mathematics

Article

GRAN3SAT: Creating Flexible Higher-Order Logic Satisfiability in the Discrete Hopfield Neural Network

Yuan Gao [1,2], Yueling Guo [1,3], Nurul Atiqah Romli [1], Mohd Shareduwan Mohd Kasihmuddin [1,*], Weixiang Chen [2], Mohd. Asyraf Mansor [4] and Ju Chen [1,2]

[1] School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia; gaoyuan@student.usm.my (Y.G.); guoyueling1982@163.com (Y.G.); nurulatiqah_@student.usm.my (N.A.R.); chenju@cdutcm.edu.cn (J.C.)
[2] School of Medical Information Engineering, Chengdu University of Traditional Chinese Medicine, Chengdu 610000, China; weixiang@cdutcm.edu.cn
[3] School of Science, Hunan Institute of Technology, Hengyang 421002, China
[4] School of Distance Education, Universiti Sains Malaysia, Penang 11800, Malaysia; asyrafman@usm.my
[*] Correspondence: shareduwan@usm.my; Tel.: +60-4-6534769

Abstract: One of the main problems in representing information in the form of nonsystematic logic is the lack of flexibility, which leads to potential overfitting. Although nonsystematic logic improves the representation of the conventional *k* Satisfiability, the formulations of the first, second, and third-order logical structures are very predictable. This paper proposed a novel higher-order logical structure, named G-Type Random *k* Satisfiability, by capitalizing the new random feature of the first, second, and third-order clauses. The proposed logic was implemented into the Discrete Hopfield Neural Network as a symbolic logical rule. The proposed logic in Discrete Hopfield Neural Networks was evaluated using different parameter settings, such as different orders of clauses, different proportions between positive and negative literals, relaxation, and differing numbers of learning trials. Each evaluation utilized various performance metrics, such as learning error, testing error, weight error, energy analysis, and similarity analysis. In addition, the flexibility of the proposed logic was compared with current state-of-the-art logic rules. Based on the simulation, the proposed logic was reported to be more flexible, and produced higher solution diversity.

Keywords: G-Type Random *k* Satisfiability; artificial neural network; Hopfield Neural Network; flexibility; random dynamics

MSC: 68T07; 68T27; 68T20

check for updates

Citation: Gao, Y.; Guo, Y.; Romli, N.A.; Kasihmuddin, M.S.M.; Chen, W.; Mansor, M.A.; Chen, J. GRAN3SAT: Creating Flexible Higher-Order Logic Satisfiability in the Discrete Hopfield Neural Network. *Mathematics* **2022**, *10*, 1899. https://doi.org/10.3390/math10111899

Academic Editor: Bogdan Oancea

Received: 14 March 2022
Accepted: 13 April 2022
Published: 1 June 2022

1. Introduction

Artificial Intelligence (AI) is a field of modelling intelligence that integrates technical science, theory development, mathematics, computer science, physics, and biology. AI has many applications [1–5], which include Artificial Neural Networks (ANN). The conventional ANN consists of interconnected neurons that divide input and output layers which are connected by synaptic weight. Generally, the input neuron receives information in the form of a problem or data, is processed by the intermediate layer, and generates the final neuron state that corresponds to the solution of the problem. This feature makes ANNs a great platform to solve and improve the solution of any given optimization problem.

In 1982, Hopfield [6] proposed the earliest variant of ANN, namely the Hopfield Neural Network (HNN), that consists of a single-layer feedback neural network. In this discussion, we only consider the application of the Discrete Hopfield Neural Network (DHNN) in solving the optimization problem. The DHNN is a two-value nonlinear dynamic system that has multiple inputs, and the firing of the output is solely based on the pre-defined threshold values. Structurally, the sufficient condition for the stability of the

DHNN is that the weighted coefficient matrix is symmetric and has zero diagonal elements. By capitalizing the coefficient matrix of the DHNN, the solution capacity creates associative memory behavior which mimics actual human intelligence. The final neuron state of the DHNN can be interpreted in terms of an energy function, where the absolute minimum energy signifies the most optimal solution for any given optimization problem. Thus, the correctness of the final energy function in a DHNN is highly dependent on the value of the synaptic weight assigned by the network. One of the main weaknesses of the conventional DHNN is the convergence issue resulting from the lack of capacity as the number of neurons increases. This is because the conventional DHNN has no symbolic rule to govern the modeling of the Discrete Hopfield Network (DHN), which causes the network to reiterate the synaptic weight until the optimal synaptic weight can be reached. Without a proper symbolic rule, the DHNN will forever iterate until the final neuron state achieves an optimal state that corresponds to global minimum energy. In order to remedy the situation, Abdullah [7] proposed a logical rule in ANN by mapping the connection of the neuron with a valid (or near valid) interpretation. This is the earliest effort to introduce an effective method to find the optimal synaptic weight that corresponds to the optimal final neuron state. Interestingly, this is the first proposal of the term “Wan Abdullah method” where the synaptic weight is obtained by comparing the final energy with the cost function of the logic. This research direction was continued by Sathasivam [8], where she proposed the Horn Satisfiability (HornSAT) logical rule in DHNNs. This proposed DHNN utilizes effective neuron relaxation to ensure the final neuron state will not be trapped in the suboptimal state. Note that this study was the first computer simulation of logic programming in DHNNs, and the result shows that logical rules indeed can be embedded into DHNNs. However, the impact of different logical rules in DHNNs is poorly understood because HornSAT has limited usability in terms of structure. Thus, there is a need for a different structure of logical rules where each variable inside the clause is not limited to only one positive literal at most.

Kasihmuddin et al. [9] proposed the first systematic logical rule, namely 2 Satisfiability (2SAT) in DHNNs. The proposed logical rule has two literals per clause, and all clauses are connected by disjunction. This logical rule was embedded into DHNN by comparing the cost function with the Lyapunov energy function. With the aid of a genetic algorithm, the learning of 2SAT in DHNNs can be carried out effectively. Mansor et al. [10] extended the order of the logical rule by proposing 3 Satisfiability in DHNNs. In this context, the third-order Lyapunov energy function is compared to obtain the third-order synaptic weight. The proposed study managed to obtain an optimal value for the global minima ratio, despite the associative memory of the DHNN with 3SAT increasing exponentially. This research was pivotal to the application of systematic SAT in ANNs. Alzaeemi et al. [11] proposed 2SAT in the Radial Basis Function Neural Network (RBFNN) by calculating the center and width that corresponds to the output weight. The implementation of 2SAT in an RBFNN was reported to yield a small iteration error during learning. Note that the proposed study has been comprehensively compared with the state-of-the-art DHNN in [12], showing the compatibility of systematic logical rules in various types of ANNs. In another development, Kasihmuddin et al. [13] proposed the first non-satisfiability logical rule, namely Maximum k Satisfiability in DHNN, by considering the nonzero cost function during the learning phase. The proposed research was shown to achieve an optimal global minima ratio with lower learning error. In another development, systematic logic has been applied to logic mining that classifies various real-life problems. Despite successful implementation of systematic logic in DHNNs, systematic logic lacks variety of clauses and produces less neuron variation during the retrieval phase of DHNNs. Thus, there is an urgent proposal for a logical rule that contains a clause with different orders to be embedded into a DHNN.

There is a great diversity of nonsystematic logical rules that were recently proposed. Sathasivam et al. [14] proposed the first Random 2 Satisfiability (RAN2SAT) in DHNNs, where the first and second-order clauses form the whole logical formulation. Interestingly, the result of the experiment shows that the first-order clause creates more logical inconsistency compared to the second-order clause. This implies that as the number of first-order clauses increases the DHNN is unable to complete the learning phase, resulting in a suboptimal retrieval phase. This research was further extended by Karim et al. [15], where the higher-order RAN*k*SAT was proposed by adding third-order logic. This study has an interesting insight because different variants of RAN3SAT, such as RAN1,3SAT and RAN2,3SAT were proposed in DHNNs. In this paper, all the variants of RAN3SAT were compared with the systematic logical rule to obtain the optimal global minima ratio. This is an interesting result because we are able to validate that DHNNs can "behave" according to the nonsystematic logical rule. In order to optimize RAN3SAT in DHNNs, Bazuhair et al. [16] intelligently proposed Election Algorithm to optimize and improve both the learning and retrieval phases. The proposed RAN3SAT is considered the best hybrid DHNN because the proposed method has low learning error, high variation value, and a high global minima ratio. In another development, Alway et al. [17] contributed to the development of nonsystematic logical rules by proposing Major 2 Satisfiability (MAJ2SAT). The proposed logical rule capitalizes a high proportion of second-order logic in comparison with other logical clauses. Based on the result reported in this paper, MAJ2SAT creates more variation in terms of logical rules, and has a very low similarity value compared to systematic logical rules. However, the existing nonsystematic logical rule does not take into account the random occurrence of the clause that makes the final formulation. In this context, nonsystematic logic must have the ability to cover all the solution sets bounded by the higher-order logical clause.

In this paper, we introduced G-Type Random 3 satisfiability (GRAN*k*SAT) that capitalizes both higher-order systematic and nonsystematic logical rules in DHNNs. The higher-order systematic logical rule provides storage capacity to GRAN3SAT, whereas the higher-order nonsystematic logical rule provides a more diversified third-order logical connection. This is the first attempt to leverage both logical rules into a DHNN which we believe can represent all sets of logical rules that have been previously proposed. The main contributions of this paper are as follows:

1. We propose a novel logical rule, namely G-Type Random 3 Satisfiability, or GRAN3SAT, by randomly generating the first-order, second-order, and third-order satisfiability logical rules. By incorporating a third-order clause, the capacity of the proposed logic increases.
2. We implement GRAN3SAT into a DHNN by minimizing the logical inconsistency of the logical rule that corresponds to the zero-cost function. The derived cost function that corresponds to GRAN3SAT will be capitalized to compute the synaptic weight of the network.
3. We conduct various extensive analyses to examine the behavior of the proposed GRAN3SAT. The final neuron state for various case studies will be evaluated based on different initial neuron states, parameter perturbation, different trial runs, and relaxation. Various performance metrics, such as learning error, synaptic weight error, energy profile, test error, and similarity metric, will be reported to justify the behavior of the proposed GRAN3SAT.
4. We compare the proposed GRAN3SAT with state-of-the-art systematic and nonsystematic logical rules.

