

7. Experimental framework

In assessing the effectiveness of the proposed model, this section offers a comprehensive overview and discussion of the experimental setup. The first subsection explained the simulation platform employed in this study, followed by a detailed explanation of the parameter setup utilized in both the learning and retrieval phases of DHNN. Subsequently, an overview of the performance metrics considered throughout this study is presented. The final section outlines the baseline model used in the learning phase and the baseline algorithms employed during the retrieval phase.

7.1. Simulation platform

In this paper, all simulations were executed using the open-source software DEV C++ (version 6.3) with Windows 10 operating system as a platform which equipped with an Intel Core i3 processor and 8GB of RAM. This deliberate choice was made to eliminate potential biases during both code execution and result recording. Consequently, the experiments were consistently performed using the

same compiler on the same computing device with identical processing power.

7.2. Data simulation

To assess the effectiveness of this proposed model, the simulated dataset utilized in this study was generated by randomly assigning bipolar values of 1 and -1, adhering to the logical structure of ℓ_{NR3SAT} . In this context, employing simulated data is a widespread practice for simulating and evaluating the performance of the proposed model, as demonstrated in the research conducted by [24,46,50,51].

7.3. Parameter setup

The parameter setup employed throughout this study will be explained, focusing on both the learning and retrieval phases of DHNN, as detailed in the upcoming subsections.

7.3.1. Parameter setup in the learning phase

In assessing the efficiency of the proposed Negative Based Higher Order Systematic Logic in the framework of DHNN, the analysis involved conducting simulations using 69 distinct logical combinations (number of clauses) which corresponds to a total of 207 neurons. However, to facilitate comparison with prior research on systematic and non-systematic logic, the number of neurons (NN) will be within a defined range. Table 1 summarizes all the essential parameters for the configuration based on ES as the learning algorithm.

Table 1. The parameters utilized in ℓ_{NR3SAT} .

Setting	Parameter Value
Order of clauses	Third order clauses
Number of neurons, (NN)	$9 \leq NN \leq 207$
Neuron combination, (C)	100 [41]
Number of trials, (NT)	100 [42]
Number of learning, (NH)	100 [42]
Tolerance value, (Tol)	0.001 [38]
Activation function	Hyperbolic tangent (HTAF) [8]
Learning algorithm	Exhaustive Search [15]

Algorithm 2. Pseudo-code of HBHA

```

1  Begin
2      {Retrieval Phase} Generate  $\ell_{NR3SAT}$  initial star population  $P[Star_i(S_i^f)]$ ;
3  do
4      while  $S_i^f \in [1, NN]$  and  $Star_i \in [1, NT]$  do
5          Calculate fitness value based and determine  $Star_i^{BH}(S_i^f)$ ;
6      for  $\beta = [0, 1]$  do
7           $n[Star_i^R(S_i^f)] = \beta \cdot P[Star_i(S_i^f)]$ ;
8          Update new population of star,  $Star_i^N(S_i^f)$ ;
9      end for
10     for  $Star_i^N(S_i^f)$  do
11          $Star_i(S_i^f(t+1)) = Star_i(S_i^f) + \theta[Star_i^{BH}(S_i^f) - Star_i(S_i^f)]$ ;
12     end for
13     for  $Star_i(S_i^f(t+1))$  do
14         Convert the star position into bipolar form;
15         if  $F[Star_i(S_i^f(t+1))] > F[Star_i^{BH}(S_i^f)]$  then
16             Assign  $Star_i^{BH}(S_i^f)$  as new star;
17         else
18             Remain  $Star_i^{BH}(S_i^f)$  as unchanged;
19         end if
20     end for
21     for  $S_i^f(t+1)$  do
22         Calculate  $R$  and  $D$ ;
23         if  $D < R$  then
24             Generate new star randomly;
25         else
26             The star remains unchanged;
27         end if
28     end for
29 end while
30 End

```

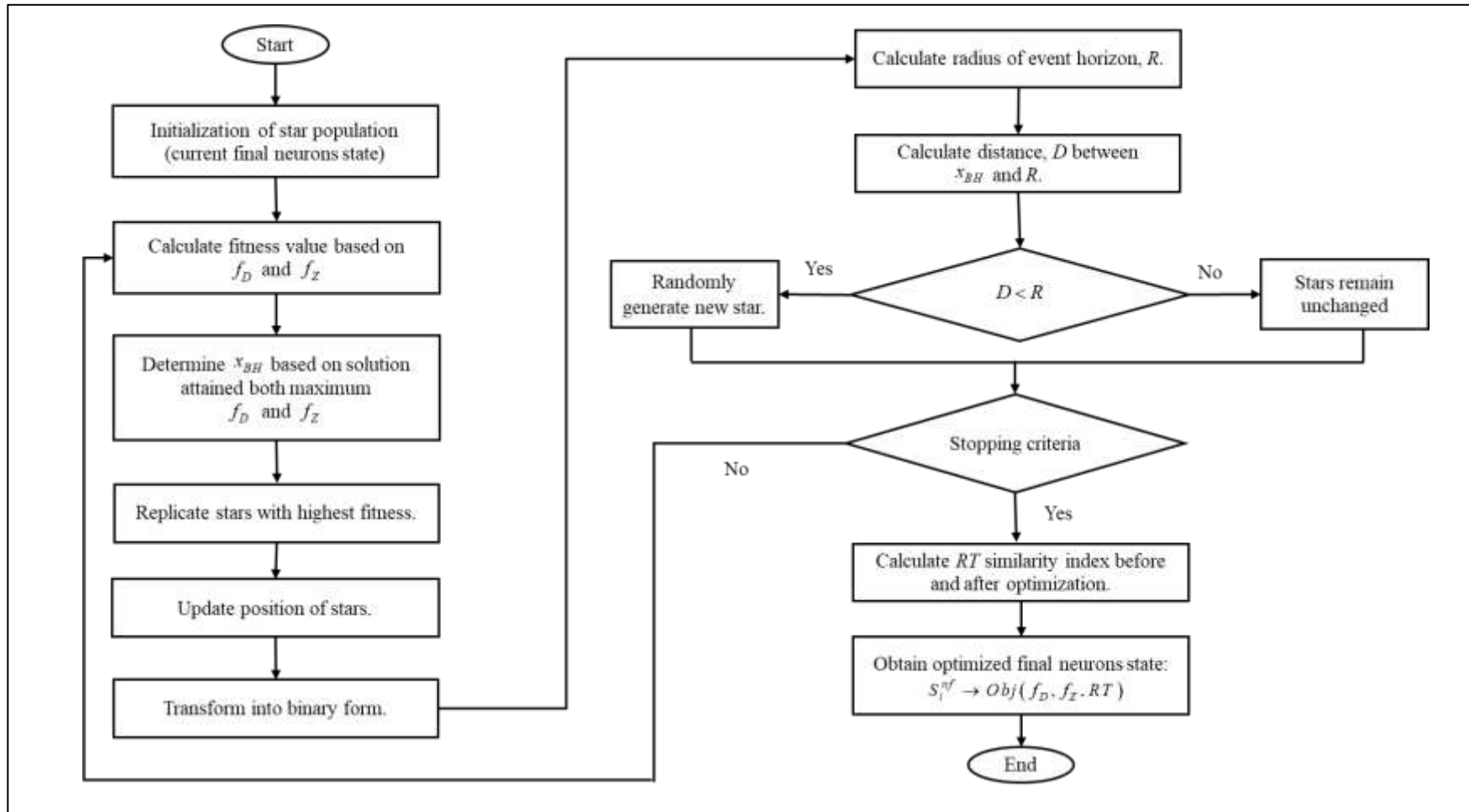


Figure 2. Flowchart of proposed Hybrid Black Hole Algorithm.

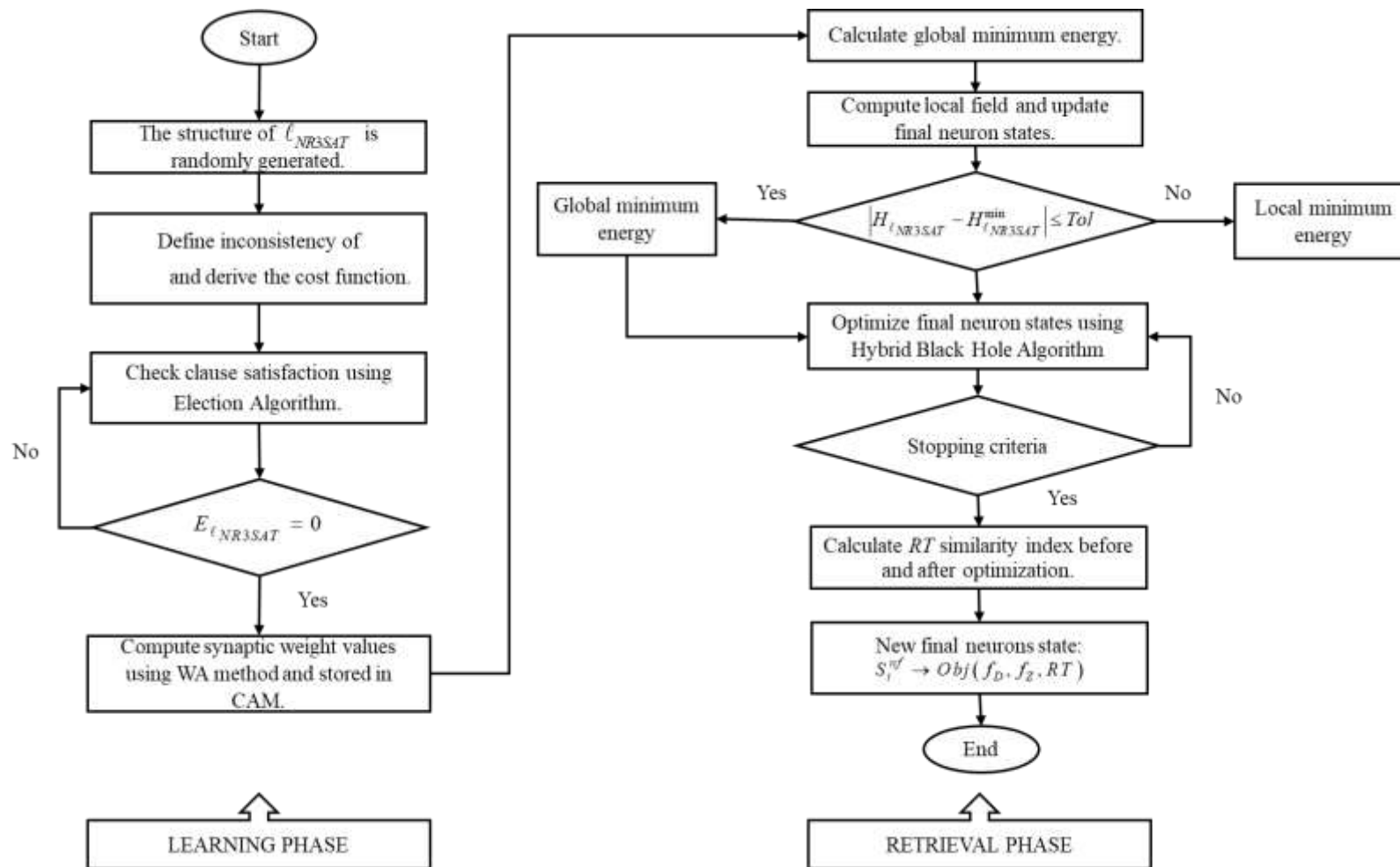


Figure 3. Flowchart of overall process in optimizing learning and retrieval capabilities of DHNN.

7.3.2. Parameter setup in the retrieval phase

To evaluate the effectiveness of HBHA to optimize the final neuron states retrieved by NR3SAT model, performance analysis will be conducted based on specific parameter setup. The parameters involved in the proposed model for simulated dataset are listed in Table 2. Additionally, detailed description of the parameters used in HBHA is presented in Table 3. Furthermore, Table 4 to Table 11 provided an overview of the parameters utilized in 10 baseline algorithms to assess their effectiveness compared to HBHA.

Table 2. The parameters utilized in NR3SAT.

Setting	Parameter Value
Minimum number of neurons, (NN)	9
Maximum number of neurons, (NN)	108
Neuron combination, (C)	100 [41]
Number of trials, (NT)	100 [42]
Number of learning, (NH)	100 [42]
Synaptic weight method	WA method [11]
Tolerance value, Tol	0.001 [38]
Initialization of neuron states	Random [35]
Activation function	HTAF [8]
Learning algorithm	Election algorithm [34]

Table 3. Parameter settings of HBHA.

Setting	Parameter Value
Operator	Clone, Updating rule, Selection
Cloning rate	0.7
Updating rule	Based on highest fitness
Mutation	Random
Selection	NT
Number of generations	100 [16]
Transfer function	HTAF [8]

Table 4. Parameter settings of GA.

Setting	Parameter Value
Operator	Selection, Crossover and Mutation
Number of generations	100 [16]
Selection rate	<i>NT</i>
Crossover rate	1 [16]
Mutation rate	1 [16]

Table 5. Parameter settings of CSA.

Setting	Parameter Value
Operator	Cloning, Somatic Hypermutation and Selection
Number of generations	100 [16]
Clone rate	0.5
Somatic Hypermutation rate	0.01
Selection rate	<i>NT</i>

Table 6. Parameter settings of DEA.

Setting	Parameter Value
Operator	Mutation, Crossover, and Selection
Number of DEA generations	100 [16]
Mutation rate	0.5
Crossover rate	0.5

Table 7. Parameter settings of ABCJ [49], ABCK [19] and ABCS [35].

Setting	Parameter Value
	ABCJ: XOR(\otimes), AND(\wedge) and OR(\vee)
Bitwise Operator	ABCK: OR (\vee), XOR(\otimes) and AND(\wedge)
	ABCS: NAND($\overline{\wedge}$)
Number of generations	10
Number of Employed bees	5
Number of Onlooker bees	5
Number of Scout bees	5

Table 8. Parameter settings of EA.

Setting	Parameter Value
Operator	Positive advertisements, Negative advertisements, and Coalition
Number of generations	100 [16]
Number of candidates	120 [33]
Number of parties	4 [34]
Positive and negative advertisement rate	0.5 [43]

Table 9. Parameter settings of SCA.

Setting	Parameter Value
Operator	Sine Updating rule or Cosine Updating rule
Updating rule	Based on highest fitness
Range r_2	$[0, 2\pi]$
Range r_3	$[0, 2]$
Range r_4	$[0, 1]$ [44]
Number of generations	100 [16]
Transfer function	Sigmoid function [44]

Table 10. Parameter settings of WOA.

Setting	Parameter Value
Operator	Shrinking encircling updating rule and Spiral updating rule
Shrinking encircling updating rule	Based on Random /Best fitness
Spiral updating rule	Based on best fitness
Range l	$[-1, 1]$ [45]
Number of generations	100 [16]
Transfer function	Sigmoid function [46]

Table 11. Parameter settings of BHA.

Setting	Parameter Value
Operator	Updating Rule and Mutation
Updating rule	Based on highest fitness
Mutation	Random
Number of generations	100 [16]
Transfer function	Hyperbolic tangent function [26]

7.4. Performance metrics

The evaluation of the DHNN will concentrate on assessing the effectiveness of the proposed logical rule NR3SAT in comparison to other logical structures. In contrast, evaluating the performance of HBHA will primarily consider the ability of the proposed HBHA in achieving a multi-objective functions, which is generating varied and global solutions with low similarity index. Consequently, we categorize the performance metrics into two parts:

- (a) The evaluation of the efficiency of NR3SAT in DHNN will involve analyzing various performance metrics such as learning error, testing error, global minimum ratio, total neuron variation, and Jaccard similarity index.
- (b) The performance metrics used to assess the effectiveness of HBHA in optimizing the retrieved final neuron states will focus on achieving an optimal solution string. This will be evaluated in terms of diversified solutions, global minima solutions, string that achieve both diversified and global minima solutions, and similarity index analysis.

7.4.1. Performance metrics in learning and retrieval phase

Performance metrics assume a critical role in evaluating the effectiveness and quality of the network's performance, particularly in relation to the proposed model. In this context, these metrics will be applied to assess the performance of the proposed ℓ_{NR3SAT} model, comparing it to the existing models. This extensive evaluation encompasses two crucial phases: The learning phase and the retrieval phase of DHNN. In the learning phase of DHNN, error analysis is conducted using Mean Absolute Percentage Error (MAPE). Inspired by Sathasivam et al. [34], we employ this metric to evaluate the efficiency and learning capabilities of ES in finding a satisfied interpretation of NR3SAT. Eq (39) provide the formulations for MAPE. Based on the Equations, f_{NC} represents the maximum fitness of satisfied clauses associated to NR3SAT and f_i represent current fitness obtained. Ideally, all these metrics were employed to assess the efficiency of ES in finding maximum satisfaction of the proposed logical rule.

$$MAPE_{learn} = \frac{100}{NH} \sum_{i=1}^{NH} \frac{|f_{NC} - f_i|}{|f_i|}. \quad (39)$$

In the retrieval phase, the performance metric will be employed to assess the quality of the retrieved final neuron states using Eq (40) [14]. Based on Eq (40), i referred to the number of iterations throughout the process. According to Roslan et al. [42], this testing error can be used to evaluate the accuracy of the retrieved final neuron states. In this case, zero testing errors indicate global solutions.

$$MAPE_{test} = \frac{100}{\alpha} \sum_{i=1}^{\alpha} |N_{TS} - N_{GS}|. \quad (40)$$

Additionally, drawing inspiration from the work of Kasihmuddin et al. [21] to ensure the stability of the retrieved final neuron states and the minimization of energy achieved, the performance will be assessed using Global Minima Ratio, R_G as outlined in Eq (41). Worth mentioning that the optimal value of R_G is when $R_G = 1$, which indicates that the retrieved states achieved global minima energy

at the number of neurons.

$$R_G = \frac{1}{\alpha} \sum_{i=1}^{\alpha} N_{GS}. \tag{41}$$

To comprehensively assess the quality of the final neuron states retrieved, an analysis of similarity index values and cumulative neuron variation will be conducted. Ideally, the similarity index analysis was used to measure the diversification of the states by comparing the final neuron states retrieved by the network with the benchmark states. In this context, the benchmark neuron state is defined as follows:

$$S_i^{\max} = \begin{cases} 1, & \text{if } q_i \\ -1, & \text{if } \neg q_i \end{cases}. \tag{42}$$

Referring to Eq (42), q_i and $\neg q_i$ defined as positive and negative literal respectively. Drawing inspiration from the work of Alway et al. [15], the Jaccard similarity index as outlined in Eq (43) will be utilized to measure the similarity of the positive value in comparison to the benchmark states, as detailed in Eq (42). Table 12 presented all the parameters utilized in both learning and retrieval phase of DHNN and the list of parameters incorporated within the similarity index is presented in Table 13.

$$Jaccard = \frac{a}{a+b+c}. \tag{43}$$

Table 12. List of parameters utilized in the experiment setup.