The organization of this paper is as follows. Section 2 provides an overview of the structure of a novel GRAN*k*SAT. Section 3 explains the implementation of GRAN*k*SAT into a DHNN. The experimental setup and performance evaluation metrics used throughout the simulation are shown in Section 4. Section 5 discusses and analyzes the behavior performance of a DHNN-GRAN*k*SAT in different parameters and phases, and compares it with several established logical structures. Finally, Section 6 presents the conclusions and future work.

2. G-Type Random *k* Satisfiability

GRAN*k*SAT is a nonsystematic logical structure expressed by conjunctive normal form CNF. GRAN*k*SAT consists of a series of clauses with random literals, and the numbers of clauses and states of literals are randomly determined. In this case, GRAN3SAT mainly consists of k-SAT ($k \leq 3$), where k-SAT is made up of a set of x literals and a set of y clauses. Each literal value has the form of $\{1, -1\}$ that represents TRUE or FALSE. The general structure of GRAN3SAT ($P_{GRAN3SAT}$) is given as follows:

(a) A set of x literals: $A_1, A_2, A_3, \ldots, A_x$

(b) A collection of clause numbers: $U = \{Nc_1, Nc_2, Nc_3 \ldots, Nc_\omega\}$, whereby

$$Nc_i = \begin{bmatrix} m_i & n_i & k_i \end{bmatrix}^T \cdot i \in [1, \omega] \tag{1}$$

$$x = 3m_i + 2n_i + k_i \tag{2}$$

$$y = m_i + n_i + k_i \tag{3}$$

where m_i is the number of the third-order clause, n_i is the number of the second-order clause, and k_i is the number of the first-order clause.

(c) A random number j, where $j \in [1, \omega]$ and $j \in N$ which corresponds to the set of clauses Nc_j.

(d) The third-order logic clause is as follows: $C_1^{(3)}, C_2^{(3)}, C_3^{(3)}, \ldots, C_{m_j}^{(3)}$, where

$$C_{m_j}^{(3)} = \left(A_{O_3} \vee A_{p_3} \vee A_{q_3}\right), O_3, p_3, q_3 \in N \tag{4}$$

(e) The second-order logic clause is defined as: $C_1^{(2)}, C_2^{(2)}, C_3^{(2)}, \ldots, C_{n_j}^{(2)}$, where

$$C_{n_j}^{(2)} = \left(A_{O_2} \vee A_{p_2}\right), O_2, p_2 \in N \tag{5}$$

(f) The first-order logic clause is stated as: $C_1^{(1)}, C_2^{(1)}, C_3^{(1)}, \ldots, C_{k_j}^{(1)}$, where

$$C_{k_j}^{(1)} = A_{O_1}, O_1 \in N \tag{6}$$

Thus, the general formulation for GRAN*k*SAT, or $P_{GRAN3SAT}$, based on the above features is as follows:

$$P_{GRAN3SAT} = \wedge_{i=1}^{m_j} C_i^{(3)} \wedge_{i=1}^{n_j} C_i^{(2)} \wedge_{i=1}^{k_j} C_i^{(1)} \tag{7}$$

where $m_j > 0$, $n_j \geq 0$, $k_j \geq 0$. Based on (4)–(6), the states of the literals are determined randomly, where $A_i \in \{A_i, \neg A_i\}$. The examples of $P_{GRAN3SAT}$ with different random structures are as follows:

$$P_{GRAN3SAT} = (A_1 \vee A_2 \vee A_3) \wedge (A_4 \vee A_5 \vee A_6) \wedge (A_7 \vee A_8) \wedge A_9 \tag{8}$$

$$P_{GRAN3SAT} = (A_1 \vee A_2 \vee A_3) \wedge (A_4 \vee A_5) \wedge (A_6 \vee A_7) \wedge A_8 \wedge A_9 \tag{9}$$

According to Equations (8) and (9), the equation is satisfied or $P_{GRAN3SAT} = 1$ if all the clauses in the formulation are fully satisfied. Another interesting point about Equation (7) is the randomness in representing the clause for $P_{GRAN3SAT}$. In this case, the clause is not only limited to Equations (8) and (9), but it has infinitely many combinations with a fixed total number of literals. Equation (7) is different from that of previous research proposed by Karim et al. [15], which proposed RAN3SAT formulation. In RAN3SAT, the proportion of the Equations (4)–(6) is pre-determined, although the state of the literal remains random. Thus, the random feature of the formulation does not consider the proportion of the clause. In this paper, we propose a higher-order logical rule of $P_{GRAN3SAT}$ by proposing a third-order clause (refer Equation (4)), yet the occurrence for each clause remains random. In other words, $P_{GRAN3SAT}$ is expected to provide more logical flexibility in terms of clauses and literals. In this case, any information in a combinatorial problem (such as logic mining) will be represented randomly in the form of a one-dimensional to three-dimensional system.

This feature helps practitioners represent any combinatorial problem in a more flexible formulation. Therefore, the proposed $P_{GRAN3SAT}$ is a breakthrough in modelling neurons in ANN.

3. GRAN3SAT in the Discrete Hopfield Neural Network

The DHNN is another variant of ANNs that has no hidden layers [18]; it can be used to solve various optimization problems. A DHNN consists of bipolar neurons where the state is represented by $\{1, -1\}$. The conventional neuron update with a pre-defined tolerance is as follows:

$$S_i = \begin{cases} 1, & \sum_{jk}^{n} W_{ijk} S_j S_k \geq \delta_i \\ -1, & \sum_{jk}^{n} W_{ijk} S_j S_k < \delta_i \end{cases} \tag{10}$$

where W_{ijk} is the synaptic weight between neuron i, neuron j, and neuron k. S_i is the state of neuron i, and δ_i is the threshold. It is the synaptic weight W_i that deserves attention, as it refers to the degree of the connection between multiple neurons. The property of the synaptic weights for the two neurons follows $W_{ij} = W_{ji}$, $W_{ijk} = W_{jik} = W_{jki} = W_{ikj} = W_{kij} = W_{kji}$, and has no self-feedback connection $W_{ii} = W_{jj} = 0$, $W_{iii} = W_{jjj} = W_{kkk} = 0$. When a higher-order connection has been added to Equation (10), any two similar neuron connections will result in zero value for synaptic weight. $P_{GRAN3SAT}$ can be implemented into the DHNN (GRAN3SAT) by assigning each neuron with a variable. Collectively, the neurons will be grouped randomly as clauses in Equations (4)–(6) until they satisfy the total number of neurons. The cost function $\chi_{P_{GRAN3SAT}}$ for the implementation of $P_{GRAN3SAT}$ into the DHNN is as follows:

$$\chi_{P_{GRAN3SAT}} = \frac{1}{8}\sum_{j=1}^{m_i} F_{j1}^{(3)} F_{j2}^{(3)} F_{j3}^{(3)} + \frac{1}{4}\sum_{j=1}^{n_i} F_{j1}^{(2)} F_{j2}^{(2)} + \frac{1}{2}\sum_{j=1}^{k_i} F_{j1}^{(1)} \tag{11}$$

$$F_j^{(\ k=1,2,3)} = \begin{cases} 1 + S_{A_j}, & if A_j \\ 1 - S_{A_j}, & if \neg A_j \end{cases} \tag{12}$$

To fully implement the modelling of $P_{GRAN3SAT}$ in the DHNN, the cost function $\chi_{P_{GRAN3SAT}}$ that is associated with $P_{GRAN3SAT}$ must be zero. In other words, the DHNN must find at least one interpretation which corresponds to a zero cost function. By finding at least one consistent interpretation, the optimal synaptic weight for GRAN3SAT via [19] can be found. However, if $\chi_{P_{GRAN3SAT}} \neq 0$, the $P_{GRAN3SAT}$ is not considered satisfiable, which results in nearly random synaptic weight. Since achieving $\chi_{P_{GRAN3SAT}} = 0$ is vital to ensure the DHNN can retrieve the correct final neuron state, effective learning methods must be employed during the learning phase of the DHNN. The probability of finding a satisfactory interpretation for $P_{GRAN3SAT}$ is as follows:

$$\theta(\chi_{P_{GRAN3SAT}} = 0) = \left(\frac{7}{8}\right)^{m_i} \left(\frac{3}{4}\right)^{n_i} \left(\frac{1}{2}\right)^{k_i} \tag{13}$$

where θ is the probability value, and $\left(1 - \frac{1}{2^k}\right)$ is the probability of satisfying the k-order clause. During the retrieval phase, the DHNN will implement the iterative update of neurons from the initial state to the final state via the local field formula and activation function. Equations (14) and (15) represent the local field formula and the formula of update neuron states, respectively. Since Hyperbolic Tangent Activation Function (HTAF) has non-linear properties, HTAF is widely used as the activation function in artificial neural networks [20].

$$h_i = \sum_{k \neq i,j}^{n} \sum_{j \neq i}^{n} W_{ijk} S_j S_k + \sum_{j \neq i}^{n} W_{ij} S_j + W_i \tag{14}$$

$$S_{f_i} = \begin{cases} 1, & \sum\limits_{k \neq i,j}^{n} \sum\limits_{j \neq i}^{n} W_{ijk} S_j S_k + \sum\limits_{j \neq i}^{n} W_{ij} S_j + W_i \geq 0 \\ -1, & \sum\limits_{k \neq i,j}^{n} \sum\limits_{j \neq i}^{n} W_{ijk} S_j S_k + \sum\limits_{j \neq i}^{n} W_{ij} S_j + W_{ii} < 0 \end{cases} \tag{15}$$

S_i and S_{f_i} represent the initial state and the updated state of neuron i, respectively. W_{ijk}, W_{ij}, and W_i represent the synaptic weights of the third, second, and first-orders of the DHNN, respectively. The main motivation of using $P_{GRAN3SAT}$ in GRAN3SAT is to obtain a more final state that has various logical rules during the retrieval phase. For instance, by using Equation (15), the final neuron state is connected in various types of clauses that have been stated in Equations (4)–(6). Thus, the magnitude of the final neuron state can be evaluated using the Lyapunov energy function $L_{P_{GRAN3SAT}}$, as shown as follows:

$$L_{P_{GRAN3SAT}} = -\frac{1}{3} \sum_{i}^{n} \sum_{j \neq i}^{n} \sum_{k \neq i,j}^{n} W_{ijk} S_i S_j S_k - \frac{1}{2} \sum_{i}^{n} \sum_{j \neq i}^{n} W_{ij} S_i S_j - \sum_{i}^{n} W_i S_i \tag{16}$$

$$L^{min}_{P_{GRAN3SAT}} = \frac{m_i}{8} + \frac{n_i}{4} + \frac{k_i}{2} \tag{17}$$

Since each logical order provides a fixed energy value, we can obtain the absolute minimum energy of the $P_{GRAN3SAT}$ by calculating $L^{min}_{P_{GRAN3SAT}}$ in Equation (17). Note that we can obtain the optimal synaptic weight by comparing Equation (16) with Equation (11), as long as the learning phase of GRAN3SAT obtained at least one interpretation that corresponds to $\chi_{P_{GRAN3SAT}} = 0$. By iteratively updating the neuron state via Equations (14) and (15), $L_{P_{GRAN3SAT}}$ always converges to the nearest local minima solution. As a result of the random nature of the proposed $P_{GRAN3SAT}$, both $L_{P_{GRAN3SAT}}$ and $L^{min}_{P_{GRAN3SAT}}$ will fluctuate and have different proportions than the research of Karim et al. [15], where the absolute final energy can be pre-determined. Despite having different clause arrangement compared to [15], the choice of literal in $P_{GRAN3SAT}$ is similar with the research of [14] and [15] where random literals are the building blocks of the clause in Equations (4)–(6). In order to separate between the global minimum solution and the local minimum solution, the convergence property of the proposed GRAN3SAT must satisfy the following condition:

$$\left| L_{P_{GRAN3SAT}} - L^{min}_{P_{GRAN3SAT}} \right| \leq Tol \tag{18}$$

where Tol is a pre-determined tolerance value of GRAN3SAT. In this context, condition (18) determines whether the final neuron state exhibits the behavior that satisfies $P_{GRAN3SAT}$.