Parameter	Definition
NH	Number of learning
NC	Number of clauses
f_{NC}	Maximum fitness obtained based on NC
f_i	Current fitness obtained
N_{TS}	Number of total solutions
N_{GS}	Number of Global minima solution
α	$NT \cdot C$
$H_{\ell_{NR3SAT}}^{\min}$	Minimum energy of ℓ_{NR3SAT}
$H_{\ell_{NR3SAT}}$	Final energy of ℓ_{NR3SAT}

Table 13. List of parameters incorporated within the similarity index.

Similarity index coefficient	S_i^{\max}	S_i^f
a	1	1
b	1	-1
c	-1	1
d	-1	-1

Additionally, cumulative neuron variation often referred to as TV , is employed to assess the model's ability to explore diverse solutions across various regions within the search space. The TV can be formulated as in Eqs (44) and (45) such that v represents the total string of retrieved final neuron states while K_i function as a scoring mechanism to determine the count of distinct neuron states between the benchmark and retrieved final neuron states.

$$TV = \sum_{i=0}^v K_i, \quad (44)$$

such that

$$K_i \begin{cases} 1, & S_i^f \neq S_i^{\max} \\ 0, & S_i^f = S_i^{\max} \end{cases}. \quad (45)$$

7.4.2. Performance metrics in optimizing the retrieval phase of DHNN

To evaluate the effectiveness of the proposed model, we will analyze how well the model optimizes the retrieval phase of DHNN to meet the objective functions. Our proposed HBHA aims to produce diverse and global solutions with low similarity indices, thereby enhancing overall performance and effectiveness of the DHNN framework. A key focus on implementing EA in the learning phase as EA contributes to achieving an optimal learning phase, which consequently leads to an optimal retrieval phase. By incorporating HBHA into the retrieval phase, we anticipate optimizing the final neuron states retrieved by the network. Therefore, various performance metrics will be introduced to assess the quality of the optimized final neuron states. The evaluation focuses on each string of final neuron states emphasizing the diversification of negativity, the attainment of global minima solutions and solutions with low similarity index. Regarding diversification, we focused on obtaining diversified solutions based on clauses. For instance, when the degree of negativity is set to $m\%$, it signifies that the optimized final neuron states must contain at least $m\%$ of the total number of clauses, each having at least one negative literal. In this context, the number of diversified solutions, N_{DS} can be defined using Eq (46) as follows with diversification rate of $m\%$.

$$N_{DS} = \sum_{i=1}^{\alpha} n(NS_{i_{DS}}^f). \quad (46)$$

In Eq (46), $n(NS_{i_{DS}}^f)$ represents the number of solution strings that achieved the maximum predefined diversity rate while α denotes the total number of simulation sets to be executed, $\alpha = NT \times C$. Note that α can also be referred to as the total number of solutions. The optimized final neuron will be evaluated based on each string according to Eq (47).

$$\begin{aligned}
NS_{1DS}^f &= S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad \dots \quad S_{NN}^f \\
NS_{2DS}^f &= S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad \dots \quad S_{NN}^f \\
NS_{3DS}^f &= S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad \dots \quad S_{NN}^f \\
&\vdots \\
NS_{NTDS}^f &= S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad \dots \quad S_{NN}^f
\end{aligned} \tag{47}$$

In the perspective of evaluating the solution string that attain a global solution, the number of solution strings that achieve global solutions, N_{GS} can be determined using Eq (48). The solution string NS_{iGS}^f that achieves a global solution is defined in Eq (49), indicating the optimized final neuron states that successfully interprets the logical rule NR3SAT.

$$N_{GS} = \sum_{i=1}^{\alpha} n(NS_{iGS}^f), \tag{48}$$

such that

$$\begin{aligned}
NS_{1GS}^f &= S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad \dots \quad S_{NN}^f \\
NS_{2GS}^f &= S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad \dots \quad S_{NN}^f \\
NS_{3GS}^f &= S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad \dots \quad S_{NN}^f \\
&\vdots \\
NS_{NTGS}^f &= S_1^f \quad S_2^f \quad S_3^f \quad S_4^f \quad \dots \quad S_{NN}^f
\end{aligned} \tag{49}$$

By evaluating the solution string that achieved N_{DS} and N_{GS} , we can evaluate the solution string that achieved both maximum $NS_{iDS}^f \cap NS_{iGS}^f$, which is computed as in Eq (50).

$$N_{DG} = \sum_{i=1}^{\alpha} n(NS_{iDS}^f \cap NS_{iGS}^f). \tag{50}$$

Thus, N_{DG} represents the number of solution strings that have met both objectives. Additionally, the global minimum ratio, N_{GR} for the total number of global solutions obtained by N_{GS} can be evaluated using Eq (51).

$$N_{GR} = \frac{1}{NT \cdot C} \sum_{i=1}^{\alpha} N_{GS}. \tag{51}$$

In the perspective of attaining low similarity index, taking inspiration by the work of Zamri et al., [16], this paper focusses on Rogers and Tanimoto similarity index, RT to assess the quality of the optimize final neuron states. This is due to the ability of this metric to measure the similarity of the negative cases for each string of solutions between before and after optimization took place. In this context, the benchmark solutions are referring to the retrieved final neuron states before optimization

takes place which can be define as in the Eq (52) such that q_i and $\neg q_i$ referring to the positive and negative literals of NR3SAT. Moreover, the optimize final neurons state can be referred to as S_i^{fa} .

$$S_i^f = \begin{cases} 1, & \text{for } q_i \\ -1, & \text{for } \neg q_i \end{cases} \quad (52)$$

Thus, the general comparison between the benchmark states and optimize final neuron states is presented as Eq (53) such that τ is referring to the similarity index coefficients defined as in Table 14. The formulation of RT is given as in Eq (54). It is crucial to emphasize that when evaluating the quality of the optimized final neuron states, the effectiveness of the proposed algorithm is measured by its capability to yield low values of RT . Lower values of RT in each comparison between string indicate that the optimized final neuron states differ from the benchmark states. Table 15 presented the list of symbols utilized in the experimental setup.

$$\tau = (S_i^f, S_i^{fa}), \quad (53)$$

$$RT = \frac{a+d}{a+d+2(b+c)}. \quad (54)$$

Table 14. List of parameters incorporated within the similarity index.

Similarity index coefficient, τ	S_i^f	S_i^{fa}
a	1	1
b	1	-1
c	-1	1
d	-1	-1

Table 15. List of symbols utilized in the experimental setup.

Parameter	Parameter's Name
NS_{iDS}^f	Diversified solution string
N_{DS}	Number of diversified solutions
NS_{iGS}^f	Global minimum solution string
N_{GS}	Number of global solutions
N_{GR}	Ratio of global solutions
N_{DG}	Number of diversified and global solutions
NT	Number of trials
N_{TS}	Number of total solutions
S_i^f	Benchmark final neuron state before the optimization
S_i^{fa}	Optimized final neuron state after the optimization
RT	Rogers and Tanimoto similarity index

7.5. Baseline logical rules

In the learning phase of DHNN, NR3SAT will be compared with several existing baseline models. Each of these baseline models is described as follows:

- (a) 3SAT [17]: 3SAT is a SAT formulation within the systematic SAT class, characterized by a restricted number of literals per clause. In this context, each clause consists of 3 literals. As described by Mansor et al. [17], 3SAT exhibits a higher probability of obtaining satisfied interpretation compared to other systematic SAT variants. Consequently, 3SAT has the potential to achieve an optimal learning phase, leading to a greater production of global minima solutions.
- (b) RAN k SAT [14]: RAN k SAT is a SAT formulation belonging to the non-systematic SAT class, which encompasses first order, second order and third order classes. Each class in RAN k SAT is designed to have an equal representation. The performance for all logical combinations was evaluated and results showed that RAN k SAT with a logical combination of $(k = 2, 3)$ yield the most promising results compared to other combinations. Notably, the trend of RAN k SAT for $(k = 2, 3)$ is more consistent in obtaining lower learning and testing errors. The finding emphasizes that higher order logical structure contributes to higher probability of achieving global solutions.
- (c) MAJ2SAT [15]: MAJ2SAT is another non-systematic SAT class which includes both second order and third order classes. In the study, Alway et al. [15] enhance the structure of the logical rule by introducing a bias of 2SAT clauses compared to 3SAT clauses. Simulated results signified MAJ2SAT successfully embedded in DHNN since the model can generate optimal final neuron states that achieve global minima energy. Notably, MAJ2SAT demonstrated its proficiency in retrieving accurate synaptic weights that play a crucial role in achieving global minima solutions.
- (d) GRAN k SAT [41]: Gao et al. [41] contribute to explore innovative logical structures by proposing a novel and flexible higher-order logic called G – Type Random k Satisfiability (GRAN3SAT) in DHNN. This newly introduced logic stands out as both systematic and non-systematic structure. GRAN3SAT consists of new random feature of first, second and third order clauses and the performance of GRAN3SAT was evaluated using order clauses, different proportions of positive and negative literals and different number of learning trials. Based on the findings, when considering GRAN3SAT with the highest proportion of 3SAT clauses, the learning process exhibited the lowest Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). This observation supports the idea that implementing third-order clauses into GRAN3SAT decreases the learning errors.

7.6. Baseline algorithms

To evaluate the effectiveness of the proposed algorithm in optimizing the retrieved final neuron states, HBHA will be compared against several baseline algorithms. This includes Genetic Algorithm (GA) [12], Clonal Selection Algorithm (CSA) [47], Differential Evolution Algorithm (DEA) [48], Artificial Bee Colony Algorithm (ABCJ) [49], Artificial Bee Colony Algorithm (ABCK) [19], Artificial Bee Colony Algorithm (ABCS) [35], Election Algorithm (EA) [34], Sine Cosine Algorithm (SCA) [44], Whale Optimization Algorithm (WOA) [50], and Black Hole Algorithm (BHA) [26]. All these baseline algorithms can be categorized into four major categories including Evolutionary-based, Swarm intelligence-based, sociopolitical-based, and Natural phenomena-based, which can be presented in Figure 4. Each of these baseline algorithms is described as follows:

- (a) GA [12]: The implementation of the GA in DHNN aims to enhance the learning phase and improve the accuracy of finding correct neuron states that can increase the probability of getting satisfied interpretation of 2SAT logical rule. By combining exploration and exploitation processes of the GA, GA helps in searching for an optimal neuron state with high fitness value. This incorporation reduces hamming distance and achieves a higher global minima ratio as compared to traditional exhaustive search methods. Thus, the performance of 2SAT in DHNN has been improved.
- (b) CSA [47]: Incorporating the CSA in this research aims to enhance the learning phase of DHNN. CSA, inspired by the natural immune system improves the learning mechanism of the DHNN through selection, cloning and somatic hypermutation. By implementing CSA into the model, it aims to optimize employee resource application approval processes by efficiently searching for an optimal induced logic.
- (c) DEA [48]: DEA involves three operators: mutation, crossover and selection. Mutation explores a wide range of potential solutions in the search space for global search. Crossover refines solutions locally near the target vector to exploit optimal results. The balance between exploration and exploitation is crucial for high-quality solutions in optimization problems. In this research, a novel binary differential evolution algorithm based on Taper-shaped transfer functions is proposed to address binary optimization problems that cannot be solved directly by traditional DE due to its reliance on real number operations for mutations. By employing transfer functions such as S-shaped, U-shaped or V-shaped curves, the binary form of DE can be attained efficiently.
- (d) ABCJ [49]: Jia et al. [49] introduced a new binary optimization algorithm that is based on the original ABC algorithm. The main process of this algorithm consists of three phases: Employed bees' phase, onlooker bees' phase, and scout bees' phase. Movement in these phases is done using bitwise operators. The proposed bitwise ABC algorithm has been compared to other binary algorithms using various benchmark functions. Additionally, this study also compared the proposed ABC with other variations of both the ABC and GA algorithms. Experimental results demonstrated that the proposed ABC achieves better accuracy in optimization as well as faster convergence speed when compared to other algorithms used in this research.
- (e) ABCK [19]: Kasihmuddin et al. [19] implemented a novel bitwise operator in the ABC algorithm to solve the 2SAT. The purpose of this implementation was to improve the learning phase of DHNN. By incorporating ABC as a searching technique with DHNN, they aimed to find a satisfied interpretation of 2SAT logic within an acceptable timeframe. Their goal was to compare the performance of solutions produced by DHNN with ABC, HNN2SAT-ABC against traditional learning algorithms, HNN2SAT-ES. The results show that HNN2SAT-ABC outperforms HNN2SAT-ES in terms of global minima ratio, hamming distance and CPU time, suggesting that using ABC is a better alternative method for finding satisfied interpretation of 2SAT.
- (f) ABCS [35]: Sidik et al. [35] provided insights into the effectiveness of different bitwise logic gate operators and their compatibility with the ABC algorithm. They focused on controlling the distribution of negative literals by implementing the ABC algorithm in the logic phase. The study compares the performance of different logic gate operators in producing desired logical structures. The results showed that implementing the NAND operator with the ABC algorithm outperforms other operators in terms of generating negative literals. Choosing the correct logic gate operator enhances fitness and improves exploration and exploitation processes for finding

optimal solutions. Different operators have varying levels of success depending on specific ratios of negative literals.

- (g) EA [34]: Sathasivam et al. [34] investigated the use of EA to enhance the learning phase of a DHNN. The goal is to optimize the learning phase of DHNN by ensuring that the cost function of RAN2SAT converges to zero, indicating a satisfied interpretation of the logical rules. Due to low probability of finding satisfied interpretation for first order clauses, complexity of logic satisfiability increases significantly in DHNN. However, by employing EA, neuron states can be effectively flipped. The results demonstrated the accuracy and effectiveness of EA as a learning algorithm in DHNN for RAN2SAT with varying numbers of neurons compared to other algorithms such as Exhaustive Search and Genetic Algorithm.
- (h) SCA [44]: This research proposed two binary SCA, S-shape and V-shape, for medical feature selection. These algorithms convert continuous values into binary using transfer functions. The goal is to select effective feature subsets from medical data, improving classification accuracy while reducing the subset length. Experimental results showed that both binary SCAs outperform other algorithms in classifying medical datasets.
- (i) WOA [50]: Hussien et al. [50] proposed two binary versions of the WOA called bWOA-S and bWOA-V for feature selection problems. These versions use specific transfer functions, either S-shaped or V-shaped, to convert continuous search space solutions into binary solutions. The objective of implementing WOA is to deal with feature selection problems by simplifying high-dimensional datasets and improving classification accuracy through selecting relevant features while removing irrelevant and redundant data. Inspired by humpback whales' bubble-net feeding hunting technique, the classical WOA sets the current best candidate solution close to either the optimum or target prey while other whales update their position towards this best solution.
- (j) BHA [26]: Pashaei & Aydin [26] used binary version of BHA and decision tree algorithm to solve the feature selection and classification problem in biological data. BHA is an optimization technique inspired by black holes in outer space, where stars simulate their behavior. The best star with the best fitness value acts as the black hole, pulling other stars towards it like gravitational force in real space. The Binary Black Hole Algorithm (BBHA) is an extended version of the BHA that solves discrete problems, such as feature selection. BBHA applied binary structure to indicate which features belong in the final set. BBHA was tested on biological datasets and compared with other methods, showing superior performance in all metrics.

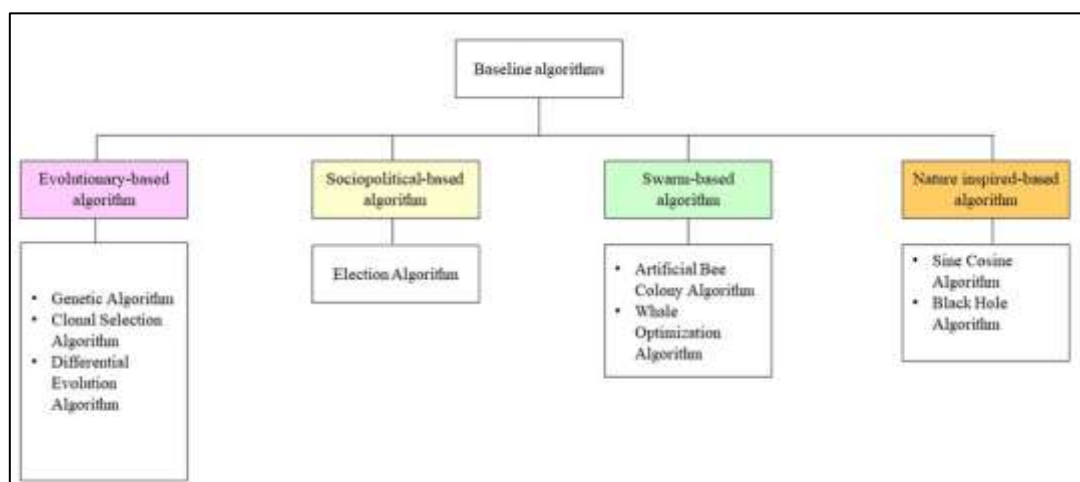


Figure 4. The categories of baseline algorithms.

8. Results and discussion

In this section, we focus on providing a comprehensive analysis of the performance of the proposed DHNN model, considering the selected performance metrics and all the parameters configured in both the learning and retrieval phases of DHNN. The analysis focused on the capabilities and efficiency of the proposed model in comparison to well-established baseline models. To elaborate further, the results and discussions are divided into two main parts. The first part will be focused on evaluating the performance and stability of DHNN in reducing the logical inconsistencies of NR3SAT compared to the existing models. This section focused on the results and discussions pertaining to the use of ES as the learning approach. The subsequent section will shift the focus to the efficiency of the proposed HBHA in optimizing the retrieval phase of DHNN when compared to existing baseline algorithms.

8.1. Performance analysis of ℓ_{NR3SAT} in the learning phase

The performance analysis of the proposed NR3SAT logical rule in the framework of DHNN will be discussed based on several metrics. In the first subsection, we focus on the learning errors analysis followed by testing errors analysis and similarity index analysis. In the last subsection, we discuss non-parametric statistical analysis. Moreover, in the second part of this section, we explain the results and findings based on optimizing the retrieval capabilities of DHNN.

8.1.1. Learning error analysis

To provide insight on the performance of DHNN by incorporating NR3SAT logical rule and ES as the learning approach in finding satisfied interpretation, the analysis was carried out using four essential metrics: Learning error, testing error, energy analysis, and similarity index. Table 16 presented the learning errors obtained through $MAPE_{learn}$ which also can be visualized as in Figure 5. In general, the proposed NR3SAT can achieve the smallest learning errors compared to other existing models as observed in Table 16. The smallest errors produced by NR3SAT highlight the superiority of the proposed model in attaining a satisfied interpretation, leading to the minimization of the cost function. Besides this, when ES is used as the learning algorithm and the number of neurons increases, the learning error also tends to increase, as shown in all illustrated figure. These higher values of $MAPE_{learn}$ indicate a high percentage of unsatisfied clauses. When ES fails to find the satisfied clauses, this causes the network to retrieve random synaptic weights to be stored in CAM. This, in turn leads to a non-optimal learning phase. This phenomenon occurs because as the number of neurons increases, the probability of finding a satisfied interpretation becomes very low. This is because it becomes more challenging for the neurons to achieve zero-cost function. One of the main reasons contributing to this trend is the nature of ES which relies on a trial-and-error approach to find satisfied interpretation. As the problem complexity increases with a greater number of neurons, ES may struggle to find a satisfied interpretation due to the increase search space. However, NR3SAT consistently maintains the lowest errors when compared to existing logics. The logic that comes closest to achieving the lowest error is 3SAT. Based on the observation between NR3SAT and 3SAT, the restriction feature in NR3SAT with at least one negative literal in each clauses promotes a higher probability of obtaining a satisfied interpretation. Additionally, when comparing the errors obtained using NR3SAT and 3SAT, it becomes evident that the inclusion of all positive literals in 3SAT clauses can disrupt the process of finding satisfied interpretation leading to higher $MAPE_{learn}$ errors.

In comparison with non-systematic logics, it is worth noting that the errors obtained when using non-systematic logics, which involved first order clauses $\text{RAN}k\text{SAT}(k=1,3)$, $\text{RAN}k\text{SAT}(k=1,2,3)$, $\text{GRAN}k\text{SAT}(k=1,2,3)$, and $\text{GRAN}k\text{SAT}(k=1,3)$, were consistently higher than those obtained by systematic logics. Additionally, an observation from Table 16 revealed that as the number of neurons surpasses 90, these non-systematic logics demonstrate suboptimal performance. Notably, no errors are generated beyond a certain threshold of neurons, indicating the inability of non-systematic logics to identify satisfied interpretation. This finding highlighted the instability of these logical rules when compared to the systematic logics. This demonstrated that increasing number of first-order clauses significantly influences the process of finding a satisfied interpretation of the logical rule. This is because there is only one possibility to achieve a satisfied interpretation, which is either 1 or -1 [16]. This contrasts with second and third order clauses which have more combinations to make it satisfied. This observation is further supported by the results obtained using $\text{GRAN}k\text{SAT}$. As evident in the $\text{MAPE}_{\text{learn}}$ metric, the learning errors produced by $\text{GRAN}k\text{SAT}$ exhibit inconsistency compared to other logics. This inconsistency can be due to the flexibility of $\text{GRAN}k\text{SAT}$, which can function as both systematic 3SAT and non-systematic SAT logic. It is noticeable that at certain neuron numbers, such as when NN falls within the range of $45 < NN < 53$ and $81 < NN < 89$, the $\text{MAPE}_{\text{learn}}$ errors experienced drastic decrease. This decrease in errors align with $\text{GRAN}k\text{SAT}$ transitioning into a 3SAT logic. The lowest errors observed when $\text{GRAN}k\text{SAT}$ behaves as a 3SAT logic indicate that at that point, $\text{GRAN}k\text{SAT}$ easily finds satisfied interpretation leading to lower errors. This observation indicated that higher-order logic increases the probability of achieving a satisfied interpretation. Therefore, it can be summarized that DHNN by incorporating NR3SAT logical rule exhibited the best learning capability across metrics in terms of its learning capability.

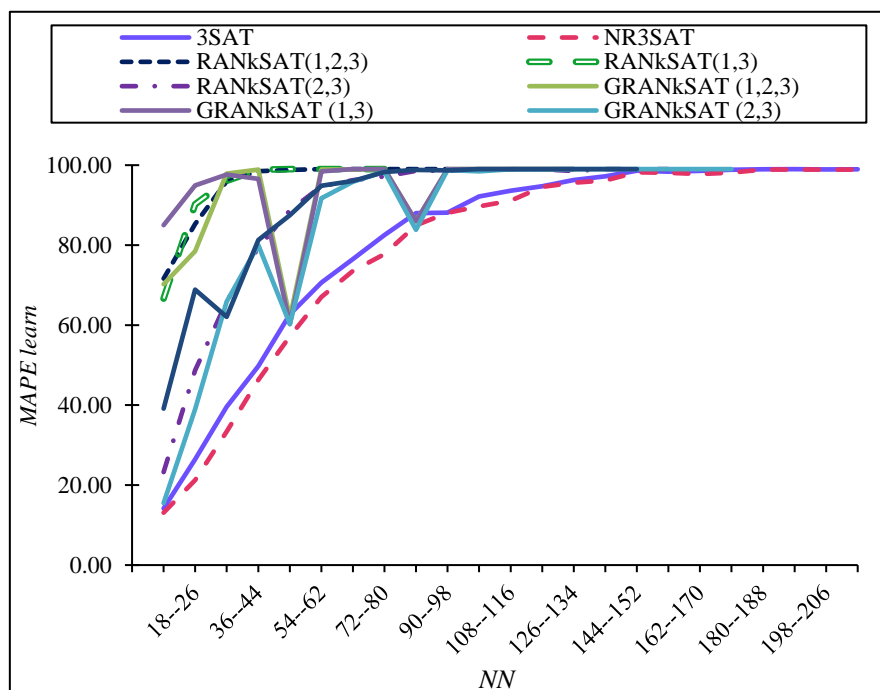


Figure 5. $\text{MAPE}_{\text{learn}}$ of NR3SAT with existing logic models.

8.1.2. Testing errors analysis

During the retrieval phase of DHNN, conducting error analysis is crucial to assess the quality of the retrieved final neuron states. This is particularly important because as pointed out by Zamri et al. [16], DHNN often generates repetitive final neuron states instead of producing new ones. To tackle this challenge, the assessment of the retrieved final neuron states was conducted from three critical perspectives: Testing error evaluation, the generation of global minima solutions, and the quality of the retrieved solution in comparison to benchmark states. Once NR3SAT has successfully found a satisfied interpretation resulting in $E_{\ell_{NR3SAT}} = 0$, the optimal synaptic weights are generated. Therefore, the conducted error analyses aim to examine how NR3SAT behaves based on the synaptic weights acquired during the learning phase specifically in terms of generating local or global minima solutions when compared to other existing baseline models. Graphical results of $MAPE_{test}$ can be observed in Figure 6 with tabulated results as shown in Table 17. Indeed, these errors exhibit a strong correlation with the errors encountered during the learning phase. In simpler terms, when learning error increases due to the challenges in finding satisfied interpretation, this subsequently leads to an increase in $MAPE_{test}$. Nonetheless, NR3SAT consistently managed to attain the lowest testing errors among all baseline models. This can be due to its high capability of finding satisfied interpretation in comparison to the other logics. Consequently, this underscores the ability of ℓ_{NR3SAT} to generate a higher number of global solutions compared to existing logic models.

Furthermore, a comparison between systematic and non-systematic logics revealed that systematic logics can achieve zero $MAPE_{test}$ when $NN < 63$. Within this range, both NR3SAT and 3SAT excel in finding satisfied interpretation leading to zero testing errors. On the other side, non-systematic logics with lower-order logic of 1SAT such as RAN k SAT ($k = 1, 2, 3$), RAN k SAT ($k = 1, 3$), GRAN k SAT ($k = 1, 2, 3$) and GRAN k SAT ($k = 1, 3$) tend to generate higher $MAPE_{test}$ compared to MAJ2SAT which does not include 1SAT. This can be due to the lower probability of obtaining satisfied interpretation in lower-order logics during the learning phase of DHNN as indicated by Karim et al. [14]. When dealing with higher-order logics like 2SAT or 3SAT, there are significantly more possible arrangements that can satisfy the clauses. In contrast, 1SAT can only be satisfied by values of 1 or -1. Consequently, when these logics fail to obtain satisfied interpretation, non-optimal synaptic weights are generated resulting in higher errors. From a systematic logic perspective, NR3SAT begins to outperform 3SAT when $NN \geq 72$. The $MAPE_{test}$ produced in this range are significantly lower when compared with 3SAT, despite both logics being of the same order. This superiority of NR3SAT can be due to its logical structure which incorporates more negative literals in a clause compared to existing 3SAT. The presence of these negativity biased clauses enhanced the process of obtaining satisfied interpretations, resulting in lower errors during the retrieval phase.

As previously discussed, the ability to achieve low error values during the retrieval phase indicates the model's effectiveness in discovering a greater number of global minima solutions, thereby ensuring the convergence property of DHNN. This can be quantified through the retrieved energy represented as the ratio of global minima solutions, R_G as depicted in Figure 7 and Table 18. Generally, NR3SAT and baseline models featuring higher-order logics such as 3SAT, RAN k SAT ($k = 2, 3$), GRAN k SAT ($k = 2, 3$), and MAJ2SAT ($k = 2, 3$), tend to attain their maximum global minima ratios when NN falls within the range of 9 to 18. This range signifies the successful implementation of the proposed SAT into DHNN. From another perspective, it can also be observed

that non-systematic models incorporating first order logics such as RANkSAT ($k = 1, 2, 3$) and RANkSAT ($k = 1, 3$) exhibit a notable decrease in their performance values as the number of neurons increases. This decline in performance can be due to the increased number of first-order clauses in these models. Essentially, the increased presence of first order clauses reduces the probability of these models achieving satisfied interpretations. This in turn complicates the process of minimizing logical inconsistencies. Consequently, the cost function minimization becomes unsuccessful, leading to the generation of non-optimal synaptic weights. In simpler terms, the inability to attain satisfied interpretation which leads to a zero-cost function results in incorrect synaptic weights and ultimately affecting the overall quality of the retrieved final neuron states. These findings are further supported by the performance of GRANkSAT ($k = 1, 2, 3$) and GRANkSAT ($k = 1, 3$) in comparison to GRANkSAT ($k = 2, 3$). GRANkSAT ($k = 2, 3$) significantly outperforms the other two models, as it can achieve high R_G approaching 1 when the number of neurons falls within the range of 9 to 62. However, as the number of neurons increases, the inclusion of 1SAT in GRANkSAT ($k = 1, 2, 3$) and GRANkSAT ($k = 1, 3$) disrupts the process of obtaining global solutions. Additionally, the flexibility of these models which can also be systematic 3SAT, demonstrated that when these models transition into systematic 3SAT (without first order logics), the global solution ratio increases dramatically. This results in fluctuations in the performance patterns of these models.

Furthermore, a close comparison between the global minima ratios produced by NR3SAT and 3SAT can be observed in Figure 8. Notably, NR3SAT consistently achieves a maximum ratio of 1 up to $NN = 80$, whereas in the case of 3SAT, the ratio of global minima solutions starts to decrease as NN exceeds 63. The significance of attaining global minima solutions is closely linked to the quality of the retrieved final neuron states which will be further explained in the context of the similarity index perspective. The question is, why is it so important to have global solutions? To address the importance of having global solutions, it is essential to understand that global solutions play a critical role in assessing the quality of the retrieved final neuron states. Without global solutions, the quality of the retrieved final neuron states remains uncertain. This connection between global solutions and the quality of final neuron states is particularly evident when examining the relationship between learning and testing errors. Thus, the quality of the retrieved final neuron states can be effectively evaluated using similarity index.

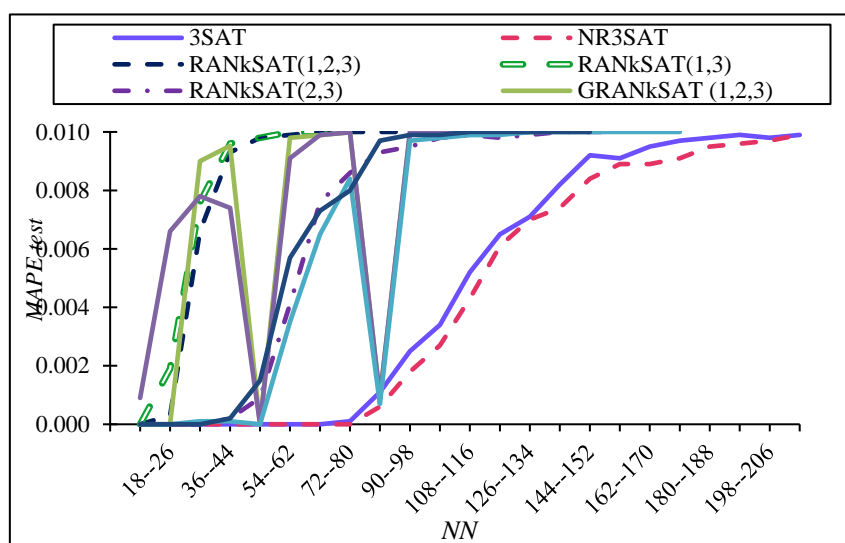


Figure 6. $MAPE_{test}$ of NR3SAT with existing logic models.

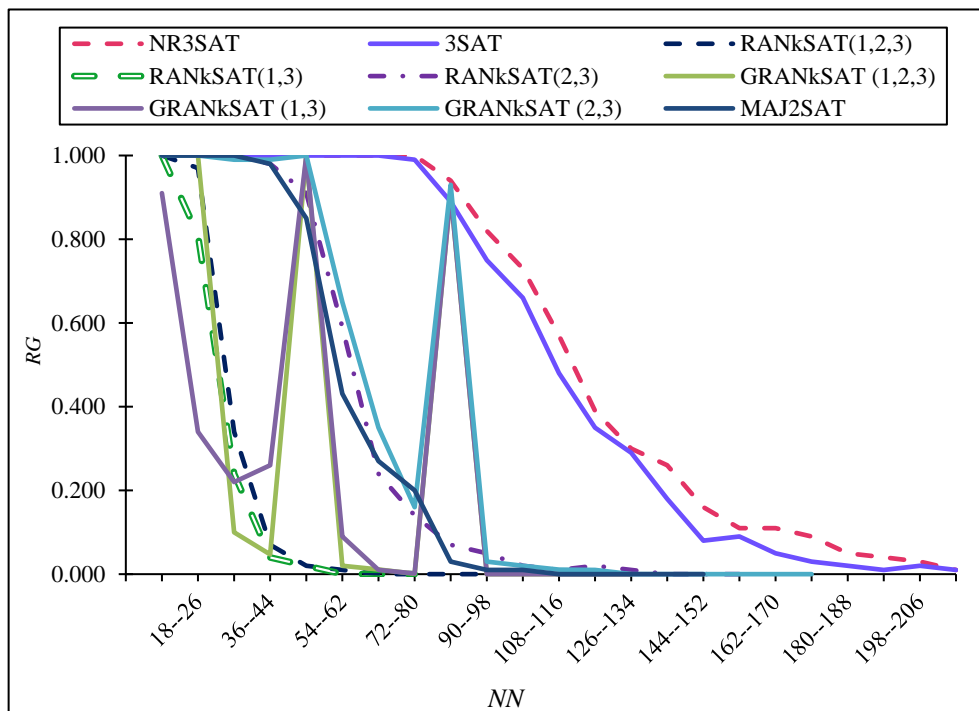


Figure 7. R_G of NR3SAT with existing logic models.

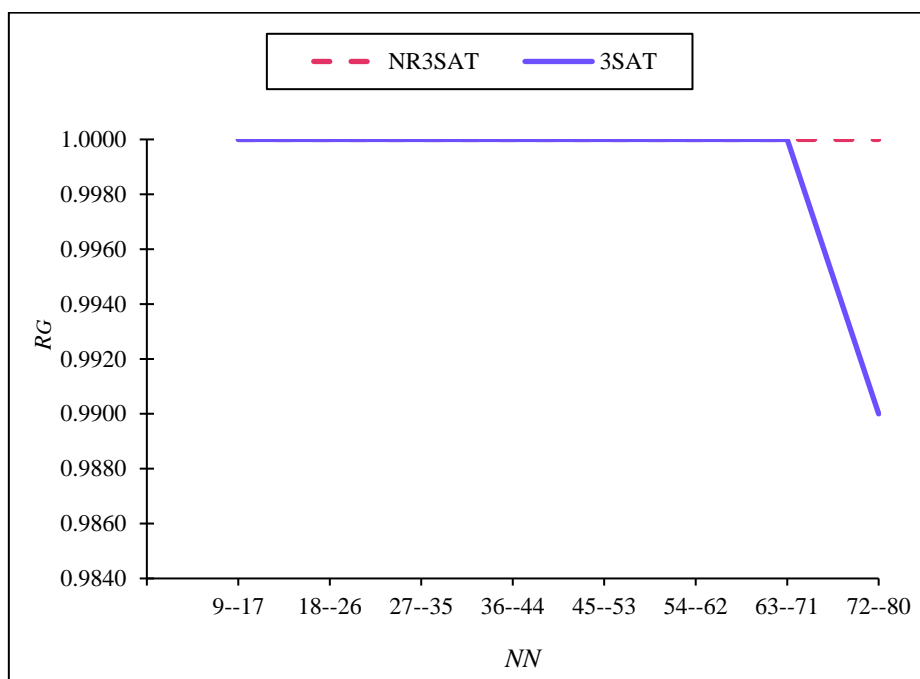


Figure 8. R_G of NR3SAT and 3SAT.