Figure 1 illustrates the schematic diagram of the implementation of $P_{GRAN3SAT}$ to the DHNN (GRAN3SAT). Generally, the schematic diagram can be divided into the learning phase and the retrieval phase. Before the learning phase, random clause arrangement for $P_{GRAN3SAT}$ was determined and was converted into Boolean algebra. After assigning each clause in $P_{GRAN3SAT}$ with a neuron, GRAN3SAT is required to assign the neuron state that satisfies the cost function in Equation (11). Thus, the optimal neuron assignment will help us compute the optimal synaptic weight that will be used in the retrieval phase.

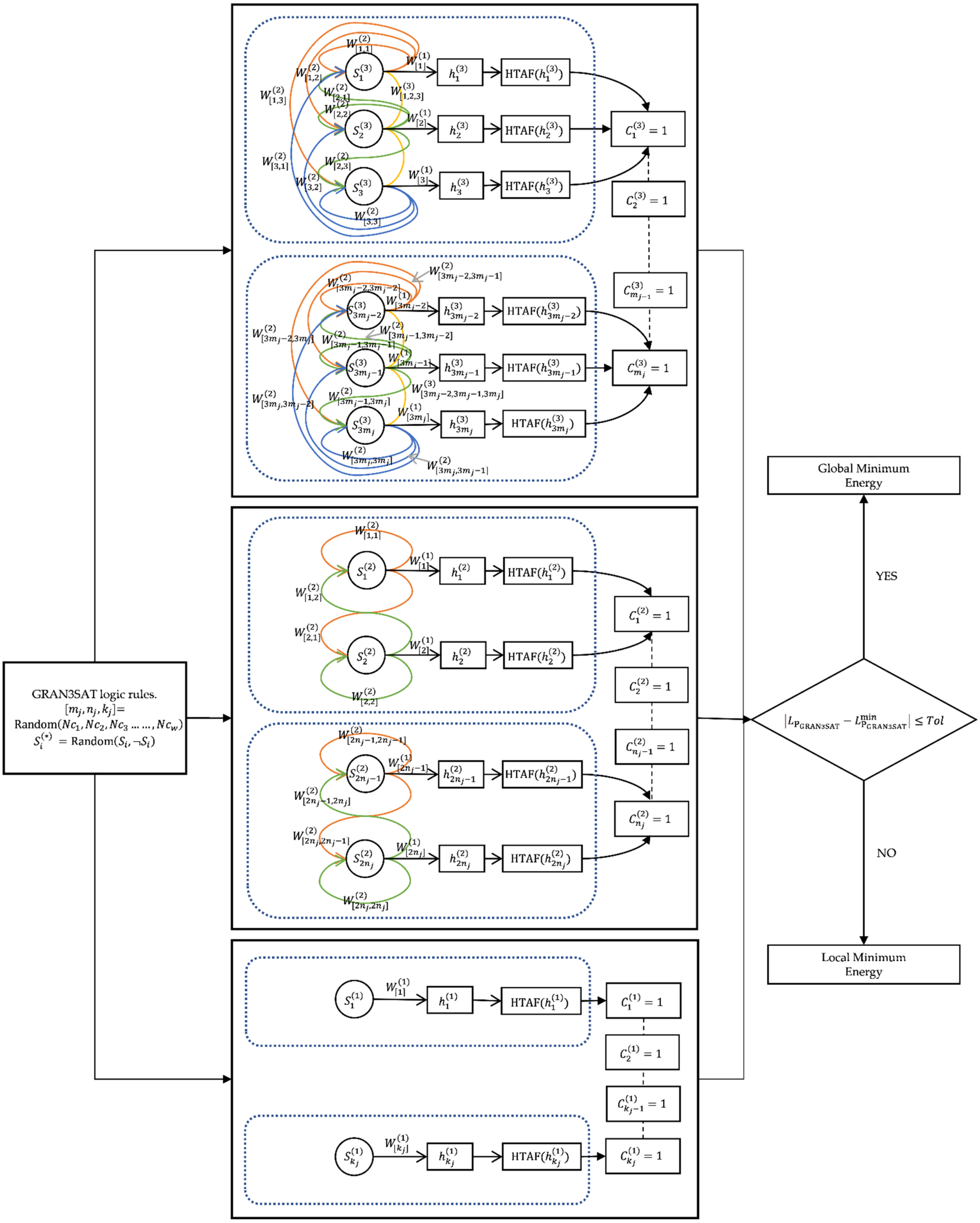


Figure 1. Schematic diagram for GRAN3SAT.

4. Experimental Setup

To further investigate the behavior of the proposed research, GRAN3SAT will be evaluated based on different parameters and learning settings. Four different simulations, including different number of clauses, different proportions of literals (positive or negative), different learning trials, and different numbers of iterations will be tested in this paper. Each part will undergo three assessments, which are the learning phase, retrieval phase, and similarity index. The details for each simulation are as follows:

- Different numbers of clauses. In this section, we evaluate and analyze the impact of different order logics on GRAN3SAT by using performance metrics at each phase, and determine the impact of parameter perturbation on GRAN3SAT.
- Different proportions of literals. In this section, we evaluate the impact of different proportions of literals (positive or negative) on GRAN3SAT by using performance metrics at each phase, and determine the impact of parameter perturbation on GRAN3SAT.
- Different learning trials. In this section, we evaluate the impact of different learning trials on GRAN3SAT on the performance metrics of each phase. This simulation provides a basis for efficiency improvement in the subsequent learning algorithm.
- Different numbers of iterations. In this section, we evaluate the impact of Sathasivam relaxation on GRAN3SAT via performance metrics of each phase to obtain the most optimal parameters.
- Flexibility analysis of the logic structure. In this section, we compare GRAN3SAT with several established logical rules in terms of flexibility of the logical rule.

All the experiments will be simulated using MATLAB 2016a with the 64-bit Windows 10 operating system. Table 1 shows the parameters involved in each experiment.

Table 1. Parameters for the proposed GRAN3SAT.

Parameter Explanation	Parameter Value
Number of neurons (NN)	$6 \leq NN \leq 150$
Number of clauses (NC)	$\{Nc_1, Nc_2, Nc_3 \ldots, Nc_\omega\}$
Number of neuron combinations $(N_{combmax})$	100
Number of learning trials (N_{trial})	100, 1000, 1000
Current number of learning trials (N_l)	$N_l \leq N_{trial}$
Initialization of neuron states in the learning phase	Random
Threshold (δ_i)	0 [9]
Relaxation rate (r)	$1 \leq r \leq 5$
Number of testing trials (N_t)	100
Initialization of neuron states in the testing phase	Random
Tolerance value (Tol)	0.001
Activation function	HTAF [20]
Type of selection	Random search

Each simulation will be evaluated using six types of performance metrics. The metrics are based on process evaluation in the learning phase (learning error analysis), outcome evaluation in the learning phase (weight analysis), process evaluation in the retrieval phase (energy analysis), outcome evaluation in the retrieval phase (global solution analysis), similarity index evaluation (solution diversity analysis), and structural error metrics (logic flexibility analysis). Tables 2–5 present a list of parameters involved in all performance evaluation metrics.

Table 2. List of parameters in the learning phase.

Parameter	Remarks
f_{NC}	Maximum fitness achieved
f_i	Current fitness achieved
W_{WA}	Synaptic weight obtained by Wan method
W_i	Current synaptic weight
N_w	Number of weights at a time
N_{w_c}	$N_{w_c} = N_w \cdot N_{combmax}$

Table 3. List of parameters in the retrieval phase.

Parameter	Remarks
L_{min}	Minimum energy value
L_f	Final energy
N_{GLO}	Number of global minimum solutions
N_{LOC}	Number of local minimum solutions
N_t	Number of testing trials
N_{t_c}	$N_{t_c} = N_t \cdot N_{combmax}$

Table 4. Parameters involved in the similarity index.

Parameter	Remarks
l	Number of $\{S_i^{ideal} = 1, S_{f_i} = 1\}$
m	Number of $\{S_i^{ideal} = 1, S_{f_i} = -1\}$
n	Number of $\{S_i^{ideal} = -1, S_{f_i} = 1\}$

Table 5. Parameters involved in structure evaluation.

Parameter	Remarks
N_{NC}	Current number of clauses
N_{mean_c}	Average number of clauses
$N_{negative}$	Current number of negative literals
N_{mean_p}	Average number of negative literals

RMSE (root mean square error), MAE (mean absolute error) [21], and MAPE (mean absolute percent error) [22] are statistical metrics that can be used as evaluation metrics for machine learning [23]. RMSE has been used as a standard statistical metric to measure the performance of models. In addition, MAE is one of the most direct measures of prediction error; the smaller the MAE value, the better the model. MAPE measures the accuracy of the proposed model by percentage value. Compared with MAE, RMSE is more sensitive to outliers which have a greater impact on it.