8.1.3. Similarity index analysis

To assess capability of NR3SAT in producing highly dissimilar states, its performance was

analyzed using two key metrics: the Jaccard value and Total Neuron Variation (TV). These metrics are utilized to compare NR3SAT with other baseline models, as illustrated in Figure 9 and Figure 10. Additionally, the tabulated data can also be presented in Table 19 and Table 20. In the context of Jaccard values, smaller values indicated high dissimilarity between the retrieved final neuron states and the benchmark states. Referring to Figure 9 and Table 19, a notable observation is that the majority of the baseline models achieved Jaccard values exceeding 0.550, while NR3SAT consistently obtained relatively low Jaccard values, falling below 0.5. These lower Jaccard values indicate that the final neuron states retrieved by NR3SAT exhibit a high degree of dissimilarity when compared to the benchmark states. This high dissimilarity signifies non-overfitting solutions. In contrast, overfitting occurs when all the retrieved final neuron states match the benchmark states precisely. It is crucial to emphasize that when addressing real-life datasets for classification problems, the presence of overfitting solutions can result in suboptimal logical rules representing the datasets. Consequently, this suboptimal logical rule may fail to precisely extract information and explain the strength of associations between variables. Furthermore, as mentioned by Guo et al. [8], the assessment of similarity index only for those solutions which achieved global solutions. When comparing systematic and non-systematic logics, the non-systematic logics failed to achieve Jaccard values for certain NN . This is because the generated solutions were trapped in local minima due to the non-optimal synaptic weights being used to update the local field. One of the main reasons why this contributes to non-optimal synaptic weights is because of the nature of ES. ES, which operates based on a trial-and-error approach, faces limitations in finding a satisfied interpretation as the number of neurons increases, impacting the minimization of the cost function [33].

Based on systematic logic, both 3SAT and NR3SAT generally obtain better Jaccard values compared to non-systematic logic. However, Table 19 reveals a significant difference in the values obtained, with NR3SAT outperforming 3SAT by consistently achieving lower Jaccard values. Additional feature of including at least one negative literal in NR3SAT has a substantial impact on the quality of the retrieved final neuron states. Consequently, this promotes more diversified final neuron states. Shifting the focus to the possible combination of 3SAT and NR3SAT, it becomes apparent that NR3SAT does not include all positive literals in a clause, such that $Q_i^* \notin (q_i \vee q_j \vee q_k)$. This observation highlights the influence of the presence of this combination on the performance of 3SAT. The underlying reason for this is that there is no variation in terms of the individual contribution of synaptic weight for this combination, resulting in repetitive final neuron states. Consequently, these repetitive final neuron states exhibit high similarity with the benchmark states.

On the other side, TV values can be explained in terms of the efficiency of the models in exploring different solutions across various areas within the search space. TV can only be measured if the retrieved final neuron states satisfy Eq (13) [8]. As shown in Figure 10 and Table 20, the non-systematic logics achieved highest TV when the number of neurons falls within the range of 35 to 62. In contrast, 3SAT and NR3SAT consistently achieve higher TV values up to $NN < 72$. However, considering all logics, NR3SAT generates the highest TV compared to all baseline models. This is due to the presence of more global minima solutions in NR3SAT, which indirectly leads to higher TV values. These higher TV demonstrates the effectiveness of the proposed model in exploring different areas in the search space [16].

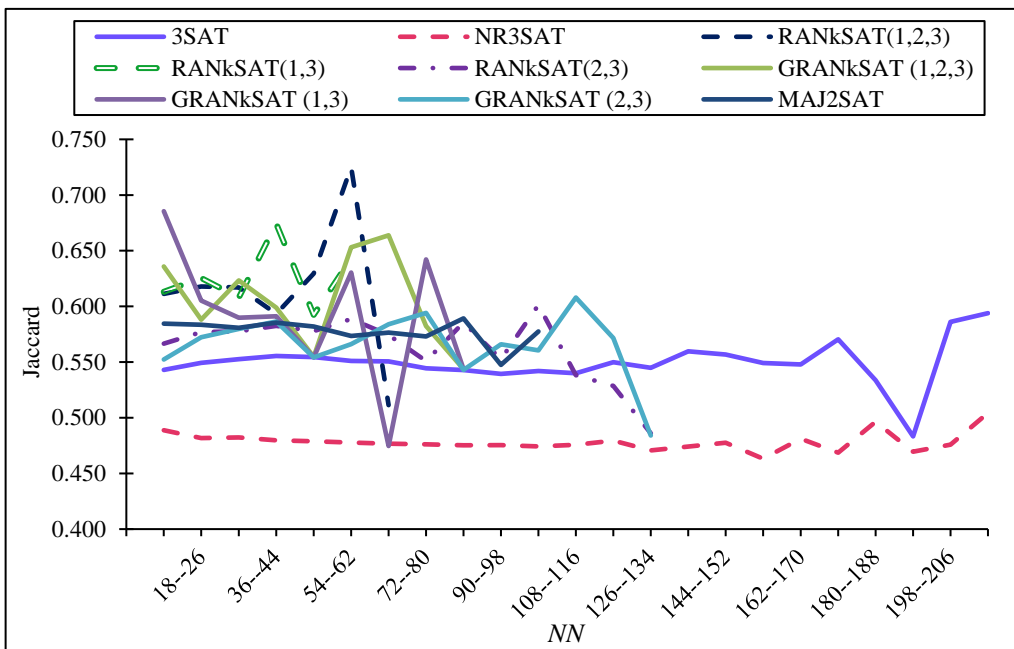


Figure 9. Jaccard value of NR3SAT with existing logic models.

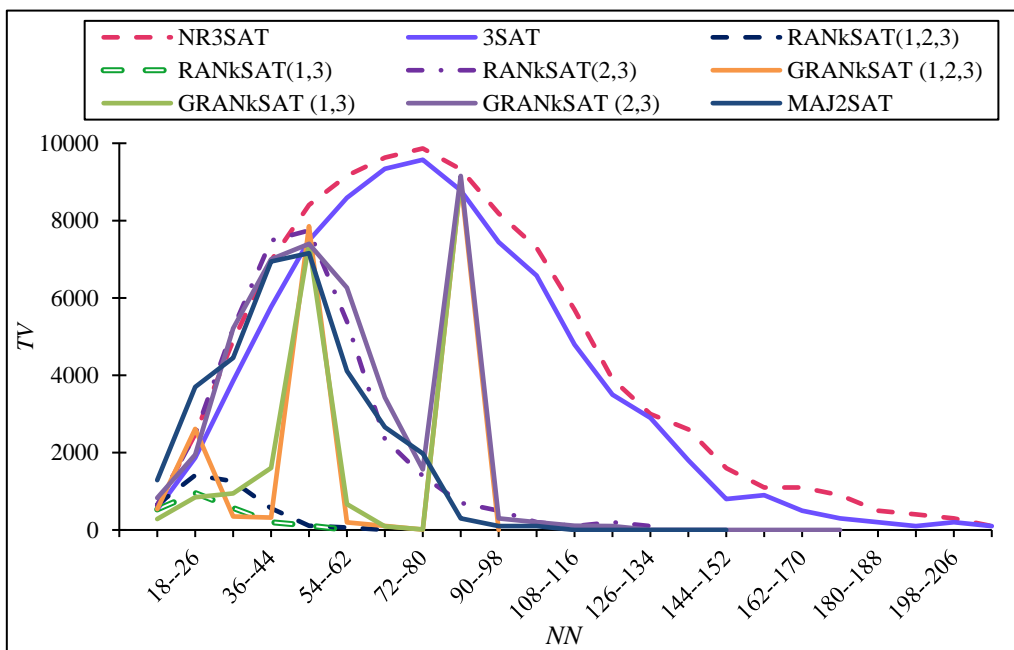


Figure 10. TV of NR3SAT with existing logic models.

In conclusion, based on the above analyses, NR3SAT has been successfully integrated into DHNN, resulting in the attainment of global minima solutions with diversified final neuron states. However, it is important to note that the performance of DHNN can be further improved by addressing certain limitations identified in this work. One notable limitation is the utilization of the proposed learning algorithm. It is evident that the trial-and-error approach in ES poses challenges in finding satisfied interpretations, especially as the number of neurons increases. This is because this approach requires

trying out all possible solutions in the search space to obtain satisfied interpretations. Moreover, this approach introduces time constraints, leading to suboptimal optimization processes. Ideally, when an algorithm is implemented into DHNN, it should possess the ability to rapidly identify optimal solutions. Therefore, implementation of learning algorithms is crucial to enhance the process of finding satisfied interpretation which led to optimal synaptic weights.

8.1.4. Non-parametric test

A non-parametric test, specifically known as Friedman test was employed in this study to assess the significance of the proposed NR3SAT in comparison to other baseline models. The importance of this test lies in its ability to determine whether there is a statistically significant difference in the performance among the models under consideration. This analysis is crucial for drawing meaningful conclusions about the effectiveness of the proposed NR3SAT in the context of DHNN based on various performance metrics, when compared to other existing models. For clear illustration, the hypothesis can be declared as follows:

- H_0 : There is no significant difference in the performance metrics between the proposed NR3SAT and other baseline models.
- H_1 : There is a significant difference in the performance metrics between the proposed NR3SAT and other baseline models.

In this study, the Statistical Package for the Social Sciences (SPSS) software was employed to conduct the Friedman test, and the results are presented in Table 21. Using a pre-defined significance level of 0.05, the null hypothesis will be rejected if the obtained p -value is less than 0.05. The observations from Table 21 clearly indicated that, for all performance metrics, the associated p -values were below 0.05. Consequently, all the null hypotheses are rejected, leading to the conclusion that a significant difference exists in the performance metrics between the proposed NR3SAT and other baseline models. Additionally, the mean rank of the models based on different performance metrics is provided in Table 22. In the context of learning errors, testing errors, and Jaccard values, lower values indicate good results. As for the global minima ratio and TV values, the maximum values of 1 for global ratio and higher values of TV indicate a positive impact on the overall performance.

Table 16. Tabulated $MAPE_{learn}$ with the bold values indicates NR3SAT obtained the lowest $MAPE_{learn}$ at NN and * indicates that there is no value generated for the particular NN . The bracket indicates the ratio of improvement.

NN	NR3SAT	3SAT	MAJ2SAT	RANkSAT (k = 1, 2, 3)	RANkSAT (k = 1, 3)	RANkSAT (k = 2, 3)	GRANkSAT (k = 1, 2, 3)	GRANkSAT (k = 1, 3)	GRANkSAT (k = 2, 3)
9-17	13.100	14.167 (-0.075)	39.125 (-0.665)	71.631 (-0.817)	66.639 (-0.803)	23.252 (-0.437)	70.247 (-0.814)	84.992 (-0.846)	15.5 (-0.155)
18-26	21.248	26.437 (-0.196)	68.867 (-0.691)	85.265 (-0.751)	90.223 (-0.764)	48.707 (-0.564)	78.463 (-0.729)	94.939 (-0.776)	38.993 (-0.455)
27-35	33.366	39.586 (-0.157)	62.08 (-0.463)	95.961 (-0.652)	96.084 (-0.653)	65.87 (-0.493)	97.875 (-0.659)	97.656 (-0.658)	65.86 (-0.493)
36-44	46.315	49.745 (-0.069)	81.272 (-0.43)	98.493 (-0.53)	98.83 (-0.531)	79.411 (-0.417)	98.909 (-0.532)	96.585 (-0.52)	79.991 (-0.421)
45-53	57.203	62.726 (-0.088)	87.492 (-0.346)	98.794 (-0.421)	99.01 (-0.422)	88.666 (-0.355)	61.812 (-0.075)	60.256 (-0.051)	60.256 (-0.051)
54-62	67.024	70.622 (-0.051)	94.842 (-0.293)	98.991 (-0.323)	99.01 (-0.323)	94.691 (-0.292)	98.947 (-0.323)	98.443 (-0.319)	91.722 (-0.269)
63-71	73.548	76.524 (-0.039)	96.048 (-0.234)	99.01 (-0.257)	99.01 (-0.257)	96.23 (-0.236)	98.929 (-0.257)	99.01 (-0.257)	95.838 (-0.233)
72-80	77.766	82.529 (-0.058)	98.276 (-0.209)	99.01 (-0.215)	99.01 (-0.215)	97.183 (-0.2)	99.01 (-0.215)	99.01 (-0.215)	98.388 (-0.21)
81-89	84.977	87.987 (-0.034)	98.805 (-0.14)	99.01 (-0.142)	*	98.556 (-0.138)	85.968 (-0.012)	85.968 (-0.012)	83.852 (0.013)
90-98	88.069	88.119 (-0.001)	98.686 (-0.108)	99.01 (-0.111)	*	98.636 (-0.107)	99.01 (-0.111)	99.01 (-0.111)	98.788 (-0.109)
99-107	89.669	92.163 (-0.027)	98.999 (-0.094)	*	*	98.87 (-0.093)	99.01 (-0.094)	99.01 (-0.094)	98.469 (-0.089)
108-116	91.121	93.593 (-0.026)	99.01 (-0.08)	*	*	98.986 (-0.079)	99.01 (-0.08)	99.01 (-0.08)	98.98 (-0.079)
117-125	94.553	94.746 (-0.002)	99.01 (-0.045)	*	*	98.971 (-0.045)	99.01 (-0.045)	99.01 (-0.045)	98.985 (-0.045)
126-134	95.591	96.29 (-0.007)	99.01 (-0.035)	*	*	98.52 (-0.03)	99.01 (-0.035)	99.01 (-0.035)	99.01 (-0.035)
135-143	96.173	97.225 (-0.011)	99.01 (-0.029)	*	*	99.01 (-0.029)	*	*	99.01 (-0.029)
144-152	98.190	98.626 (-0.004)	99.01 (-0.008)	*	*	99.01 (-0.008)	*	*	99.01 (-0.008)
153-161	98.090	98.386 (-0.003)	*	*	*	99.01 (-0.009)	*	*	99.01 (-0.009)
162-170	97.764	98.581 (-0.008)	*	*	*	*	*	*	99.01 (-0.013)
171-179	98.128	98.836 (-0.007)	*	*	*	*	*	*	99.01 (-0.009)
180-188	98.900	98.987 (-0.001)	*	*	*	*	*	*	*
189-197	98.912	98.995 (-0.001)	*	*	*	*	*	*	*
198-206	98.909	98.923 (0)	*	*	*	*	*	*	*
207-215	98.877	98.957 (-0.001)	*	*	*	*	*	*	*
(+/-/-)		23/0/0	23/0/0	23/0/0	23/0/0	23/0/0	23/0/0	23/0/0	23/0/0
Min	13.1000	14.1667	39.1254	71.6310	66.6387	23.2515	61.8120	60.2562	15.5000
Max	98.9124	98.9947	99.0098	99.0098	99.0098	99.0098	99.0098	99.0098	99.0098
Avg	79.0214	80.9891	88.7214	94.5174	93.4770	87.2693	91.8006	93.7078	85.2463

Table 17. Tabulated $MAPE_{test}$ with * indicates that there is no value generated for the particular NN . The bold values indicates that the proposed NR3SAT obtained the lowest testing error compared to baseline logical rules at the particular NN .

NN	NR3SAT	3SAT	MAJ2SAT	RANkSAT ($k = 1, 2, 3$)	RANkSAT ($k = 1, 3$)	RANkSAT ($k = 2, 3$)	GRANkSAT ($k = 1, 2, 3$)	GRANkSAT ($k = 1, 3$)	GRANkSAT ($k = 2, 3$)
9-17	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0009	0.0000
18-26	0.0000	0.0000	0.0000	0.0003	0.0019	0.0000	0.0000	0.0066	0.0000
27-35	0.0000	0.0000	0.0000	0.0066	0.0076	0.0000	0.0090	0.0078	0.0001
36-44	0.0000	0.0000	0.0002	0.0093	0.0096	0.0002	0.0095	0.0074	0.0001
45-53	0.0000	0.0000	0.0015	0.0098	0.0098	0.0009	0.0000	0.0000	0.0000
54-62	0.0000	0.0000	0.0057	0.0099	0.0100	0.0041	0.0098	0.0091	0.0035
63-71	0.0000	0.0000	0.0073	0.0100	0.0100	0.0076	0.0099	0.0099	0.0065
72-80	0.0000	0.0001	0.0080	0.0100	0.0100	0.0086	0.0100	0.0100	0.0084
81-89	0.0006	0.0011	0.0097	0.0100	*	0.0093	0.0009	0.0009	0.0007
90-98	0.0018	0.0025	0.0099	0.0100	*	0.0095	0.0100	0.0100	0.0097
99-107	0.0027	0.0034	0.0099	*	*	0.0098	0.0100	0.0100	0.0098
108-116	0.0043	0.0052	0.0100	*	*	0.0099	0.0100	0.0100	0.0099
117-125	0.0061	0.0065	0.0100	*	*	0.0098	0.0100	0.0100	0.0099
126-134	0.0070	0.0071	0.0100	*	*	0.0099	0.0100	0.0100	0.0100
135-143	0.0074	0.0082	0.0100	*	*	0.0100	*	*	0.0100
144-152	0.0084	0.0092	0.0100	*	*	0.0100	*	*	0.0100
153-161	0.0089	0.0091	*	*	*	0.0100	*	*	0.0100
162-170	0.0089	0.0095	*	*	*	*	*	*	0.0100
171-179	0.0091	0.0097	*	*	*	*	*	*	0.0100
180-188	0.0095	0.0098	*	*	*	*	*	*	*
189-197	0.0096	0.0099	*	*	*	*	*	*	*
198-206	0.0097	0.0098	*	*	*	*	*	*	*
207-215	0.0099	0.0099	*	*	*	*	*	*	*
(+/-/-)		15/8/0	20/3/0	22/1/0	22/1/0	20/3/0	21/2/0	23/0/0	21/2/0
Min	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Max	0.0099	0.0099	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
Avg	0.0045	0.0048	0.0064	0.0076	0.0074	0.0064	0.0071	0.0073	0.0062

Table 18. Tabulated R_G with * indicates that there is no value generated for the particular NN . The bold values indicates that the proposed NR3SAT obtained the highest ratio compared to baseline logical rules at the particular NN .