In the learning phase, we measure the fitness of neuron states and examine the satisfied clause which generates the optimal synaptic weights. Equations (19)–(21) will be used to measure the fitness of the neuron, whereas the error in synaptic weight used will be evaluated based on Equations (22) and (23). Table 2 shows the parameters used in synaptic weight analysis.

$$MAE_{learn} = \sum_{i=1}^{N_l} \frac{|f_{NC} - f_i|}{N_l} \tag{19}$$

$$RMSE_{learn} = \sum_{i=1}^{N_l} \sqrt{\frac{(f_{NC} - f_i)^2}{N_l}} \tag{20}$$

$$MAPE_{learn} = \sum_{i=1}^{N_l} \frac{1}{N_c} \frac{|f_{NC} - f_i|}{f_{NC}} \tag{21}$$

$$MAE_{weight} = \frac{\sum_{i=1}^{N_{w_c}} |W_{WA} - W_i|}{N_{w_c}} \tag{22}$$

$$RMSE_{weight} = \sqrt{\frac{\sum_{i=1}^{N_{w_c}} (W_{WA} - W_i)^2}{N_{w_c}}} \tag{23}$$

In the retrieval phase, the energy analysis is used to determine the efficiency of GRAN3SAT [24]. Equations (24) and (25) represents the formulation for RMSE and MAE of the neuron during retrieval phase. The quality of GRAN3SAT solutions is evaluated, and the RMSE, MAE, and ZM formulas are used to evaluate the test errors that are defined in Equations (26)–(28). In addition, Table 3 describes the parameters used in the retrieval phase.

$$MAE_{energy} = \frac{\sum_{i=1}^{N_{t_c}} \left| L_{min} - L_f \right|}{N_{t_c}} \tag{24}$$

$$RMSE_{energy} = \sqrt{\frac{\sum_{i=1}^{N_{t_c}} \left(L_{min} - L_f \right)^2}{N_{t_c}}} \tag{25}$$

$$MAE_{test} = \frac{N_{LOC}}{N_{t_c}} \tag{26}$$

$$RMSE_{test} = \sqrt{\frac{{N_{LOC}}^2}{N_{t_c}}} \tag{27}$$

$$ZM_{test} = \frac{N_{GLO}}{N_{t_c}} \tag{28}$$

The similarity index quantifies the relationship between the final state of the neuron and the ideal neuron state during the retrieval phase of GRAN3SAT. The definition of the S_i^{ideal} is as follows:

$$\mathrm{S}_i^{ideal} = \begin{cases} 1 & if\ A \\ -1 & if \neg A \end{cases} \tag{29}$$

where A is the positive literal and $\neg A$ is the negative literal existing in each clause of $P_{GRAN3SAT}$, and S_i^{ideal} is the ideal neuron state. Note that Equation (29) will consider the final neuron state that achieves global minimum energy. In this case, the Jaccard index $S_{Jaccard}$ [25] will be used to evaluate the quality of the final neuron state:

$$S_{Jaccard} = \frac{l}{l + m + n} \tag{30}$$

where S_{f_i} is the current final neuron state. Note that a lower value of $S_{Jaccard}$ is favored, since it shows higher diversity of the final neuron state.

To evaluate the flexibility of the logic structure, this paper proposes Equations (31)–(34) to quantify the degree of change in the logic structure from the perspectives of the number of clauses and the literal state. Equations (31) and (32) represent the number of clauses that fluctuate during the learning phase of GRAN3SAT, and Equations (33) and (34) represent the error resulting from the number of negative literals. In addition, Table 5 represents the parameters involved in evaluating the flexibility of the logic structure.

$$MAE_{NC} = \frac{\sum_{i=1}^{N_{combmax}} \left| N_{NC} - N_{mean\_c} \right|}{N_{combmax}} \tag{31}$$

$$RMSE_{NC} = \sqrt{\frac{\sum_{i=1}^{N_{combmax}} \left(N_{NC} - N_{mean\_c} \right)^2}{N_{combmax}}} \tag{32}$$

$$MAE_{literal} = \frac{\sum_{i=1}^{N_{combmax}} \left| N_{negative} - N_{mean\_p} \right|}{N_{combmax}} \tag{33}$$

$$RMSE_{literal} = \sqrt{\frac{\sum_{i=1}^{N_{combmax}} \left(N_{negative} - N_{mean\_p} \right)^2}{N_{combmax}}} \tag{34}$$

Figure 2 shows the overall implementation of the proposed GRAN3SAT. Figure 2 can be divided into two parts. First, the implementation process represents the actual process of the proposed GRAN3SAT from the learning phase to the retrieval phase. Second, the process line describes the phases of the parameter influence with different performance evaluation metrics.

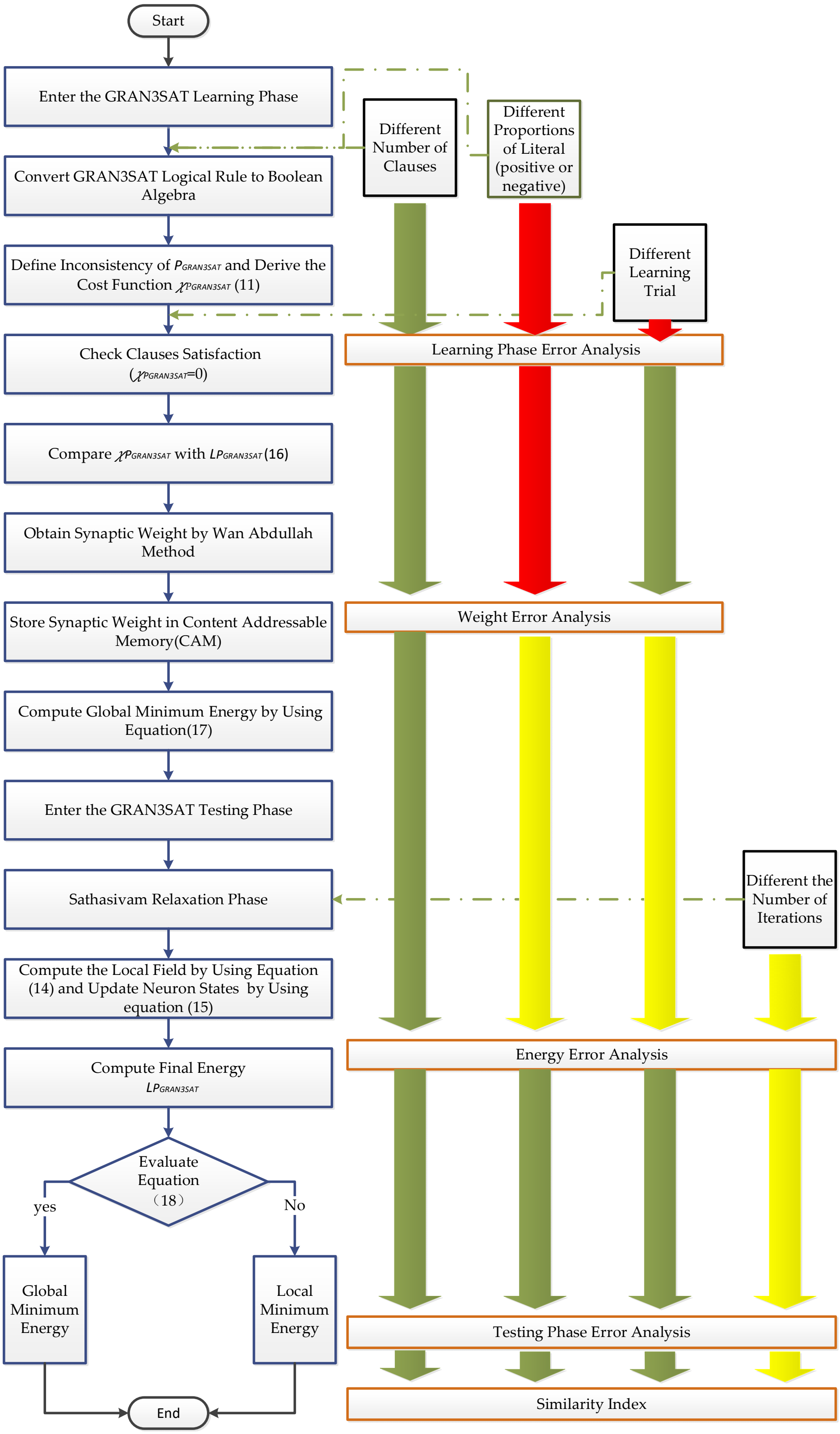

Start
Enter the GRAN3SAT Learning Phase
Convert GRAN3SAT Logical Rule to Boolean Algebra
Define Inconsistency of $P_{GRAN3SAT}$ and Derive the Cost Function $\chi_{P_{GRAN3SAT}}$ (11)
Check Clauses Satisfaction ($\chi_{P_{GRAN3SAT}}$=0)
Compare $\chi_{P_{GRAN3SAT}}$ with $L_{P_{GRAN3SAT}}$ (16)
Obtain Synaptic Weight by Wan Abdullah Method
Store Synaptic Weight in Content Addressable Memory(CAM)
Compute Global Minimum Energy by Using Equation(17)
Enter the GRAN3SAT Testing Phase
Sathasivam Relaxation Phase
Compute the Local Field by Using Equation (14) and Update Neuron States by Using equation (15)
Compute Final Energy $L_{P_{GRAN3SAT}}$
Evaluate Equation （18）
yes
No
Global Minimum Energy
Local Minimum Energy
End
Different Number of Clauses
Different Proportions of Literal (positive or negative)
Different Learning Trial
Learning Phase Error Analysis
Weight Error Analysis
Different the Number of Iterations
Energy Error Analysis
Testing Phase Error Analysis
Similarity Index

Figure 2. Implementation flowchart for the proposed GRAN3SAT.

5. Results and Discussion

To evaluate the effectiveness of the proposed model, GRAN3SAT will be evaluated based on four perspectives: different orders of clauses, different proportions between positive and negative literals, different numbers of iterations, and different numbers of learning trials. These perspectives will be evaluated based on various performance metrics. After finding the best setting from Sections 5.1–5.4, the proposed GRAN3SAT will be compared with existing model stated in Section 4.