NN	NR3SAT	3SAT	MAJ2SAT	RANkSAT ($k = 1,2,3$)	RANkSAT ($k = 1,3$)	RANkSAT ($k = 2,3$)	GRANkSAT ($k = 1,2,3$)	GRANkSAT ($k = 1,3$)	GRANkSAT ($k = 2,3$)
9-17	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9100	1.0000
18-26	1.0000	1.0000	1.0000	0.9700	0.8100	1.0000	1.0000	0.3400	1.0000
27-35	1.0000	1.0000	1.0000	0.3400	0.2400	1.0000	0.1000	0.2200	0.9900
36-44	1.0000	1.0000	0.9800	0.0700	0.0400	0.9800	0.0473	0.2600	0.9900
45-53	1.0000	1.0000	0.8500	0.0200	0.0200	0.9100	1.0000	1.0000	1.0000
54-62	1.0000	1.0000	0.4309	0.0100	0.0007	0.5900	0.0200	0.0900	0.6500
63-71	1.0000	1.0000	0.2700	0.0003	0.0000	0.2400	0.0112	0.0100	0.3500
72-80	1.0000	0.9900	0.2003	0.0000	0.0000	0.1400	0.0008	0.0015	0.1600
81-89	0.9400	0.8900	0.0300	0.0000	*	0.0700	0.9100	0.9100	0.9300
90-98	0.8200	0.7500	0.0100	0.0000	*	0.0500	0.0000	0.0000	0.0300
99-107	0.7300	0.6600	0.0100	*	*	0.0201	0.0000	0.0000	0.0200
108-116	0.5700	0.4800	0.0000	*	*	0.0100	0.0000	0.0000	0.0110
117-125	0.3900	0.3500	0.0000	*	*	0.0200	0.0000	0.0000	0.0102
126-134	0.3000	0.2900	0.0000	*	*	0.0100	0.0000	0.0000	0.0003
135-143	0.2600	0.1800	0.0000	*	*	0.0000	*	*	0.0000
144-152	0.1600	0.0800	0.0000	*	*	0.0000	*	*	0.0000
153-161	0.1100	0.0900	*	*	*	0.0000	*	*	0.0000
162-170	0.1100	0.0500	*	*	*	*	*	*	0.0000
171-179	0.0900	0.0300	*	*	*	*	*	*	0.0000
180-188	0.0500	0.0200	*	*	*	*	*	*	*
189-197	0.0402	0.0100	*	*	*	*	*	*	*
198-206	0.0300	0.0200	*	*	*	*	*	*	*
207-215	0.0100	0.0101	*	*	*	*	*	*	*
(+/-/-)		16/7/0	20/3/0	22/1/0	22/1/0	20/3/0	20/3/0	22/1/0	20/3/0
Min	0.0100	0.0100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Max	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Avg	0.5483	0.5174	0.3613	0.2410	0.2638	0.3553	0.2921	0.2673	0.3759

Table 19. Tabulated Jaccard values with * indicates that there is no value generated for the particular *NN*. The bold values indicates that the proposed NR3SAT obtained the lowest Jaccard values compared to all baseline logical rules at the particular *NN*. The bracket indicates the ratio of improvement with negative values indicate the proposed logic outperform the baseline logics in generating lower Jaccard value.

<i>NN</i>	NR3SAT	3SAT	MAJ2SAT	RAN <i>k</i> SAT (<i>k</i> = 1, 2, 3)	RAN <i>k</i> SAT (<i>k</i> = 1, 3)	RAN <i>k</i> SAT (<i>k</i> = 2, 3)	GRAN <i>k</i> SAT (<i>k</i> = 1, 2, 3)	GRAN <i>k</i> SAT (<i>k</i> = 1, 3)	GRAN <i>k</i> SAT (<i>k</i> = 2, 3)
9-17	0.489	0.543 (-0.1)	0.585 (-0.164)	0.611 (-0.2)	0.613 (-0.203)	0.567 (-0.137)	0.636 (-0.231)	0.685 (-0.287)	0.552 (-0.115)
18-26	0.482	0.549 (-0.123)	0.584 (-0.174)	0.618 (-0.22)	0.625 (-0.23)	0.577 (-0.165)	0.588 (-0.181)	0.605 (-0.204)	0.572 (-0.158)
27-35	0.482	0.553 (-0.127)	0.581 (-0.17)	0.617 (-0.218)	0.608 (-0.207)	0.578 (-0.165)	0.623 (-0.226)	0.59 (-0.182)	0.58 (-0.168)
36-44	0.48	0.556 (-0.137)	0.585 (-0.181)	0.593 (-0.191)	0.674 (-0.289)	0.582 (-0.177)	0.599 (-0.199)	0.591 (-0.189)	0.586 (-0.182)
45-53	0.479	0.554 (-0.136)	0.582 (-0.177)	0.63 (-0.239)	0.592 (-0.192)	0.578 (-0.171)	0.555 (-0.137)	0.554 (-0.136)	0.554 (-0.136)
54-62	0.478	0.551 (-0.133)	0.573 (-0.167)	0.725 (-0.341)	0.643 (-0.257)	0.589 (-0.188)	0.653 (-0.269)	0.63 (-0.242)	0.566 (-0.156)
63-71	0.477	0.551 (-0.134)	0.577 (-0.173)	0.511 (-0.068)	*	0.574 (-0.17)	0.664 (-0.282)	0.475 (0.004)	0.584 (-0.183)
72-80	0.476	0.544 (-0.125)	0.573 (-0.169)	*	*	0.551 (-0.136)	0.582 (-0.182)	0.642 (-0.259)	0.594 (-0.199)
81-89	0.475	0.543 (-0.125)	0.589 (-0.193)	*	*	0.585 (-0.188)	0.544 (-0.126)	0.544 (-0.126)	0.543 (-0.125)
90-98	0.475	0.539 (-0.119)	0.547 (-0.132)	*	*	0.553 (-0.14)	*	*	0.566 (-0.16)
99-107	0.474	0.542 (-0.125)	0.577 (-0.179)	*	*	0.601 (-0.211)	*	*	0.561 (-0.154)
108-116	0.476	0.54 (-0.119)	*	*	*	0.538 (-0.116)	*	*	0.608 (-0.218)
117-125	0.479	0.55 (-0.128)	*	*	*	0.529 (-0.093)	*	*	0.572 (-0.161)
126-134	0.471	0.545 (-0.136)	*	*	*	0.486 (-0.032)	*	*	0.484 (-0.028)
135-143	0.474	0.56 (-0.153)	*	*	*	*	*	*	*
144-152	0.478	0.557 (-0.142)	*	*	*	*	*	*	*
153-161	0.463	0.549 (-0.157)	*	*	*	*	*	*	*
162-170	0.481	0.548 (-0.122)	*	*	*	*	*	*	*
171-179	0.469	0.57 (-0.178)	*	*	*	*	*	*	*
180-188	0.496	0.534 (-0.07)	*	*	*	*	*	*	*
189-197	0.47	0.483 (-0.028)	*	*	*	*	*	*	*
198-206	0.476	0.586 (-0.188)	*	*	*	*	*	*	*
207-215	0.504	0.594 (-0.151)	*	*	*	*	*	*	*
(+/-/-)		23/0/0	23/0/0	23/0/0	23/0/0	23/0/0	23/0/0	23/0/0	23/0/0
Min	0.4632	0.4833	0.5475	0.5114	0.5924	0.4860	0.5436	0.4747	0.4841
Max	0.5043	0.5939	0.5892	0.7251	0.6743	0.6012	0.6638	0.6855	0.6079
Avg	0.4784	0.5496	0.5776	0.6150	0.6261	0.5634	0.6048	0.5908	0.5659

Table 20. Tabulated *TV* values with * indicates that there is no value generated for the particular *NN*. The bold values indicates that the model able to obtain the highest *TV* compared to all logical rules at the particular *NN*. The bracket indicates the ratio of improvement with positive values indicate the proposed logic outperform the baseline logics in generating higher *TV* value.

<i>NN</i>	NR3SAT	3SAT	MAJ2SAT	RANkSAT (<i>k</i> = 1,2,3)	RANkSAT (<i>k</i> = 1,3)	RANkSAT (<i>k</i> = 2,3)	GRANkSAT (<i>k</i> = 1,2,3)	GRANkSAT (<i>k</i> = 1,3)	GRANkSAT (<i>k</i> = 2,3)
9-17	636	529 (0.2023)	1287 (-0.5058)	653 (-0.026)	533 (0.1932)	645 (-0.014)	539 (0.18)	284 (1.2394)	830 (-0.2337)
18-26	2476	1866 (0.3269)	3699 (-0.3306)	1416 (0.7486)	955 (1.5927)	2477 (-0.0004)	2613 (-0.0524)	846 (1.9267)	1947 (0.2717)
27-35	4894	3854 (0.2698)	4452 (0.0993)	1260 (2.8841)	568 (7.6162)	5213 (-0.0612)	349 (13.0229)	947 (4.1679)	5213 (-0.0612)
36-44	6969	5766 (0.2086)	6941 (0.004)	561 (11.4225)	199 (34.0201)	7483 (-0.0687)	322 (20.6429)	1602 (3.3502)	7002 (-0.0047)
45-53	8404	7488 (0.1223)	7157 (0.1742)	110 (75.4)	111 (74.7117)	7745 (0.0851)	7855 (0.0699)	7399 (0.1358)	7399 (0.1358)
54-62	9160	8589 (0.0665)	4113 (1.2271)	64 (142.125)	4 (2289)	5391 (0.6991)	195 (45.9744)	667 (12.7331)	6262 (0.4628)
63-71	9629	9338 (0.0312)	2658 (2.6226)	3 (3208.6667)	*	2353 (3.0922)	106 (89.8396)	94 (101.4362)	3428 (1.8089)
72-80	9866	9573 (0.0306)	1974 (3.998)	*	*	1398 (6.0572)	8 (1232.25)	14 (703.7143)	1566 (5.3001)
81-89	9325	8780 (0.0621)	300 (30.0833)	*	*	700 (12.3214)	9034 (0.0322)	9034 (0.0322)	9155 (0.0186)
90-98	8183	7445 (0.0991)	100 (80.83)	*	*	500 (15.366)	*	*	300 (26.2767)
99-107	7298	6578 (0.1095)	100 (71.98)	*	*	200 (35.49)	*	*	200 (35.49)
108-116	5696	4796 (0.1877)	*	*	*	100 (55.96)	*	*	110 (50.7818)
117-125	3900	3500 (0.1143)	*	*	*	200 (18.5)	*	*	102 (37.2353)
126-134	3000	2898 (0.0352)	*	*	*	100 (29)	*	*	3 (999)
135-143	2600	1800 (0.4444)	*	*	*	*	*	*	*
144-152	1600	800 (1)	*	*	*	*	*	*	*
153-161	1100	900 (0.2222)	*	*	*	*	*	*	*
162-170	1100	500 (1.2)	*	*	*	*	*	*	*
171-179	900	300 (2)	*	*	*	*	*	*	*
180-188	500	200 (1.5)	*	*	*	*	*	*	*
189-197	402	100 (3.02)	*	*	*	*	*	*	*
198-206	300	200 (0.5)	*	*	*	*	*	*	*
207-215	100	100 (0)	*	*	*	*	*	*	*
Min	100	100	0	3	4	100	0	14	0
Max	9866	9573	7157	1416	955	7745	9034	9034	9155
Avg	4263	3735	2049	581	395	2465	2102	2321	2290

Thus, it can be observed that the proposed NR3SAT achieved the lowest mean rank for learning errors, testing errors, and Jaccard values, indicating that this proposed model can generate the lowest errors and has high dissimilarity between the retrieved final neuron states and the benchmark states. On the other hand, the proposed NR3SAT attained the highest mean rank for global ratio and TV , indicating that NR3SAT attained more global solutions, and the model explored different solutions in various areas of the search space compared to other baseline models. Therefore, it can be concluded that the proposed NR3SAT has been successfully integrated into the DHNN framework, enhancing the overall performance of DHNN.

Table 21. Friedman test between the proposed NR3SAT model and existing models.

	$MAPE_{learn}$	$MAPE_{test}$	R_G	Jaccard	TV
T	51.437	42.358	42.564	39.243	26.061
p -value	2.1614×10^{-8}	1.1602×10^{-6}	1.0612×10^{-6}	4.4297×10^{-6}	0.001
Conclusion	Reject H_0	Reject H_0	Reject H_0	Reject H_0	Reject H_0

Table 22. Mean rank values of NR3SAT and other baseline models with different performance metrics.

	$MAPE_{learn}$	$MAPE_{test}$	R_G	Jaccard	TV
NR3SAT	1.00	2.38	7.63	1.00	7.17
3SAT	2.38	2.50	7.50	2.33	5.00
MAJ2SAT	4.63	4.25	5.75	5.00	6.33
RANkSAT ($k = 1, 2, 3$)	7.44	7.13	2.81	7.83	3.33
RANkSAT ($k = 1, 3$)	7.81	7.75	2.06	7.83	1.83
RANkSAT ($k = 2, 3$)	4.38	4.38	5.63	4.17	7.25
GRANkSAT ($k = 1, 2, 3$)	6.94	6.13	4.00	7.33	4.33
GRANkSAT ($k = 1, 3$)	7.00	6.88	3.25	6.08	2.92
GRANkSAT ($k = 2, 3$)	3.44	3.63	6.38	3.42	6.83

8.2. Performance analysis of optimizing retrieval phase of ℓ_{NR3SAT}

In optimizing the retrieval phase of DHNN, it is essential to ensure that all the retrieved final neurons state attained global minima solutions. Therefore, we implemented EA during the learning phase of DHNN to acknowledge all the limitations mentioned in the previous section. Researchers [28,33,39,51] employed EA as the learning algorithm to handle the task of SAT minimization within DHNN. By implementing this optimization algorithm, EA can significantly improve the converge of network towards minimizing the cost function while maintaining a high level of dissimilarity compared to the benchmark states. However, the diversification of the solutions particularly concerning negativity remains uncertain. Therefore, incorporating a degree of negativity in the final neuron states becomes imperative to boost DHNN's performance in classification tasks. To achieve this, HBHA was employed during the retrieval phase of DHNN to optimize the final neuron states. These optimized solutions aim to satisfy the multi-objective function outlined in Eqs (15)–(17), with a specific target of achieving negativity within each solution, $Star_i : S_1^f \ S_2^f \ S_3^f \ S_4^f \ S_5^f \ \dots \ S_{NN}^f$. This raises the question of to what extent HBHA can maximize negativity while ensuring the attainment of global solutions and maintaining low similarity indices. Therefore, the performance of HBHA in maximizing the degree of negativity within the solution states will be thoroughly evaluated.

Figure 11 illustrated the impact of optimizing the retrieval phase of DHNN through HBHA in achieving a degree of negativity in the solution states while preserving the attainment of global solutions in comparison to not optimizing the retrieved final neuron states. Several observations can be drawn from Figure 11. First, it is evident that the implementation of a metaheuristic algorithm significantly enhances the performance of DHNN in achieving maximum diversification while retaining the ability to obtain global solutions. As indicated by the black line representing the state "Before HBHA," without optimizing the retrieved final neuron states, the solutions obtained fail to achieve maximum diversity. With the implementation of HBHA, optimized final neuron states can achieved maximum diversity and achieve global solutions. Based on the findings presented in Figure 12, it can be concluded that HBHA performs optimally when the degree of negativity in a solution string is set to be at least 60%. However, when the number of neurons exceeds 81, HBHA struggles to ensure all solution strings can achieve maximum diversity and achieve global minima energy. To address this challenge, parameter tuning was conducted by specifically focusing on adjusting the rate of replicating the best solutions. HBHA with at least 60% negativity performs effectively when $\beta = 0.7$ ensuring that all optimized final neuron states achieve both diversity and global minima energy. Additionally, the performance of HBHA start to drop when degree of negativity is set to be at least 70% because of the limited solution space to obtained optimal solutions. Besides that, although all solution strings can achieve maximum diversity and reach global minima energy when m is in the range of 0.1 to 0.5 as shown in Figure 11, the optimized states encounter overfitting issue. This occurs because when number of percentages is too low, there is a high tendency to obtain repetitive final neuron states due to the lack of variation in terms of negativity.

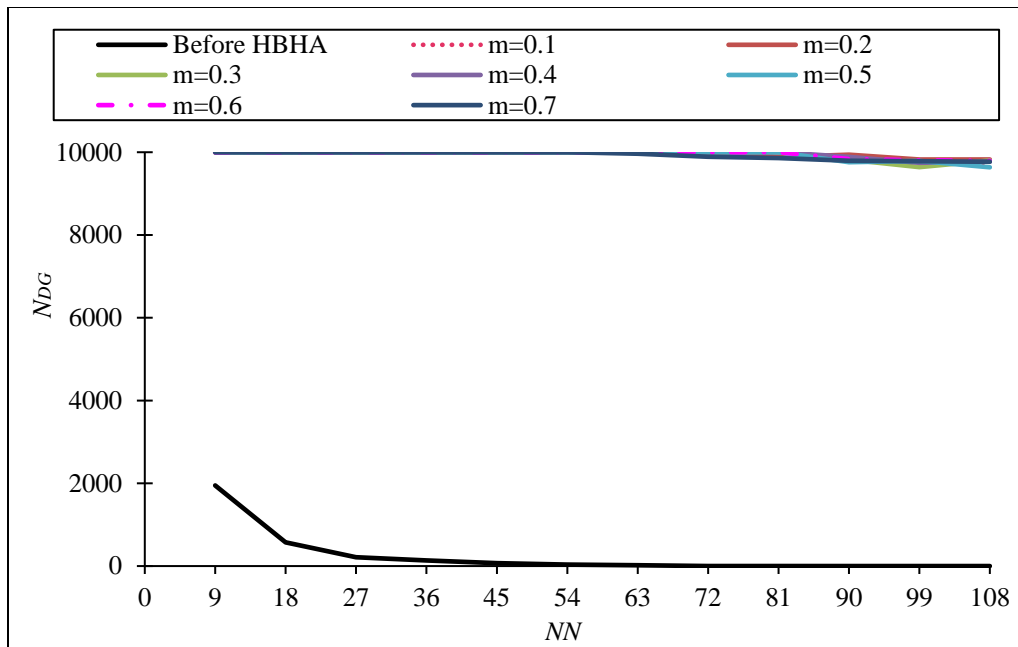


Figure 11. N_{DG} of HBHA with different m .

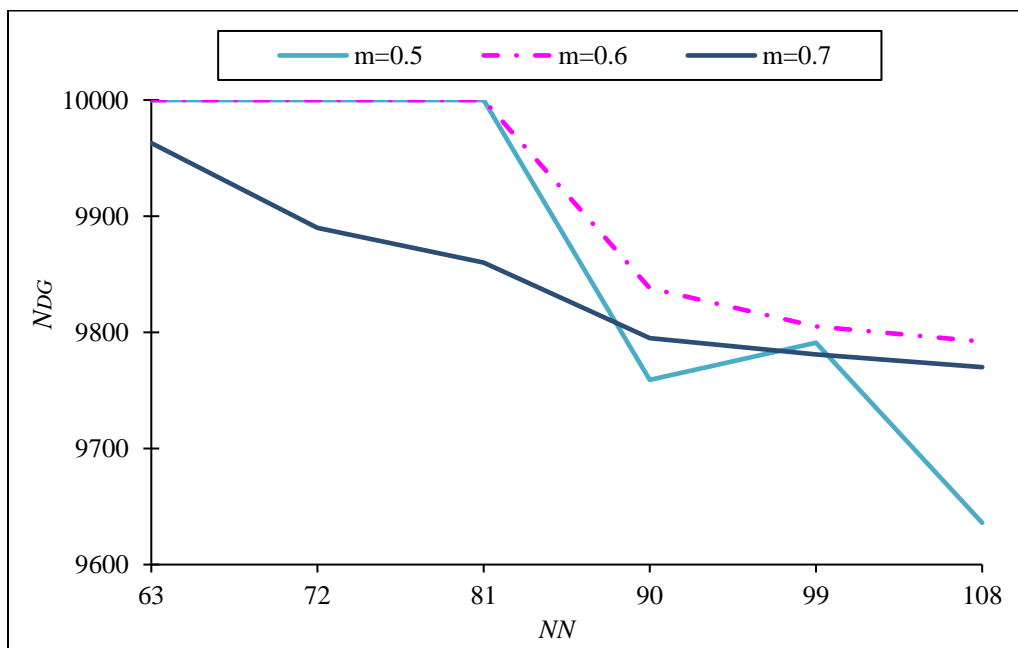


Figure 12. N_{DG} of HBHA when $m=0.5$, $m=0.6$ and $m=0.7$.

To assess the effectiveness of HBHA in achieving the desired at least 60% negativity in a solution, the performance of HBHA will be compared with 10 baseline algorithms. The evaluation will focus on achieving high diversity, attaining a high number of global solutions and obtaining solutions that exhibit both diversity and attained global solutions with low similarity indices. A performance comparison between HBHA and 10 other algorithms with a focus on diversity, N_{DS} are provided as in Figure 13 and Table 23. In Figure 13, it is evident that several algorithms including ABCK, GA, ABCJ, ABCS and DEA fail to attain maximum N_{DS} when the negativity variation is set to be at least 60%.

One contributing factor to these findings is the lack of an efficient exploitation process in these algorithms. For example, in the case of GA, the mutation process is designed to enhance exploitation but it only selects one random chromosome out of 100 solution sets to mutate. Regarding the ABC algorithms, all three of them primarily focus on global search by introducing various bitwise operators to update the positions of employed bees. However, they lack an exploitation process to improve neighbourhood solutions, which ultimately results in low N_{DS} being obtained. Another issue related to the lower N_{DS} values for ABCK is the implementation of bitwise operations to update the employed bee solutions. In this scenario, the proposed bitwise operator results in more 1 output compared to -1. This indirectly reduces the efficiency of ABC in producing more negative states compared to positive states. Other than that, an issue can also be observed in the DEA process, where there is an imbalance between the exploration and exploitation processes. The absence of an effective exploitation process in DEA leads to a lower N_{DS} of solutions obtained.

In contrast, several algorithms demonstrated exceptional performance in achieving maximum N_{DS} as shown in Table 23. It is worth noting that SCA, EA, WOA and BHA exhibited an increase in N_{DS} and were able to achieve maximum N_{DS} as the number of neurons increases. However, when considering overall performance, HBHA and CSA stood out as the only algorithms that consistently achieved maximum diversity across all neuron numbers. One shared feature of these two algorithms was their capacity for duplicating solutions to boost the performance of the algorithms. In CSA, a cloning process was employed to clone the best solution using a predefined clone rate 0.5 based on the total number of populations. In the case of HBHA, solution replication was performed based on the top best solution obtained, utilizing a replication rate of 0.7. This highlights the effectiveness of the operator in improving global search capabilities, leading to their remarkable achievement of maximum diversity. From the perspective of global minima solutions, N_{GR} the performance of all algorithms was evaluated to determine whether the solution converge to local or global solution while achieving maximum diversity. The result as presented in Table 24 and Figure 14 reveal a notable trend such that BHA, SCA, WOA and EA experience a significant decline in performance when striving to achieve a maximum global solution. This contrasts with the performance trends depicted in Figure 13. This observation implies that while these algorithms are effective at attaining a high N_{DS} , the solutions they produce may not necessarily be global since it led to unsatisfied interpretation. This is due to the updating rule concept in BHA, SCA and WOA where updates occur for each state of the solutions string. While this process enhances diversity, the alterations in each state disrupt the attainment of a satisfied interpretation, resulting in lower global solutions [26,44,50].

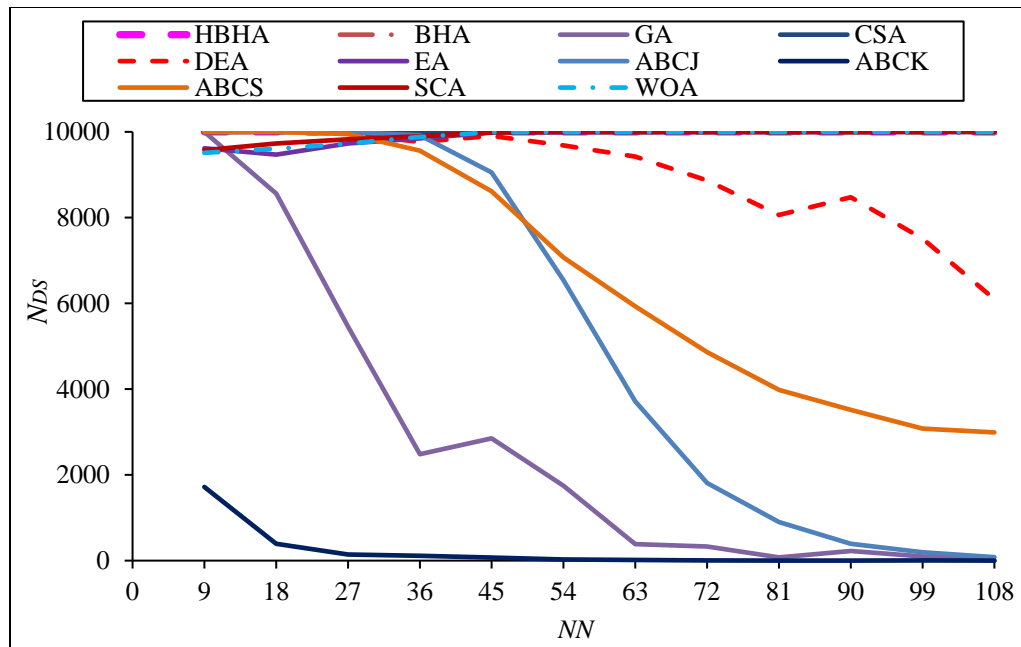


Figure 13. N_{DS} of HBHA with baseline algorithm. Refer Table 23 for details.

A similar situation arises in EA where at each stage of the algorithm, the random mutation process contributes to an increased chance of achieving the desired negativity rate. However, this also leads to a reduced probability of obtaining a satisfied interpretation, resulting in lower global solutions. In other words, random mutations may introduce alterations to the solution that deviate from the conditions required for a solution to be considered satisfied. Another observation from Table 24 and Figure 14 is the remarkable performance of DEA, ABCJ, ABCK, ABCS, CSA and HBHA in consistently maintaining global solutions across all numbers of neurons. While ABCJ, ABCK, ABCS and DEA may not have excelled in achieving high N_{DS} , it is important to highlight that all the solutions generated by these algorithms consistently exhibited are global. One contributing factor to this finding is the implementation of bitwise operators to update the position of employed bees, facilitating a higher output of 1. This increases the probability of obtaining satisfied interpretation, leading to increase chances of obtaining global minima solutions. In the case of HBHA and CSA, the optimized final neuron states consistently achieved a global ratio of 1. This indicates that the cloning operator in both algorithms which expands the solution space has enhanced the exploration of optimal solutions and has a significant impact on the overall performance of HBHA and CSA.

Figure 15 and Table 25 illustrates the performance of HBHA and other baseline algorithms in achieving both maximum diversity and a maximum global solutions, N_{DG} . As the number of neurons increases, several algorithms including ABCJ, ABCK, EA, WOA, SCA and GA encounter challenges in obtaining N_{DG} . This highlights that, as these algorithms aim to achieve the desired negativity in the solution string, it has an impact on the state of the neurons. Notably, GA exhibits the poorest performance in optimizing the retrieved final neuron states as the algorithm failed to achieve any of the predefined objectives. One significant factor contributing to this outcome is the

absence of additional operators in these algorithms to enhance the exploration of global solutions. As highlighted by Karim et al. [51], EA, ABC and GA face limitations in optimizing the learning phase of DHNN to fulfil multi-objective functions due to constraints in improving local solutions. The findings also reveal that only HBHA and CSA possess the capability to achieve N_{DG} . However, the quality of the optimized final neuron states obtained by these algorithms remain uncertain. Figure 16 and Table 26 presents an evaluation of the quality of the optimized final neuron states obtained through various algorithms. During the retrieval phase of DHNN, the evaluation of the retrieved final neuron states involved a comparison with benchmark states.

However, in optimizing the retrieved final neurons state, the assessment focuses on comparing the quality of the optimized final neuron states to their initial, unoptimized counterparts. The low values of RT indicate a significant dissimilarity between the states before and after optimization and the high values of RT indicate high similarity between before and after optimization took place. In this context, RT was chosen to assess the similarity within the solutions strings as this metric able to measure the similarity of negative cases [16]. Based on the results in Figure 16, ABCK demonstrated the worst performance. Regardless of the number of neurons, the similarity index RT consistently achieved the maximum value of 1. This indicates that there was no improvement in the quality of the final neuron states after optimization. The performance patterns of ABCJ and ABCS also show a significant increase in the similarity of the optimized final neuron states as the number of neurons increases. In contrast, when comparing HBHA with all the algorithms, it is evident that HBHA consistently achieved the lowest values of RT for most of the number of neurons, indicating that DHNN produced high-quality final neuron states through the HBHA optimization process. The quantitative performance of the proposed HBHA and other baseline algorithms is presented as in Table 27. Based on the findings, it can be concluded that HBHA was effectively implemented to optimize the retrieval phase of DHNN, successfully achieving all the objectives outlined in Eqs (15)–(17). The Friedman test is used to assess the significant difference between the proposed HBHA and other baseline algorithms in general for a specific metric. For clarity, the null hypothesis and the alternative hypothesis for the Friedman test are outlined as follows:

- H_0 : There is no significant difference in the performance metrics between the proposed HBHA and other baseline algorithms.
- H_1 : There is a significant difference in the performance metrics between the proposed HBHA and other baseline algorithms.

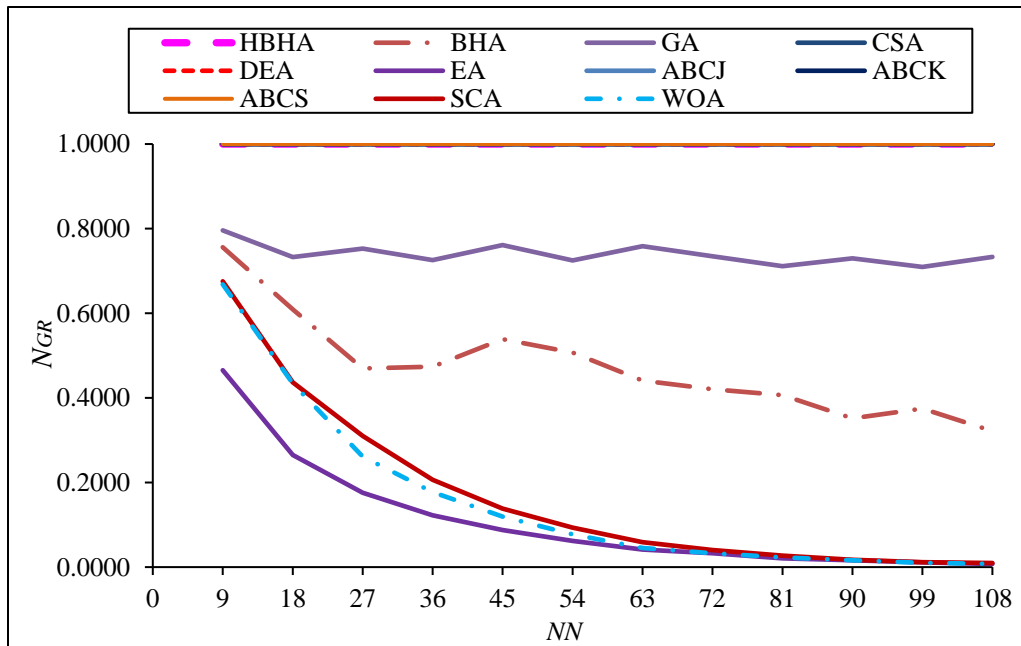


Figure 14. N_{GR} of HBHA with baseline algorithm. Refer Table 24 for details comparison.

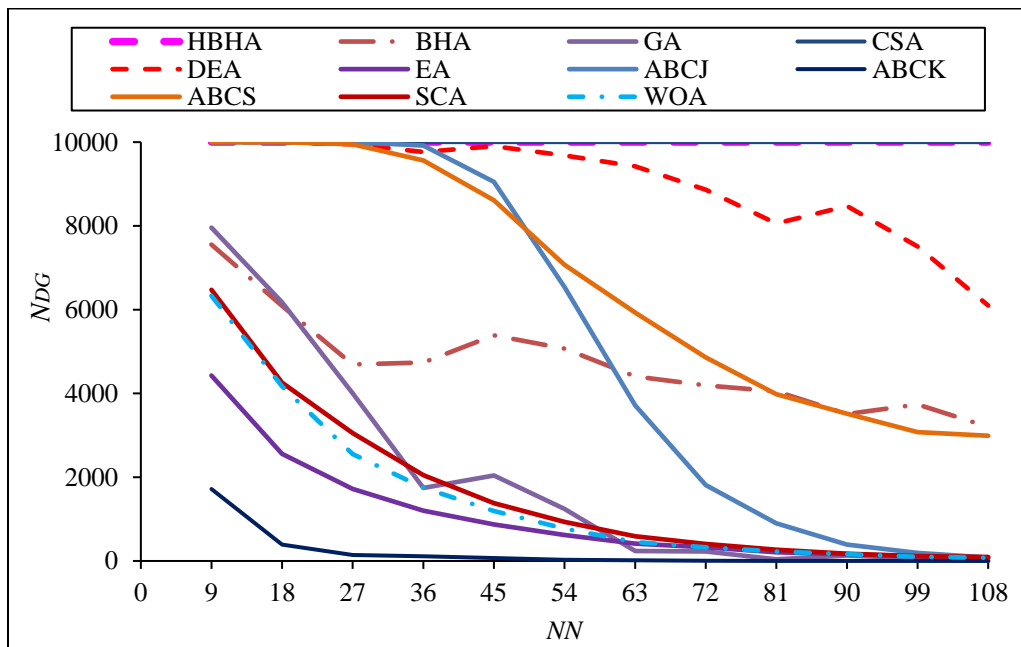


Figure 15. N_{DG} of HBHA and other baseline algorithms. Refer Table 25 for details comparison.

Table 23. Tabulated N_{DS} values of HBHA and baseline algorithms. The bracket indicates the ratio of improvement with * denotes there is no ratio of improvement since the baseline obtained zero value at that NN.