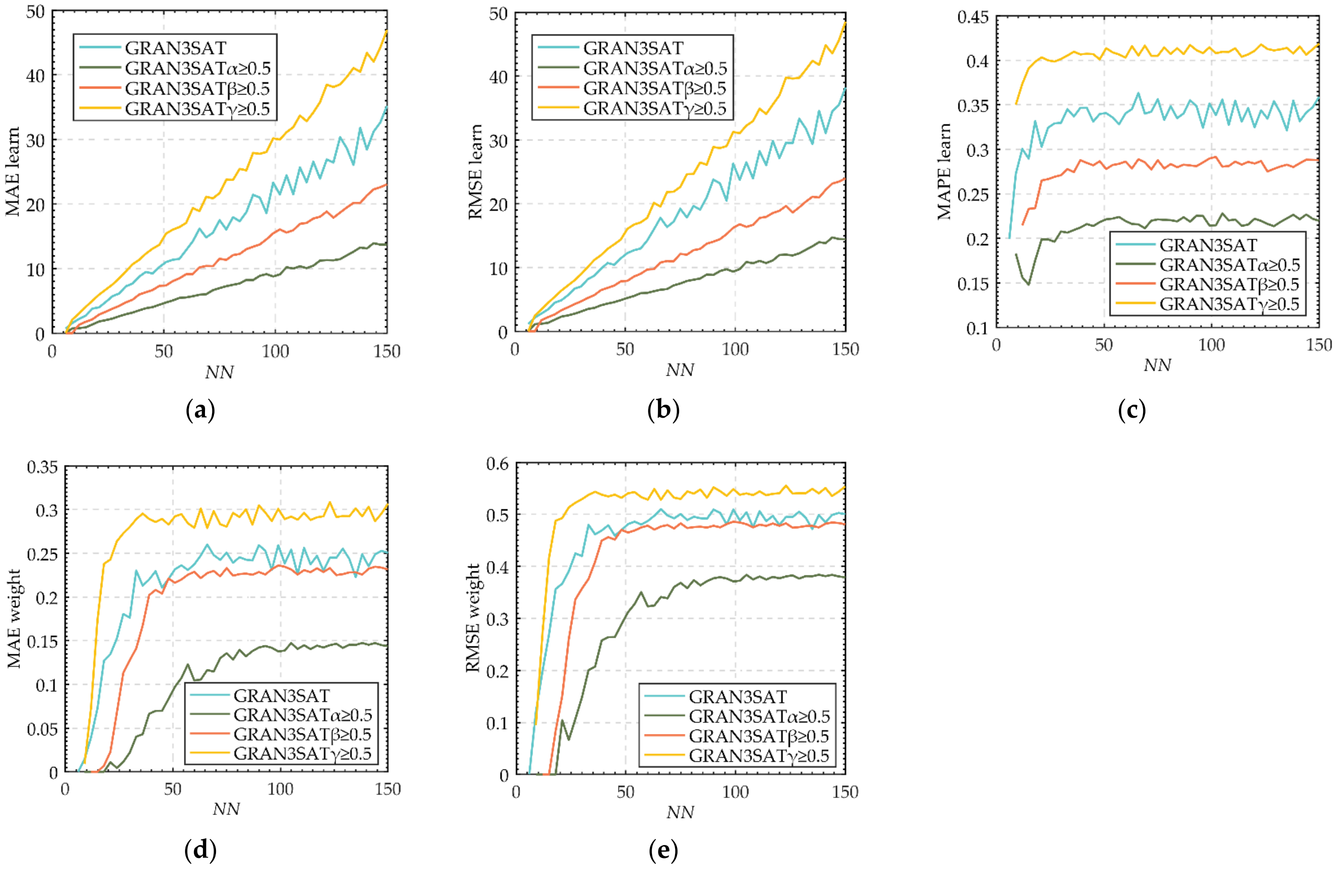
5.1. The Effect of Different Types of Clauses

The purpose of this section is to analyze the influence of the numbers of first-order k_i, second-order n_i, and third-order logic m_i clauses on the performance of the proposed GRAN3SAT. Since (k_i, n_i, m_i) are randomly generated, the proportions of the clauses will be adjusted based on the proportion of third-order (α), second-order (β), and first-order logicals (γ); the cases are shown in Table 6.

Table 6. Different cases for the GRAN3SAT model.

Case	Model	Proportion
Case I	GRAN3SAT	$rand(m_i, n_i, k_i)$
Case II	GRAN3SAT$\cdot\alpha \geq 0.5$	$\alpha = \frac{m_i}{m_i+n_i+k_i} \geq 0.5$
Case III	GRAN3SAT$\cdot\beta \geq 0.5$	$\beta = \frac{n_i}{m_i+n_i+k_i} \geq 0.5$
Case IV	GRAN3SAT$\cdot\gamma \geq 0.5$	$\gamma = \frac{k_i}{m_i+n_i+k_i} \geq 0.5$

Figure 3 demonstrates the performance of different GRAN3SAT models in terms of MAE_{learn}, $RMSE_{learn}$, $MAPE_{learn}$, MAE_{weight}, and $RMSE_{weight}$ during the learning phase of the DHNN. In order to assess the actual capability of GRAN3SAT with different proportions, Exhaustive Search Algorithm (ES) was implemented with all models during the retrieval phase. This learning method was proposed by [26], where the algorithm capitalizes trial and error to achieve a minimized cost function $\chi_{P_{GRAN3SAT}} = 0$. This causes the values of MAE_{learn} and $RMSE_{learn}$ for all GRAN3SAT models to increase as the number of neurons increases. Based on Figure 3a,b, the GRAN3SAT that has the highest proportion of α has the lowest values of MAE_{learn} and $RMSE_{learn}$. This shows that the third order clause has the capability to reduce the learning error of the proposed GRAN3SAT. This pattern was supported by the high error for the GRAN3SAT model that has a high value of β and γ. Another interesting perspective is that as the number of first-order logics increased, the performance of GRAN3SAT during the learning phase detriorated. This is due to the difficulty of the ES to find the consistent interpretation that satisfies GRAN3SAT that has more first-order logic. Despite the increase in error as the number of neurons increases, the ratio of values for both MAE_{learn} and $RMSE_{learn}$ are close to 1:1. This indicates the absence of outliers that potentially influence neuron fitness [27]. Based on Figure 3c, when $NN \geq 15$, the value of $MAPE_{learn}$ is relatively stable, reflecting the proportion of the number of unsatisfied clauses to the total number of clauses [24]. It is reported that Cases I, II, III, and IV will stabilize at around 0.33, 0.210, 0.280, and 0.410, respectively. As reported in Figure 3a,b, most of the clauses that are not satisfied are a result of high values of β and γ. This also confirms that lower $MAPE_{learn}$ can be achieved by GRAN3SAT if more third-order logic was generated in the formulation.



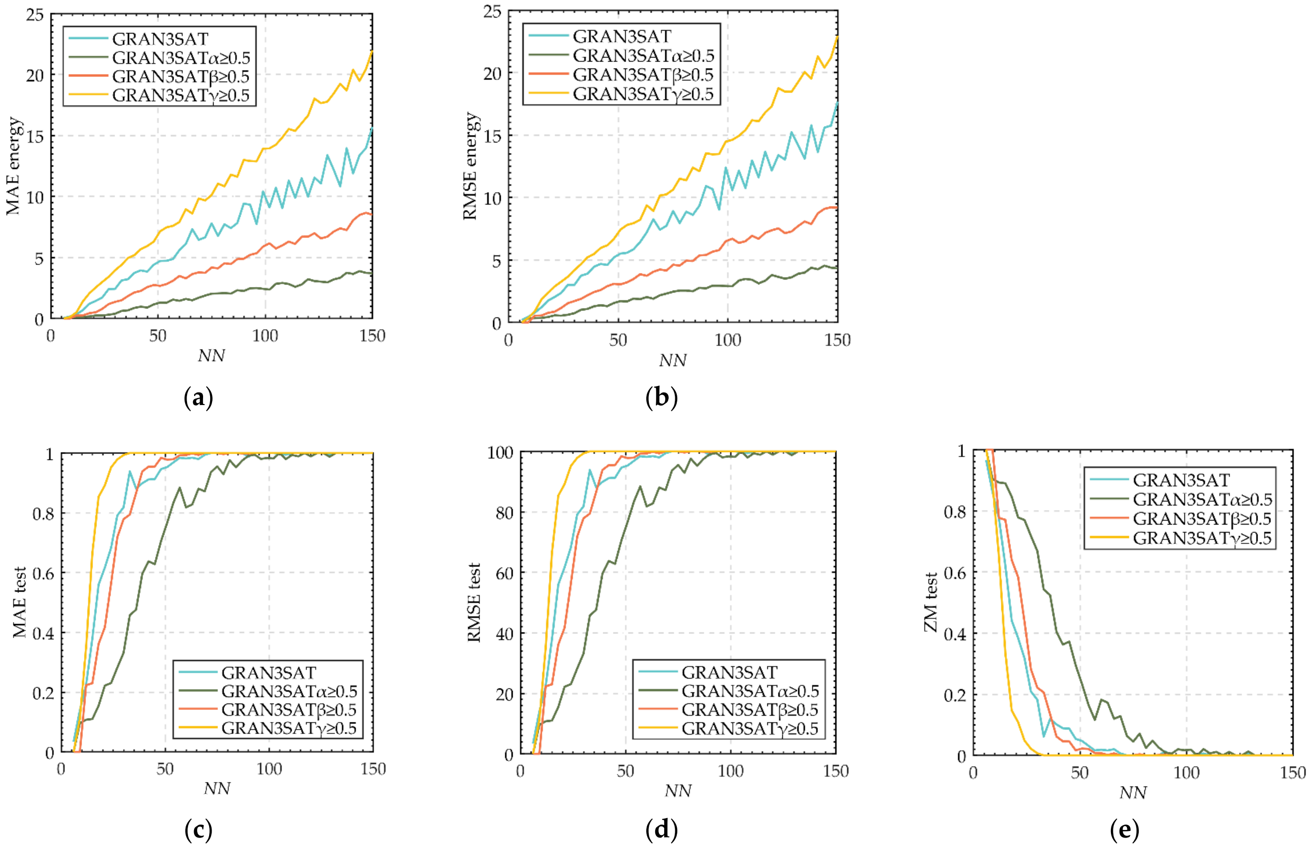
(**a**) (**b**) (**c**)

(**d**) (**e**)

Figure 3. Error performance for (**a**) MAE_{learn}, (**b**) $RMSE_{learn}$, (**c**) $MAPE_{learn}$, (**d**) MAE_{weight}, and (**e**) $RMSE_{weight}$.