NN	HBHA	BHA	GA	CSA	DEA	EA	ABCJ	ABCK	ABCS	SCA	WOA
9	10000	9992 (0.001)	10000 (0)	10000 (0)	10000 (0)	9616 (0.04)	10000 (0)	1718 (4.821)	10000 (0)	9572 (0.045)	9511 (0.051)
18	10000	9999 (0)	8562 (0.168)	10000 (0)	9999 (0)	9469 (0.056)	10000 (0)	391 (24.575)	10000 (0)	9728 (0.028)	9607 (0.041)
27	10000	10000 (0)	5458 (0.832)	10000 (0)	9951 (0.005)	9728 (0.028)	10000 (0)	141 (69.922)	9948 (0.005)	9827 (0.018)	9725 (0.028)
36	10000	10000 (0)	2482 (3.029)	10000 (0)	9765 (0.024)	9849 (0.015)	9918 (0.008)	114 (86.719)	9562 (0.046)	9897 (0.01)	9881 (0.012)
45	10000	10000 (0)	2852 (2.506)	10000 (0)	9905 (0.01)	9982 (0.002)	9052 (0.105)	70 (141.857)	8612 (0.161)	9986 (0.001)	9991 (0.001)
54	10000	10000 (0)	1746 (4.727)	10000 (0)	9683 (0.033)	9991 (0.001)	6542 (0.529)	28 (356.143)	7071 (0.414)	9996 (0)	9993 (0.001)
63	10000	9998 (0)	384 (25.042)	10000 (0)	9422 (0.061)	9995 (0.001)	3713 (1.693)	17 (587.235)	5929 (0.687)	9997 (0)	9993 (0.001)
72	10000	9989 (0.001)	328 (29.488)	10000 (0)	8865 (0.128)	9997 (0)	1812 (4.519)	6 (1665.667)	4862 (1.057)	9996 (0)	9994 (0.001)
81	10000	9983 (0.002)	74 (134.135)	10000 (0)	8059 (0.241)	9994 (0.001)	899 (10.124)	2 (4999)	3981 (1.512)	9999 (0)	9997 (0)
90	10000	9991 (0.001)	223 (43.843)	10000 (0)	8475 (0.18)	10000 (0)	392 (24.51)	2 (4999)	3515 (1.845)	9999 (0)	10000 (0)
99	10000	9990 (0.001)	96 (103.167)	10000 (0)	7507 (0.332)	10000 (0)	193 (50.814)	3 (3332.333)	3078 (2.249)	10000 (0)	10000 (0)
108	10000	9976 (0.002)	0 (*)	10000 (0)	6099 (0.64)	10000 (0)	81 (122.457)	1 (9999)	2990 (2.345)	10000 (0)	10000 (0)
(+/-/-)		8/4/0	11/1/0	0/12/0	11/1/0	9/3/0	9/3/0	12/0/0	10/2/0	10/2/0	9/3/0
Min	10000	9976	0	10000	6099	9469	81	1	2990	9572	9511
Max	10000	10000	10000	10000	10000	10000	10000	1718	10000	10000	10000
Avg	10000	9993	2684	10000	8978	9885	5217	208	6629	9916	9891

Table 24. Tabulated N_{GR} values of HBHA and other baseline algorithms. The bracket indicates the ratio of improvement between proposed HBHA and baseline algorithms.

<i>NN</i>	HBHA	BHA	GA	CSA	DEA	EA	ABCJ	ABCK	ABCS	SCA	WOA
9	1	0.756 (0.323)	0.796 (0.256)	1 (0)	1 (0)	0.465 (1.149)	1 (0)	1 (0)	1 (0)	0.676 (0.48)	0.668 (0.497)
18	1	0.609 (0.642)	0.733 (0.365)	1 (0)	1 (0)	0.265 (2.772)	1 (0)	1 (0)	1 (0)	0.437 (1.289)	0.434 (1.304)
27	1	0.469 (1.13)	0.753 (0.328)	1 (0)	1 (0)	0.176 (4.692)	1 (0)	1 (0)	1 (0)	0.31 (2.227)	0.261 (2.827)
36	1	0.474 (1.111)	0.726 (0.378)	1 (0)	1 (0)	0.123 (7.163)	1 (0)	1 (0)	1 (0)	0.206 (3.845)	0.177 (4.643)
45	1	0.539 (0.856)	0.761 (0.314)	1 (0)	1 (0)	0.088 (10.429)	1 (0)	1 (0)	1 (0)	0.138 (6.225)	0.12 (7.368)
54	1	0.507 (0.973)	0.725 (0.38)	1 (0)	1 (0)	0.062 (15.129)	1 (0)	1 (0)	1 (0)	0.093 (9.707)	0.078 (11.903)
63	1	0.441 (1.268)	0.759 (0.318)	1 (0)	1 (0)	0.042 (23.039)	1 (0)	1 (0)	1 (0)	0.059 (16.007)	0.045 (21.173)
72	1	0.421 (1.378)	0.735 (0.361)	1 (0)	1 (0)	0.033 (29.488)	1 (0)	1 (0)	1 (0)	0.041 (23.691)	0.033 (29.675)
81	1	0.407 (1.459)	0.711 (0.406)	1 (0)	1 (0)	0.021 (47.077)	1 (0)	1 (0)	1 (0)	0.027 (36.037)	0.023 (42.86)
90	1	0.352 (1.844)	0.73 (0.37)	1 (0)	1 (0)	0.016 (61.112)	1 (0)	1 (0)	1 (0)	0.017 (56.804)	0.016 (60.728)
99	1	0.375 (1.669)	0.71 (0.409)	1 (0)	1 (0)	0.011 (88.286)	1 (0)	1 (0)	1 (0)	0.012 (85.207)	0.01 (97.039)
108	1	0.322 (2.11)	0.733 (0.364)	1 (0)	1 (0)	0.007 (134.135)	1 (0)	1 (0)	1 (0)	0.01 (103.167)	0.008 (128.87)
(+/-/-)		12/0/0	12/0/0	0/12/0	0/12/0	12/0/0	0/12/0	0/12/0	0/12/0	12/0/0	12/0/0
Min	1.000	0.322	0.710	1.000	1.000	0.007	1.000	1.000	1.000	0.010	0.008
Max	1.000	0.756	0.796	1.000	1.000	0.465	1.000	1.000	1.000	0.676	0.668
Avg	1.000	0.473	0.739	1.000	1.000	0.109	1.000	1.000	1.000	0.169	0.156

Table 25. Tabulated N_{DG} values of HBHA versus baseline algorithms. The bracket indicates the ratio of improvement with * denotes there is no ratio of improvement since the baseline obtained zero value at that NN .

NN	HBHA	BHA	GA	CSA	DEA	EA	ABCJ	ABCK	ABCS	SCA	WOA
9	10000	7554 (0.3238)	7959 (0.2564)	10000 (0)	10000 (0)	4427 (1.2589)	10000 (0)	1718 (4.8207)	10000 (0)	6474 (0.5446)	6331 (0.5795)
18	10000	6090 (0.642)	6174 (0.6197)	10000 (0)	9999 (0.0001)	2558 (2.9093)	10000 (0)	391 (24.5754)	10000 (0)	4256 (1.3496)	4168 (1.3992)
27	10000	4694 (1.1304)	4004 (1.4975)	10000 (0)	9951 (0.0049)	1720 (4.814)	10000 (0)	141 (69.922)	9948 (0.0052)	3055 (2.2733)	2554 (2.9154)
36	10000	4738 (1.1106)	1742 (4.7405)	10000 (0)	9765 (0.0241)	1202 (7.3195)	9918 (0.0083)	114 (86.7193)	9562 (0.0458)	2049 (3.8804)	1749 (4.7176)
45	10000	5389 (0.8556)	2043 (3.8948)	10000 (0)	9905 (0.0096)	874 (10.4416)	9052 (0.1047)	70 (141.8571)	8612 (0.1612)	1381 (6.2411)	1193 (7.3822)
54	10000	5069 (0.9728)	1245 (7.0321)	10000 (0)	9683 (0.0327)	620 (15.129)	6542 (0.5286)	28 (356.1429)	7071 (0.4142)	932 (9.7296)	774 (11.9199)
63	10000	4408 (1.2686)	240 (40.6667)	10000 (0)	9422 (0.0613)	416 (23.0385)	3713 (1.6932)	17 (587.2353)	5929 (0.6866)	588 (16.0068)	451 (21.1729)
72	10000	4194 (1.3844)	226 (43.2478)	10000 (0)	8865 (0.128)	328 (29.4878)	1812 (4.5188)	6 (1665.6667)	4862 (1.0568)	405 (23.6914)	326 (29.6748)
81	10000	4049 (1.4697)	41 (242.9024)	10000 (0)	8059 (0.2408)	208 (47.0769)	899 (10.1235)	2 (4999)	3981 (1.5119)	270 (36.037)	228 (42.8596)
90	10000	3507 (1.8514)	133 (74.188)	10000 (0)	8475 (0.1799)	161 (61.1118)	392 (24.5102)	2 (4999)	3515 (1.845)	173 (56.8035)	162 (60.7284)
99	10000	3737 (1.6759)	61 (162.9344)	10000 (0)	7507 (0.3321)	112 (88.2857)	193 (50.8135)	3 (3332.3333)	3078 (2.2489)	116 (85.2069)	102 (97.0392)
108	10000	3202 (2.123)	0 (*)	10000 (0)	6099 (0.6396)	74 (134.1351)	81 (122.4568)	1 (9999)	2990 (2.3445)	96 (103.1667)	77 (128.8701)
(+/-/-)		12/0/0	12/0/0	0/12/0	11/1/0	12/0/0	9/3/0	12/0/0	10/2/0	12/0/0	12/0/0
Min	10000	3202	0	10000	6099	74	81	1	2990	96	77
Max	10000	7554	7959	10000	10000	4427	10000	1718	10000	6474	6331
Avg	10000	4719	1989	10000	8978	1058	5217	208	6629	1650	1510

Table 26. Tabulated *RT* values for HBHA versus baseline algorithms. The bracket indicates the ratio of improvement, and the negative ratio implies the proposed HBHA outperform baseline algorithms. The bold values indicate that the proposed HBHA obtained the lowest *RT* values compared to other baseline algorithms.

<i>NN</i>	HBHA	BHA	GA	CSA	DEA	EA	ABCJ	ABCK	ABCS	SCA	WOA
9	0.104	0.185 (-0.436)	0.469 (-0.778)	0.315 (-0.669)	0.329 (-0.683)	0.323 (-0.677)	0.41 (-0.745)	1 (-0.896)	0.147 (-0.289)	0.331 (-0.685)	0.342 (-0.695)
18	0.114	0.179 (-0.366)	0.578 (-0.803)	0.322 (-0.647)	0.337 (-0.663)	0.336 (-0.662)	0.342 (-0.668)	1 (-0.886)	0.083 (0.372)	0.331 (-0.656)	0.349 (-0.674)
27	0.111	0.179 (-0.383)	0.622 (-0.822)	0.333 (-0.668)	0.342 (-0.676)	0.336 (-0.671)	0.325 (-0.659)	1 (-0.889)	0.09 (0.231)	0.331 (-0.665)	0.345 (-0.679)
36	0.095	0.139 (-0.318)	0.656 (-0.856)	0.334 (-0.716)	0.354 (-0.732)	0.335 (-0.717)	0.321 (-0.704)	1 (-0.905)	0.106 (-0.11)	0.329 (-0.712)	0.341 (-0.722)
45	0.089	0.106 (-0.159)	0.674 (-0.867)	0.338 (-0.736)	0.346 (-0.742)	0.334 (-0.732)	0.358 (-0.75)	1 (-0.911)	0.147 (-0.393)	0.331 (-0.73)	0.34 (-0.737)
54	0.085	0.1 (-0.149)	0.683 (-0.875)	0.346 (-0.754)	0.358 (-0.762)	0.333 (-0.744)	0.494 (-0.827)	1 (-0.915)	0.23 (-0.629)	0.331 (-0.743)	0.341 (-0.75)
63	0.085	0.096 (-0.112)	0.698 (-0.878)	0.349 (-0.755)	0.372 (-0.77)	0.334 (-0.744)	0.674 (-0.873)	1 (-0.915)	0.295 (-0.711)	0.33 (-0.741)	0.339 (-0.748)
72	0.081	0.094 (-0.135)	0.699 (-0.884)	0.357 (-0.773)	0.403 (-0.799)	0.333 (-0.756)	0.824 (-0.902)	1 (-0.919)	0.376 (-0.784)	0.33 (-0.755)	0.335 (-0.758)
81	0.081	0.092 (-0.124)	0.71 (-0.886)	0.363 (-0.778)	0.453 (-0.822)	0.334 (-0.758)	0.909 (-0.911)	1 (-0.919)	0.453 (-0.822)	0.33 (-0.756)	0.334 (-0.759)
90	0.078	0.091 (-0.144)	0.722 (-0.892)	0.371 (-0.79)	0.435 (-0.821)	0.333 (-0.766)	0.959 (-0.919)	1 (-0.922)	0.495 (-0.843)	0.331 (-0.765)	0.334 (-0.767)
99	0.075	0.092 (-0.183)	0.716 (-0.895)	0.378 (-0.801)	0.498 (-0.849)	0.334 (-0.774)	0.979 (-0.923)	1 (-0.925)	0.542 (-0.861)	0.33 (-0.772)	0.335 (-0.775)
108	0.076	0.09 (-0.165)	0.723 (-0.896)	0.386 (-0.805)	0.589 (-0.872)	0.333 (-0.773)	0.991 (-0.924)	1 (-0.924)	0.55 (-0.863)	0.33 (-0.771)	0.332 (-0.773)
(+/-/-)		12/0/0	12/0/0	12/0/0	12/0/0	12/0/0	12/0/0	12/0/0	10/2/0	12/0/0	12/0/0
Min	0.075	0.090	0.469	0.315	0.329	0.323	0.321	1.000	0.083	0.329	0.332
Avg	0.090	0.120	0.662	0.349	0.401	0.333	0.632	1.000	0.293	0.331	0.339
Max	0.114	0.185	0.723	0.386	0.589	0.336	0.991	1.000	0.550	0.331	0.349

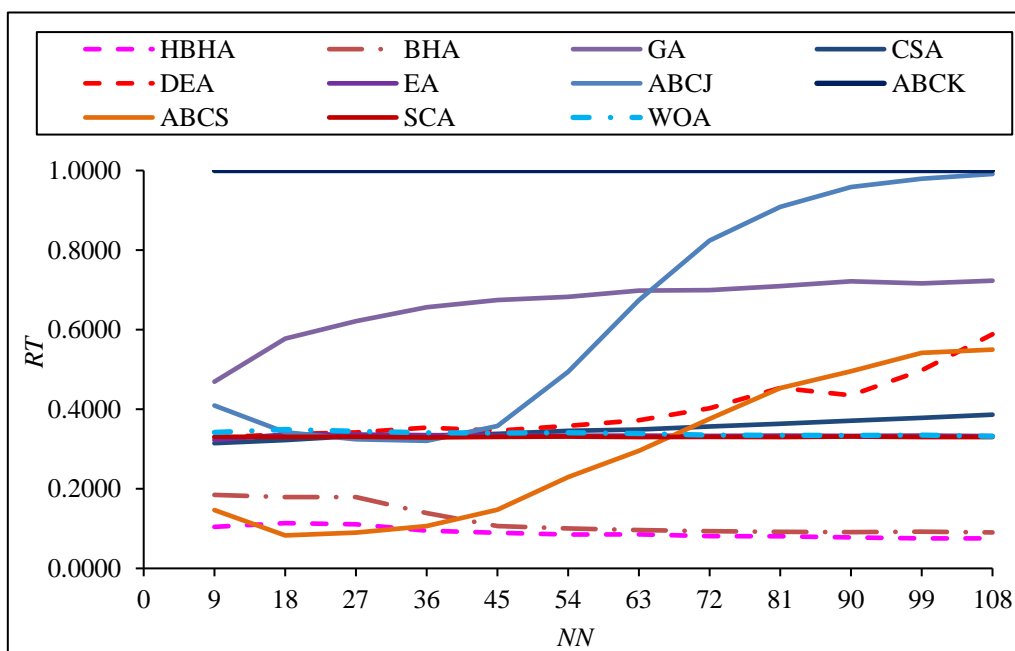


Figure 16. RT values of HBHA with baseline algorithm.

Table 27. Overall performance of HBHA and baseline algorithms.

Parameter	N_{DS}	N_{GR}	N_{DG}	RT
HBHA (Proposed model)	✓	✓	✓	✓
GA				
CSA	✓	✓	✓	
DEA		✓		
ABCJ		✓		
ABCK		✓		
ABCS		✓		
EA	✓			
SCA	✓			
WOA	✓			
BHA	✓			

The Friedman test results are presented in Table 28. Based on the results, the p -value for all performance metrics is less than 0.05 . Since $0.000 < 0.05$, we can reject the null hypothesis. Thus, it can be concluded that there is a significant difference in the performance metrics between the proposed HBHA and other baseline algorithms. Furthermore, Table 29 presents the mean rank of the algorithms across various performance metrics. In the context of maximizing the diversity and global solutions of the optimized final neuron state, higher values indicate the algorithms' capacity to generate maximum diversity and achieve global solutions. Notably, HBHA and CSA stand out with the highest mean rank

in this regard. Given their strong performance in maximizing these metrics, it is evident that both algorithms also exhibit the lowest mean rank in terms of error rate when seeking maximum diversified solutions and maximum attainment of global solutions. However, concerning the similarity index, HBHA secures the lowest rank of 1.17 compared to other baseline algorithms, underscoring its efficiency in producing optimized solutions with high dissimilarity compared to benchmark states.

Table 28. Friedman Test of various performance metrics.

Performance Metrics	p -value	Conclusion
N_{DS}	3.3421×10^{-13}	Reject H_0
N_{GR}	5.9879×10^{-21}	Reject H_0
N_{DG}	7.3346×10^{-19}	Reject H_0
RT	7.0067×10^{-17}	Reject H_0

Table 29. Mean rank values of HBHA and other baseline algorithms with different performance metrics.