Figure 3d,e demonstrate the MAE_{weight} and $RMSE_{weight}$ for all GRAN3SAT models. Note that this perspective is only for the restricted learning phase where the number of learning trials is pre-determined. Based on the result obtained, the values of MAE_{weight} and $RMSE_{weight}$ were the lowest when higher α was generated in the formula. This shows that the correct synaptic weight can be obtained when more third-order clauses are considered in the GRAN3SAT logical rules. The lowest performance of the GRAN3SAT model is when γ is the highest, since the DHNN requires more learning iterations to minimize the cost function. When MAE_{weight} was minimized, the DHNN is able to retrieve the optimal final neuron state which corresponds to the behavior of the GRAN3SAT model.

We now consider the capability of GRAN3SAT during the retrieval phase. Figure 4 demonstrates the performance of different GRAN3SAT models in terms of MAE_{energy}, $RMSE_{energy}$, MAE_{test}, $RMSE_{test}$, and ZM_{test} during the retrieval phase of the DHNN. In terms of MAE_{energy} and $RMSE_{energy}$, the GRAN3SAT with the highest proportion of α has the lowest values of MAE_{energy} and $RMSE_{energy}$. This shows that the difference between final energy and the absolute minimum energy is minimized as more third-order clauses are generated in GRAN3SAT. This is a result of the lower values of MAE_{weight} and $RMSE_{weight}$ that lead to optimal final neuron states. It is also reported that the MAE_{energy} and $RMSE_{energy}$ were obtained from lower-order clauses compared to third-order clauses. These findings have good agreement with the research of [28], where the closer the energy is towards the absolute minimum energy, the more stable the final neuron state becomes. It can be seen from Figure 4 that as the NN increases, the final state that corresponds to the global solution is difficult to obtain. Most of the solution is trapped as a local solution which results in higher values of MAE_{energy} and $RMSE_{energy}$. Thus, the values of MAE_{test} and $RMSE_{test}$ will also increase (refer to Figure 3c,d). In order to fully understand the value of MAE_{test} obtained by GRAN3SAT, we report the specific values of MAE_{energy} and $RMSE_{energy}$ for $MAE_{test} = 0.994$ when all final states of the DHNN reach local minima solutions in Table 7.



(**a**) (**b**)

(**c**) (**d**) (**e**)

Figure 4. Error performance for (**a**) MAE_{energy}, (**b**) $RMSE_{energy}$, (**c**) MAE_{test}, (**d**) $RMSE_{test}$, and (**e**) ZM_{test}.

Table 7. The values of MAE_{energy} and $RMSE_{energy}$ for different NN when $MAE_{test} = 0.994$.

Symbol	Case IV	Case III	Case II	Case I
NN	30	57	83	69
MAE_{energy}	3.949	3.087	2.471	6.444
$RMSE_{energy}$	4.183	3.474	2.938	7.416
$mean(NC_{3-SAT})$	2.760	4.100	21.370	8.590
$mean(NC_{2-SAT})$	3.460	19.100	5.860	9.930
$mean(NC_{1-SAT})$	14.760	6.470	7.130	23.370

According to Table 7, the final neuron state failed to achieve the global minimum solution in $2.471 \leq MAE_{energy} \leq 3.949$ for Case II to Case IV, whereas $MAE_{energy} = 6.444$ and $RMSE_{energy} = 7.416$ for Case I. GRAN3SAT also reported higher MAE_{energy} and $RMSE_{energy}$ at the critical value since there are no final neuron states that achieve the global minimum solution. This shows the importance of higher-order clauses to retrieve the best behavior of GRAN3SAT. The ratio of MAE_{energy} and $RMSE_{energy}$ is close to 1:1, indicating that there is no outlier in the energy distribution.

Using linear fitting in MATLAB [29], the relationship between the average number of k-order logic clauses and energy errors can be obtained (refer Table 8). It can be found that the average energy error of a single higher-order logical clause is less than that of a lower-order logical clause. This shows that the behavior of GRAN3SAT can be portrayed by assigning higher values of α.

Table 8. Average individual *k*-order logic clause and energy error linear fit coefficients.

Error	Third-Order Clause	Second-Order Clause	First-Order Clause
MAE_{energy}	0.012	0.074	0.247
$RMSE_{energy}$	0.027	0.088	0.257

According to Figure 4e, GRAN3SAT with higher α was reported to produce higher ZM_{test} compared to other proportions. The reason of the higher value of ZM_{test} is because the lower values of MAE_{learn} and $RMSE_{learn}$ in finding the consistent interpretation during the learning phase. Thus, optimal synaptic weight drives the final neuron state to converge to the global minimum solution. In this case, the local field in Equation (14) has a higher chance to satisfy the condition in Equation (18). On the contrary, a higher value of γ will reduce the probability of the DHNN to converge to the optimal final neuron state. This is due to only one value of synaptic weight that contributes to the update of the neuron state. Thus, the final neuron state is likely to be trapped in a local minimum solution.

While all the previous metrics focus on the evaluation of the number of GRAN3SAT solutions, it is equally important to evaluate the quality of the neuron state. We conduct similarity analysis to measure the similarity of the global solution for each GRAN3SAT model. In this section, the $S_{Jaccard}$ with the second-order fitting [30] are mainly applied to evaluate the performance for each model. Figure 5 represents the relationship between the $S_{Jaccard}$ with the second-order fitting and the NN under different models. Note that the bar chart reflects the fluctuation of the $S_{Jaccard}$ for the final neuron state where the curve represents the overall trend of the $S_{Jaccard}$ with the number of NN. As shown in Figure 5, the result for $S_{Jaccard}$ can be explained in three intervals. At interval $6 \leq NN \leq 27$ the $S_{Jaccard}$ for the GRAN3SAT model is $Case\ IV > Case\ III > Case\ I > Case\ II$, where α has a larger global solution space. Next, at interval $27 \leq NN \leq 36$, Case III was reported to outperform Case IV due to higher lower-order clauses (β,γ). This results from the greater probability for this case to obtain random W_{ij} and W_i during the learning phase of the DHNN. Finally, the value of Case I declines steadily at $NN > 36$ compared with the other cases. In this case, the global solution of GRAN3SAT at $NN > 36$ has stable and more diversified final neuron states. This can be explained by referring to Equation (14) where more synaptic weight was responsible for the neuron updates during the retrieval phase. However, the ineffectiveness of the ES during the learning phase of the DHNN creates more local minimum solutions which reduce the number of final neuron states that are different from each other. This phenomenon is more obvious as the NN increases. Consequently, the choice of GRAN3SAT that capitalizes higher-order clauses will increase the number of global solutions that eventually reduce the value of $S_{Jaccard}$. The feature makes GRAN3SAT with higher α more advantageous compared to other models.

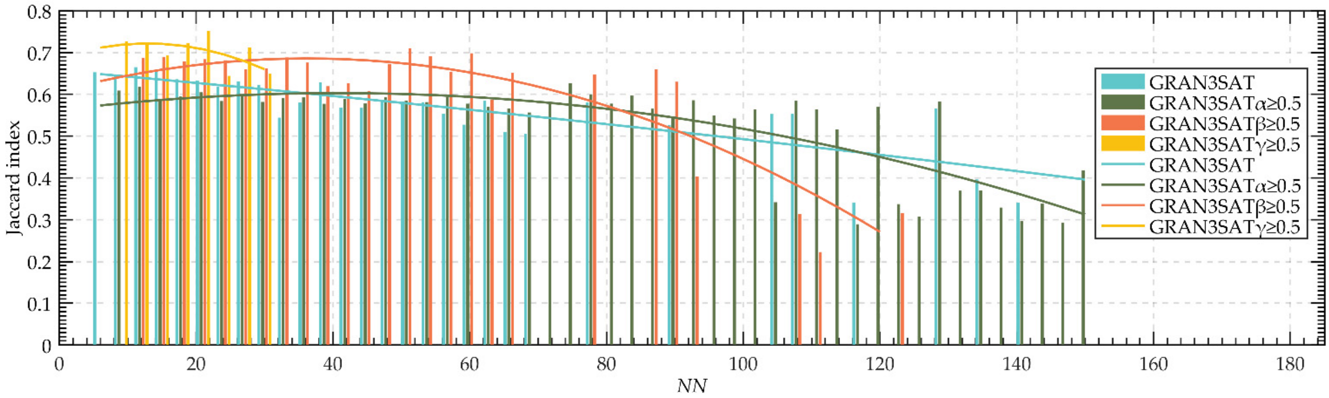


Figure 5. $S_{Jaccard}$ for different GRAN3SAT models.

5.2. The Effect of Different Proportions of Literals (Positive or Negative)

The purpose of this section is to analyze the influence of different proportions of literals (positive or negative) towards the performance of the GRAN3SAT model. Note that although the initial neuron state is randomly generated (1 or -1), the effect of the number of positive and negative literals can provide us insight on the behavior of our proposed GRAN3SAT model. Table 9 shows the proportion of the negative literal, P_N, of the neurons based on different GRAN3SAT models. Using the information in Table 9, we will investigate each GRAN3SAT model based on the effectiveness of the learning phase, retrieval phase, and similarity analysis.

Table 9. Different literal proportions for GRAN3SAT models.