	N_{DS}	N_{GR}	N_{DG}	RT
HBHA	9.71	8.50	10.25	1.17
BHA	7.50	4.00	6.58	2.33
GA	2.46	5.00	3.42	9.58
CSA	9.71	8.50	10.25	5.83
DEA	5.50	8.50	8.67	7.67
EA	6.54	1.17	2.67	5.08
ABCJ	5.04	8.50	7.29	8.50
ABCK	1.08	8.50	1.08	11.00
ABCS	4.83	8.50	7.71	4.33
SCA	7.08	3.00	4.67	4.08
WOA	6.54	1.83	3.42	6.42

8.3. Impact analysis of ℓ_{NR3SAT}

So far, the sections have demonstrated the overall performance of the proposed model focusing on the implementation of SAT in DHNN and the optimization of the retrieval phase of DHNN. This section discusses the impact of the proposed work on the overall performance of DHNN. Ideally, NR3SAT was formulated to address the limitation exists in the existing 3SAT. Both SAT are focusing on higher order systematic logic. The main difference between the existing 3SAT and the proposed NR3SAT lies in the appearance of negative literals within the clauses. In 3SAT, the appearance of positive and negative literals is random, often resulting in more positive literals compared to negative ones. In fact, this issue was supported by the finding of Abdeen et al. [36] which highlighted that the occurrence of negative literals for SAT is exceptionally infrequent. Neglecting the role of these negative literals not only limits the expressiveness of the logical structure but also inhibits the exploration of undiscovered solutions. Consequently, when this logic is embedded into DHNN, the network tends to produce repetitive final neuron states, leading to overfitting solutions. In other words,

there is a high similarity between the retrieved states and the benchmark states. However, controlling the appearance of negative literals within NR3SAT clauses has impacted the overall performance of DHNN. This is supported by the learning errors shown in Figure 5 and Figure 6. Controlling the appearance of negative literals in NR3SAT increases the chances of obtaining satisfied interpretation, resulting in more global solutions. Additionally, NR3SAT produces high dissimilarity between the retrieved states and the benchmark states, as shown in Figure 9. Therefore, the formulation of NR3SAT is significant for enhancing DHNN's performance in obtaining global solutions and avoiding overfitting.

Next, to ensure this intelligent model can classify and extract knowledge, optimizing the retrieval capabilities of DHNN is crucial. One might wonder why it is necessary to propose multi-objective functions in the retrieval phase. The answer lies in the potential use of this model in logic mining. In this context, the final neuron states retrieved will be converted in the form of induced logic [52]. This logic will have the capability to classify and extract knowledge. Unlike other classification approaches that only indicate the model's accuracy, this induced logic can extract patterns and relationships, explaining the reasoning behind the value obtained. Therefore, it is vital to ensure the network can retrieve high-quality induced logic to enhance interpretability in explaining these relationships. It is worth noting that existing works focus solely on obtaining global solutions but there is a high risk of all solutions being global but overfitted. How does this overfitting issue affect induced logic? Overfitting may cause the network to retrieve the same pattern of induced logic. In other words, this will reduce the interpretability issue. Therefore, it is essential to consider these multi-objective functions in the retrieval phase to ensure that the network can retrieve high quality final neuron states which will be beneficial in doing logic mining.

9. Conclusions

The primary objective of optimizing the DHNN model is to enhance its applicability in solving real-life classification problems. We introduce a new logical rule called NR3SAT to be embedded into the DHNN framework. The motivation behind proposing NR3SAT is to address the limitations of existing 3SAT models. The 3SAT model, which operates with randomized positive and negative activated neuron connections, raises concerns about the interpretability of DHNN. Moreover, the final neuron states retrieved by the 3SAT model tends to be overfitted due to the lack of variations compared to benchmark states in terms of negativity. The implementation of NR3SAT into DHNN results in improved performance. We found that this implementation leads to the lowest learning and testing errors and enables the model to attain more global minima solutions compared to existing baseline models. Additionally, the quality of the retrieved final neuron states is enhanced as evidenced by the lowest Jaccard similarity index values, indicating high dissimilarity compared to benchmark states.

In addressing the variation of negativity in the retrieved final neuron states, the study proposes a multi-objectives function during the optimization of the retrieval phase of DHNN. The goal is to achieve final neurons states with maximum diversity, maximum global solutions and the lowest RT values. To optimize the retrieval phase, the study introduces the Hybrid Black Hole Algorithm (HBHA). The results indicate that HBHA performs optimally when the degree of negativity in a solution is configured to achieve the desired proportion of clauses containing at least one negative state. The results also demonstrate that HBHA achieves maximum diversity for all numbers of neurons with all optimized solutions are global and attain the lowest similarity index.

The study opens opportunities for future research to explore the efficacy of the model in the

context of non-systematic logics, involving higher-order logical structures such as RAN3SAT [14], GRAN3SAT [41], and MAJ2SAT [15]. In the context of the proposed structure of SAT, researchers may explore the impact of controlling the appearance of negative literals in a clause without considering all positive literals in perspective of non-systematic logics [8,28]. Additionally, researchers may focus on achieving diversified solutions by considering the overall literals rather than clauses. Besides this, to assess the efficiency of this proposed model, researchers may extend this work from the perspective of logic mining in solving real-life classification problems. This is due to the limitation of the logic mining model, which does not consider on optimizing both learning and retrieval phase in enhancing the performance in addressing classification problems [29,47,53]. Additionally, from another perspective, the success of the proposed Hybrid Black Hole Algorithm in addressing multi-objective functions within the DHNN framework can be extended to other neural networks such as Cohen-Grossberg Neural Networks and to different problems such as parameter estimation within neural networks by making appropriate adjustments and modifications [54,55].

Author contributions

Nur ‘Afifah Rusdi: Conceptualization, Project Administration, Writing – Original Draft. Mohd. Asyraf Mansor: Funding Acquisition, Resources. Nur Ezlin Zamri: Visualization, Formal Analysis, Investigation. Mohd Shareduwan Mohd Kasihmuddin: Supervision. Nurul Atiqah Romli: Methodology. Gaeithry Manoharam: Writing – Review & Editing. Suad Abdeen: Validation. All authors have read and approve the final version of the manuscript for publication.

Acknowledgments

This research is financially supported by the Ministry of Higher Education Malaysia for the financial support of the Fundamental Research Grant Scheme (FRGS) with Project Code: FRGS/1/2022/STG06/USM/02/6 and Universiti Sains Malaysia. All the authors gratefully acknowledge the support.

Conflict of interest

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

References

1. M. Nakıp, E. Çakan, V. Rodoplu, C. Güzeliş, Dynamic automatic forecaster selection via artificial neural network based emulation to enable massive access for the Internet of Things, *J. Netw. Comput. Appl.*, **201** (2022), 103360. <https://doi.org/10.1016/j.jnca.2022.103360>
2. H. Azgomi, F. R. Haredasht, M. R. S. Motlagh, Diagnosis of some apple fruit diseases by using image processing and artificial neural network, *Food Control*, **145** (2023), 109484. <https://doi.org/10.1016/j.foodcont.2022.109484>
3. G. Dede, M. H. Sazlı, Speech recognition with artificial neural networks, *Digit. Signal Prog.*, **20** (2010), 763–768. <https://doi.org/10.1016/j.dsp.2009.10.004>

4. O. Surakhi, M. A. Zaidan, P. L. Fung, N. H. Motlagh, S. Serhan, M. AlKhanafseh, et al., Time-lag selection for time-series forecasting using neural network and heuristic algorithm, *Electronics*, **10** (2021), 2518. <https://doi.org/10.3390/electronics10202518>
5. G. D'Angelo, F. Palmieri, Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial–temporal features extraction, *J. Netw. Comput. Appl.*, **173** (2021), 102890. <https://doi.org/10.1016/j.jnca.2020.102890>
6. N. Ahad, J. Qadir, N. Ahsan, Neural networks in wireless networks: Techniques, applications and guidelines, *J. Netw. Comput. Appl.*, **68** (2016), 1–27. <https://doi.org/10.1016/j.jnca.2016.04.006>
7. J. J. Hopfield, D. W. Tank, “Neural” computation of decisions in optimization problems, *Biol. Cybern.*, **52** (1985), 141–152. <https://doi.org/10.1007/BF00339943>
8. 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. Softw.*, **171** (2022), 103169. <https://doi.org/10.1016/j.advengsoft.2022.103169>
9. C. Hu, Y. Ma, T. Chen, Application on online process learning evaluation based on optimal discrete hopfield neural network and entropy weight TOPSIS method, *Complexity*, **2021** (2021), 2857244. <https://doi.org/10.1155/2021/2857244>
10. L. Hu, F. Sun, H. Xu, H. Liu, X. Zhang, Mutation Hopfield neural network and its applications, *Inf. Sci.*, **181** (2011), 92–105. <https://doi.org/10.1016/j.ins.2010.08.007>
11. 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>
12. M. S. M. Kasihmuddin, M. A. Mansor, S. Sathasivam, Hybrid genetic algorithm in the hopfield network for logic satisfiability problem, *Pertanika J. Sci. Technol.*, **25** (2017). <https://doi.org/10.1063/1.4995911>
13. 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>
14. 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>
15. 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>
16. 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>
17. 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). <https://doi.org/10.9781/ijimai.2017.448>
18. L. C. Kho, M. S. M. Kasihmuddin, M. A. Mansor, S. Sathasivam, Propositional Satisfiability Logic via Ant Colony Optimization in Hopfield Neural Network, *Malays. J. Math. Sci.*, **16** (2022), 37–53. <https://doi.org/10.47836/mjms.16.1.04>
19. M. S. M. Kasihmuddin, M. A. Mansor, S. Sathasivam, Robust artificial bee colony in the hopfield network for 2-satisfiability problem, *Pertanika J. Sci. Technol.*, **25** (2017). <https://doi.org/10.1063/1.4995911>

20. M. A. Mansor, M. S. M. Kasihmuddin, S. Sathasivam, Grey wolf optimization algorithm with discrete hopfield neural network for 3 Satisfiability analysis, *J. Phys. Conf. Ser.*, **1821** (2021), 012038. <https://doi.org/10.1088/1742-6596/1821/1/012038>
21. 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>
22. A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, *Inf. Sci.*, **222** (2013), 175–184. <https://doi.org/10.1016/j.ins.2012.08.023>
23. W. Xie, J. S. Wang, C. Xing, S. S. Guo, M. W. Guo, L. F. Zhu, Extreme learning machine soft-sensor model with different activation functions on grinding process optimized by improved black hole algorithm, *IEEE Access*, **8** (2020), 25084–25110. <https://doi.org/10.1109/ACCESS.2020.2970429>
24. W. Gao, X. Wang, S. Dai, D. Chen, Study on stability of high embankment slope based on black hole algorithm, *Environ. Earth Sci.*, **75** (2016), 1–13. <https://doi.org/10.1007/s12665-016-6208-y>
25. M. K. Smail, H. R. E. H. Boucekara, L. Pichon, H. Boudjefdjouf, A. Amloune, Z. Lacheheb, Non-destructive diagnosis of wiring networks using time domain reflectometry and an improved black hole algorithm, *Nondestruct. Test. Eval.*, **32** (2017), 286–300. <https://doi.org/10.1080/10589759.2016.1200576>
26. 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>
27. J. L. Johnson, A neural network approach to the 3-satisfiability problem, *J. Parallel Distrib. Comput.*, **6** (1989), 435–449. [https://doi.org/10.1016/0743-7315\(89\)90068-3](https://doi.org/10.1016/0743-7315(89)90068-3)
28. M. A. F. Roslan, N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, Major 3 Satisfiability logic in Discrete Hopfield Neural Network integrated with multi-objective Election Algorithm, *AIMS Math.*, **8** (2023), 22447–22482. <https://doi.org/10.3934/math.20231145>
29. A. Alway, N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, S. Z. M. Jamaludin, M. F. Marsani, A novel hybrid exhaustive search and data preparation technique with multi-objective discrete hopfield neural network, *Decis. Anal.*, **9** (2023), 100354. <https://doi.org/10.1016/j.dajour.2023.100354>
30. F. S. Gharehchopogh, H. Shayanfar, H. Gholizadeh, A comprehensive survey on symbiotic organisms search algorithms, *Artif. Intell. Rev.*, **53** (2020), 2265–2312. <https://doi.org/10.1007/s10462-019-09733-4>
31. D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, *Appl. Math. Comput.*, **214** (2009), 108–132. <https://doi.org/10.1016/j.amc.2009.03.090>
32. 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>
33. V. Someetheram, M. F. Marsani, M. S. M. Kasihmuddin, N. E. Zamri, S. S. M. Sidik, S. Z. M. Jamaludin, et al., Random maximum 2 satisfiability logic in discrete hopfield neural network incorporating improved election algorithm, *Mathematics*, **10** (2022), 4734. <https://doi.org/10.3390/math10244734>
34. S. Sathasivam, M. A. Mansor, M. S. M. Kasihmuddin, H. Abubakar, Election algorithm for random k satisfiability in the hopfield neural network, *Processes*, **8** (2020), 568. <https://doi.org/10.3390/PR8050568>

35. 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>
36. S. Abdeen, M. S. M. Kasihmuddin, N. E. Zamri, G. Manoharam, M. A. Mansor, N. Alshehri, S-type random k satisfiability logic in discrete hopfield neural network using probability distribution: Performance optimization and analysis, *Mathematics*, **11** (2023), 984. <https://doi.org/10.3390/math11040984>
37. J. Chen, M. S. M. Kasihmuddin, Y. Gao, Y. Guo, M. A. Mansor, N. A. Romli, et al., PRO2SAT: Systematic probabilistic satisfiability logic in discrete hopfield neural network, *Adv. Eng. Softw.*, **175** (2023), 103355. <https://doi.org/10.1016/j.advengsoft.2022.103355>
38. S. Sathasivam, Upgrading logic programming in Hopfield network, *Sains Malays.*, **39** (2010), 115–118.
39. H. Abubakar, M. L. Danrimi, Hopfield type of artificial neural network via election algorithm as heuristic search method for random boolean ksatisfiability, *Int. J. comput. Digit. Syst.*, **10** (2021), 659–673. <https://doi.org/10.12785/ijcds/100163>
40. S. Mirjalili, A. Lewis, S-shaped versus V-shaped transfer functions for binary particle swarm optimization, *Swarm Evol. Comput.*, **9** (2013), 1–14. <https://doi.org/10.1016/j.swevo.2012.09.002>
41. 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>
42. N. Roslan, S. Sathasivam, F. L. Azizan, Conditional random k satisfiability modeling for k= 1, 2 (CRAN2SAT) with non-monotonic Smish activation function in discrete Hopfield neural network, *AIMS Math.*, **9** (2024), 3911–3956. <https://doi.org/10.3934/math.2024193>
43. M. M. Bazuhair, S. Z. M. Jamaludin, N. E. Zamri, M. S. M. Kasihmuddin, M. A. Mansor, A. Alway, et al., Novel Hopfield neural network model with election algorithm for random 3 satisfiability, *Processes*, **9** (2021), 1292. <https://doi.org/10.3390/pr9081292>
44. S. Taghian, M. H. Nadimi-Shahraki, Binary Sine Cosine Algorithms for Feature Selection from Medical Data, *Adv. Comput. Int. J.*, **10** (2019). <https://doi.org/10.5121/acij.2019.10501>
45. S. Chakraborty, A. K. Saha, S. Sharma, S. Mirjalili, R. Chakraborty, A novel enhanced whale optimization algorithm for global optimization, *Comput. Ind. Eng.*, **153** (2021), 107086. <https://doi.org/10.1016/j.cie.2020.107086>
46. A. G. Hussien, A. E. Hassanien, E. H. Houssein, M. Amin, A. T. Azar, New binary whale optimization algorithm for discrete optimization problems, *Eng. Optimiz.*, **52** (2020), 945–959. <https://doi.org/10.1080/0305215X.2019.1624740>
47. N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, A. Alway, S. Z. M. Jamaludin, S. A. Alzaemi, Amazon employees resources access data extraction via clonal selection algorithm and logic mining approach, *Entropy*, **22** (2020), 596. <https://doi.org/10.3390/E22060596>
48. L. Wang, X. Fu, Y. Mao, M. I. Menhas, M. Fei, A novel modified binary differential evolution algorithm and its applications, *Neurocomputing*, **98** (2012), 55–75. <https://doi.org/10.1016/j.neucom.2011.11.033>
49. D. Jia, X. Duan, M. K. Khan, Binary Artificial Bee Colony optimization using bitwise operation, *Comput. Ind. Eng.*, **76** (2014), 360–365. <https://doi.org/10.1016/j.cie.2014.08.016>
50. 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>

51. S. A. Karim, M. S. M. Kasihmuddin, S. Sathasivam, M. A. Mansor, S. Z. M. Jamaludin, M. R. Amin, A novel multi-objective hybrid election algorithm for higher-order random satisfiability in discrete hopfield neural network, *Mathematics*, **10** (2022), 1963. <https://doi.org/10.3390/math10121963>
52. 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.–Com.*, **35** (2023), 101554. <https://doi.org/10.1016/j.jksuci.2023.101554>
53. M. S. M. Kasihmuddin, S. Z. M. Jamaludin, M. A. Mansor, H. A. Wahab, S. M. S. Ghadzi, Supervised learning perspective in logic mining, *Mathematics*, **10** (2022), 915. <https://doi.org/10.3390/math10060915>
54. C. Wang, H. Zhang, D. Wen, M. Shen, L. Li, Z. Zhang, Novel passivity and dissipativity criteria for discrete-time fractional generalized delayed Cohen–Grossberg neural networks, *Commun. Nonlinear Sci. Numer. Simul.*, **133** (2024), 107960. <https://doi.org/10.1016/j.cnsns.2024.107960>
55. X. Wang, H. R. Karimi, M. Shen, D. Liu, L. W. Li, J. Shi, Neural network-based event-triggered data-driven control of disturbed nonlinear systems with quantized input, *Neural Netw.*, **156** (2022), 152–159. <https://doi.org/10.1016/j.neunet.2022.09.021>



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 (<https://creativecommons.org/licenses/by/4.0>)