Cases	Models	Remarks
Case I	GRAN3SAT	Literals (positive or negative) are randomly selected according to the system default Settings
Case II	GRAN3SAT$\cdot P_N = 0.9$	90% negative literals generated
Case III	GRAN3SAT$\cdot P_N = 0.7$	70% negative literals generated
Case IV	GRAN3SAT$\cdot P_N = 0.5$	50% negative literals generated
Case V	GRAN3SAT$\cdot P_N = 0.3$	30% negative literals generated
Case VI	GRAN3SAT$\cdot P_N = 0.1$	10% negative literals generated

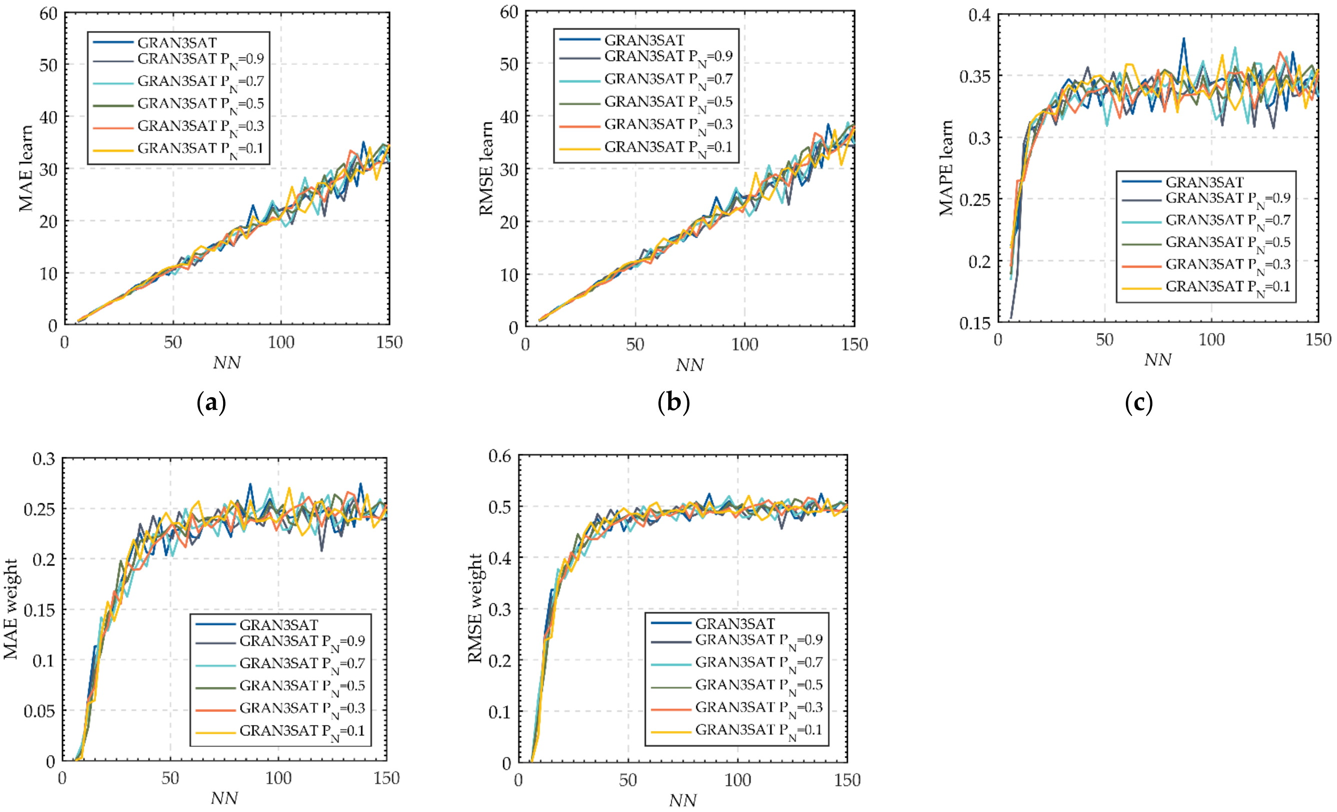
Figure 6 demonstrates the performance of different GRAN3SAT models in terms of MAE_{learn}, $RMSE_{learn}$, $MAPE_{learn}$, MAE_{weight}, and $RMSE_{weight}$ during the learning phase. Note that MAE_{learn}, $RMSE_{learn}$, and $MAPE_{learn}$ analyze the fitness of the neuron with different P_N, whereas MAE_{weight} and $RMSE_{weight}$ analyze the error as a result of incorrect synaptic weights with different P_N. As shown in Figure 6a–c, there is no obvious difference in terms of MAE_{learn}, $RMSE_{learn}$, and $MAPE_{learn}$ for all cases. This indicates that the learning phase of the GRAN3SAT model is not affected by different P_N. In order to support this statement, we illustrate the linear fittings of MAE_{learn} and $RMSE_{learn}$ in Table 10. Based on the value of the error, the slope of the fitting remains the same, which indicates that different P_N do not influence the learning capability of the GRAN3SAT model. We magnified our finding by extracting the value of $MAPE_{learn}$ for three different NN (refer Table 11). The curve remains stable at about 0.31–0.36 when $NN \geq 24$, which indicates that the percentage of neuron fitness does not change as the NN increases. Interestingly, the mixture of random occurrence clauses for $P_{GRAN3SAT}$ as stated in Equations (4)–(6) with different P_N does not increase or decrease the performance of the ES in finding the consistent interpretation. Similar observations were reported in MAE_{weight} and $RMSE_{weight}$ in Figure 6d,e. Compared to MAE_{learn}, the values of MAE_{weight} and $RMSE_{weight}$ increase in different ways. At $NN < 40$, MAE_{weight} and $RMSE_{weight}$ increase up to 0.23 and 0.48, respectively. After that, the values become stable since the optimal synaptic weight can no longer be obtained. Thus, GRAN3SAT is still able to perform stably, even with different P_N.

Table 10. The slopes of the straight-line fit of the error metrics.

Error	Case I	Case II	Case III	Case IV	Case V	Case VI
MAE_{learn}	0.226	0.230	0.223	0.221	0.229	0.227
$RMSE_{learn}$	0.247	0.251	0.243	0.240	0.249	0.248

Table 11. The values of $MAPE_{learn}$ for different NN.

NN	Case I	Case II	Case III	Case IV	Case V	Case VI
24	0.333	0.339	0.305	0.315	0.325	0.316
66	0.345	0.344	0.333	0.342	0.323	0.330
126	0.345	0.347	0.343	0.349	0.363	0.352



(**a**) (**b**) (**c**)

(**d**) (**e**)

Figure 6. Error performance for (**a**) MAE_{learn}, (**b**) $RMSE_{learn}$, (**c**) $MAPE_{learn}$, (**d**) MAE_{weight}, and (**e**) $RMSE_{weight}$.

Figure 7 presents the energy error distribution in terms of MAE_{energy}, $RMSE_{energy}$, MAE_{test}, and ZM_{test} for different values of P_N. Generally, the value of MAE_{energy} and $RMSE_{energy}$ fluctuate greatly as the NN increases. This is due to two contributing factors. The first is that it is not always possible to obtain satisfactory synaptic weights in the learning phase, thus that the energy function does not converge, resulting in a large MAE_{energy} and $RMSE_{energy}$. The second factor is the random structure of $P_{GRAN3SAT}$ from Equations (4)–(6). It is worth noting that the larger the proportion of P_N, the easier it is to form the two types of clauses $\neg A_i \vee \neg A_j$ and $\neg A_o \vee \neg A_p \vee \neg A_q$, and these clauses are more likely to produce larger final energy values. For instance, at $P_N = 0.9$ for $NN = 100$ (refer to Table 12) forms an average of 15.41 all negative for second-order clauses and 8.04 all negative for third-order clauses. On the contrary, at $P_N = 0.1$ for the same NN (refer to Table 12) forms an average of 0.13 all negative for second-order clauses and 0.01 all negative for third-order clauses. This shows that the behavior of the proposed GRAN3SAT is easier to display when lower P_N was employed. Next, MAE_{test}, $RMSE_{test}$, and ZM_{test} in Figure 7c–e show the impact of the different P_N towards the final neuron state. Based on these figures, the smaller the value of P_N, the higher the number of final states that achieve global minimum energy. For example, at $r = 2$ when the state $\neg A_i \vee \neg A_j$ and $\neg A_o \vee \neg A_p \vee \neg A_q$ are $\{1, 1\}$ and $\{1, 1, 1\}$, the proposed GRAN3SAT model was observed to obtain local minimum solutions. This is because a larger proportion of negative literals means that the polynomial coefficients of the energy function become more positive (100% negative literals correspond to 100% positive coefficients of the energy function). In this case, it is easier to produce larger final energy that results in local minimum energy. The trend of MAE_{test}, $RMSE_{test}$, and ZM_{test} can be divided into two intervals. The first interval is $6 \leq NN \leq 60$; the overall performance shows that the local minimum solution increases as the NN increases. Moreover, as for second interval at $NN > 60$, it can be observed that the final state of all cases are basically local solutions. Consequently, GRAN3SAT can increase the number of optimal final neuron states by increasing the number of positive literals.

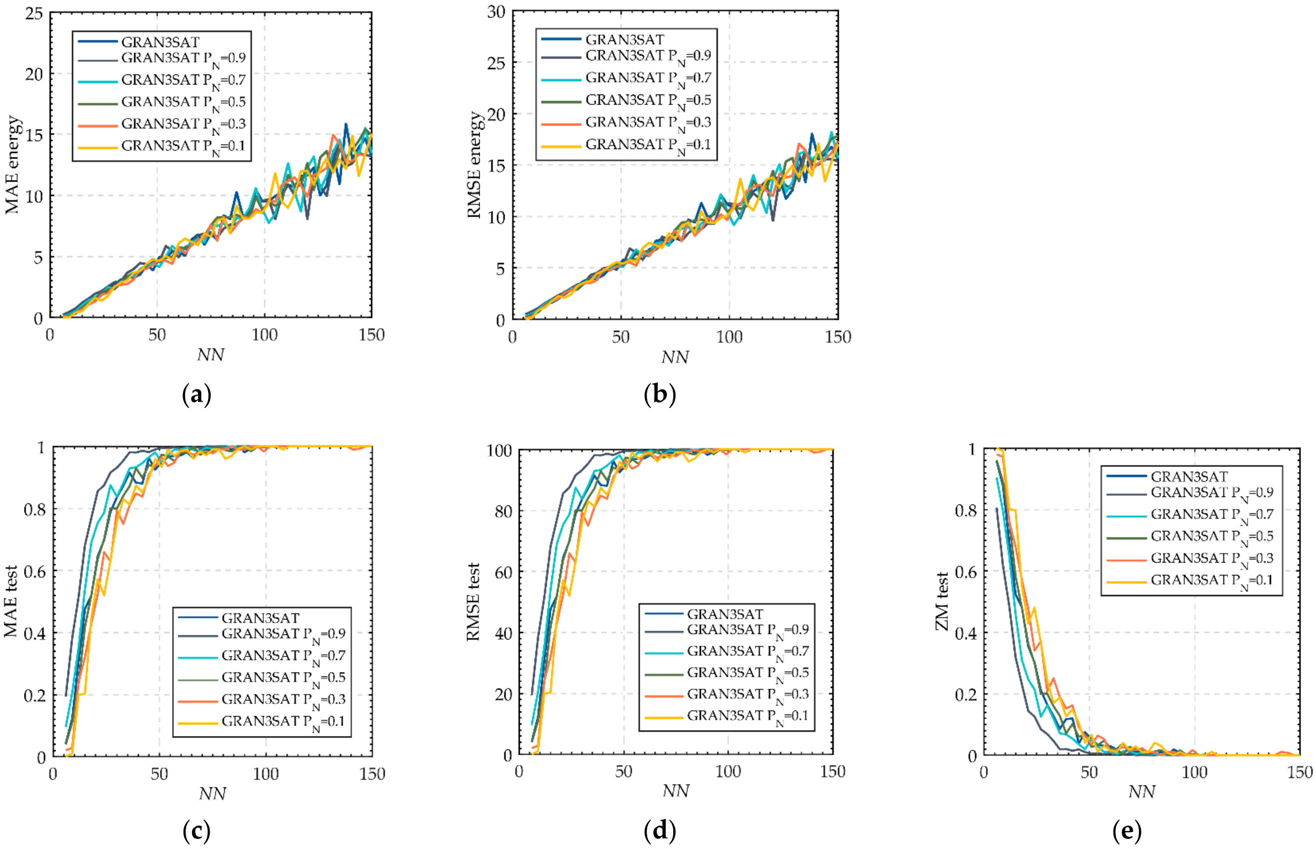


Figure 7. Error performance for (**a**) MAE_{energy}, (**b**) $RMSE_{energy}$, (**c**) MAE_{test}, (**d**) $RMSE_{test}$, and (**e**) ZM_{test}.

Table 12. The average NC when $NN = 100$.

Clause	Case I	Case II	Case III	Case IV	Case V	Case VI
$-A_i \vee -A_j$	4.080	15.410	6.700	4.070	1.480	0.130
$-A_o \vee -A_p \vee -A_q$	1.440	8.040	3.880	1.780	0.340	0.010

Figure 8 indicates the $S_{Jaccard}$ of GRAN3SAT for different values of P_N. Based on Figure 8, the final state of GRAN3SAT varies significantly and fluctuates in a certain range. In order to understand the actual performance of the final neuron state, the minimum and maximum values of the $S_{Jaccard}$ for GRAN3SAT are shown in Table 13. Since Case I generates completely random literals, the final neuron state in regards to $S_{Jaccard}$ shows the highest differences in magnitude. This shows that more random literals in GRAN3SAT will create different final neuron states. Another point to ponder is that high values of P_N in GRAN3SAT will increase the diversity of the final neuron state that achieved the global minimum solution. According to Table 13, the value of $S_{Jaccard}$ increased from 0.033 to 0.966 by just reducing the value of P_N from 0.9 to 0.1, respectively. This is because the lower the value of P_N, the greater the probability for GRAN3SAT to produce $A_o \vee A_p \vee A_q$ and $A_i \vee A_j$, which results in similar final neuron states. The main problem with producing the mentioned clauses is that the synaptic weight obtained during the learning phase tends to be monotonous in terms of vector and magnitude. As the proposed GRAN3SAT retrieves the final neuron state using Equations (14) and (15), the HTAF will classify the final state more towards a single type of neuron state; this will increase the value of $S_{Jaccard}$. Therefore, the proposed GRAN3SAT model will achieve an optimal final neuron state by reducing positive literals in $P_{GRAN3SAT}$ formulation.

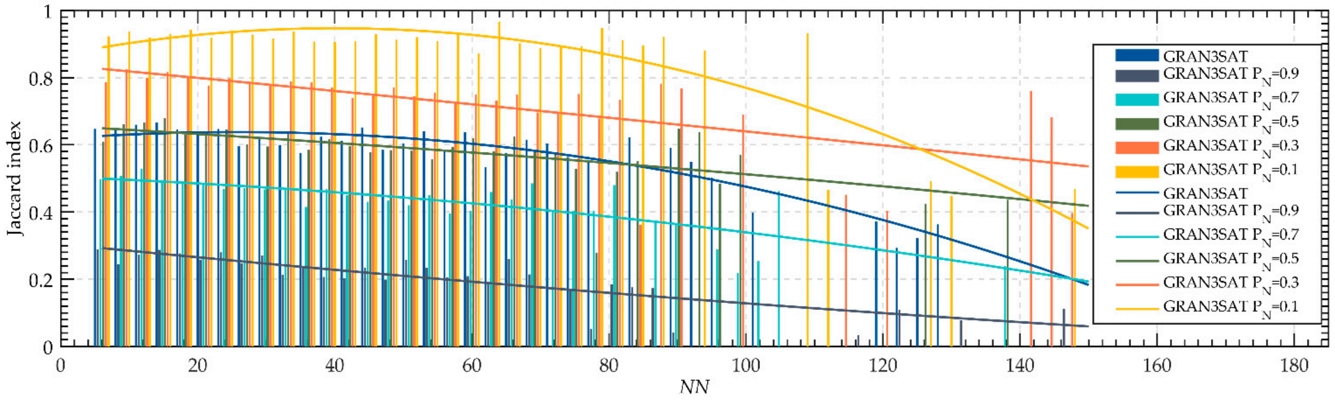


Figure 8. $S_{Jaccard}$ for GRAN3SAT for different values of P_N.

Table 13. The ranges of the $S_{Jaccard}$ values.

	Case I	**Case II**	**Case III**	**Case IV**	**Case V**	**Case VI**
MAX	0.666	0.288	0.528	0.679	0.82	0.966
MIN	0.292	0.033	0.217	0.278	0.362	0.448

From the above discussion, it can be concluded that different P_N will affect the energy error, test error, and especially the similarity index, but have no effect on clause satisfaction and synaptic weight error. During the retrieval phase, the larger the value of P_N, the higher the energy error, and the fewer global solutions. In the similarity index, the larger the value of P_N, the smaller the value of $S_{Jaccard}$. This means that a larger proportion of P_N will have fewer solutions that satisfy Equation (18). In this case, the tradeoff between P_N must be determined to ensure the optimality of our proposed GRAN3SAT.

5.3. The Effect of Different Learning Trials

The purpose of this section is to analyze the effect of different learning trials, N_{trial}, on the performance of the GRAN3SAT models. During the learning phase, the ES will be given a set of trials to ensure the cost function of the DHNN is minimized. In this case, the fitness of the neurons will be based on the number of satisfied clauses obtained from each N_{trial}. Note that the higher number of learning trials will assist the ES to explore a larger search space for the highest fitness that corresponds to a GRAN3SAT model. The results from this section will provide the theoretical support for improving the efficiency of the learning algorithm of our proposed GRAN3SAT model.

Figure 9a–c show the effect of the N_{trial} on the errors MAE_{learn}, $RMSE_{learn}$, and $MAPE_{learn}$ for the GRAN3SAT model. As shown in Figure 9a–c, the MAE_{learn}, $RMSE_{learn}$, and $MAPE_{learn}$ are similar despite having different values of N_{trial}. During the learning phase of GRAN3SAT, the neuron fitness is dependent on the number of clauses in $P_{GRAN3SAT}$, although it has been reported in the research of [15] that higher values of N_{trial} will increase the probability of the ES to minimize the cost function. The random feature of $P_{GRAN3SAT}$ in Equations (4)–(6) makes high values of N_{trial} appear insignificant. When the ES failed to achieve $\chi_{P_{GRAN3SAT}} = 0$, the difference between the current neuron fitness and the desired neuron fitness will increase. This was supported by the increase in MAE_{learn}, $RMSE_{learn}$, and $MAPE_{learn}$ as the number of NN increased.

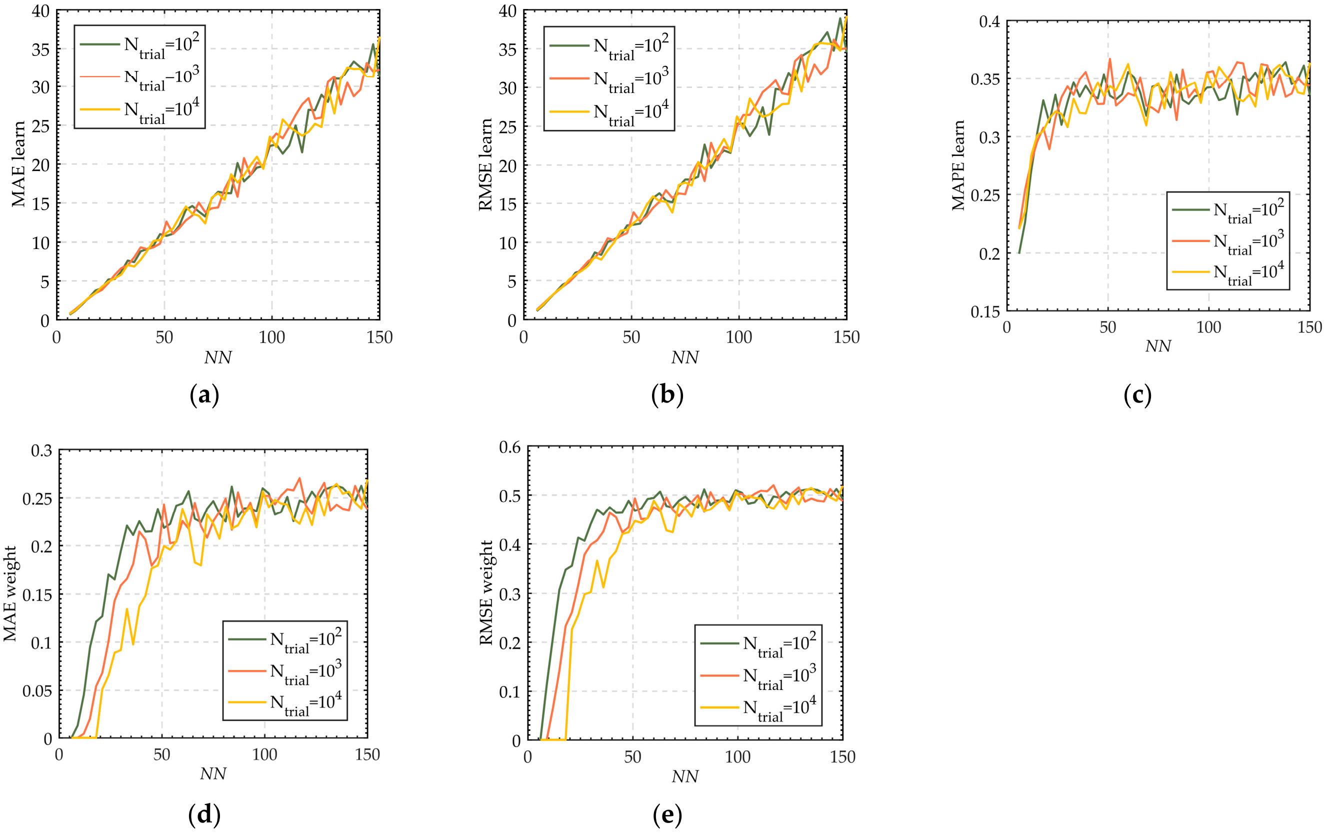


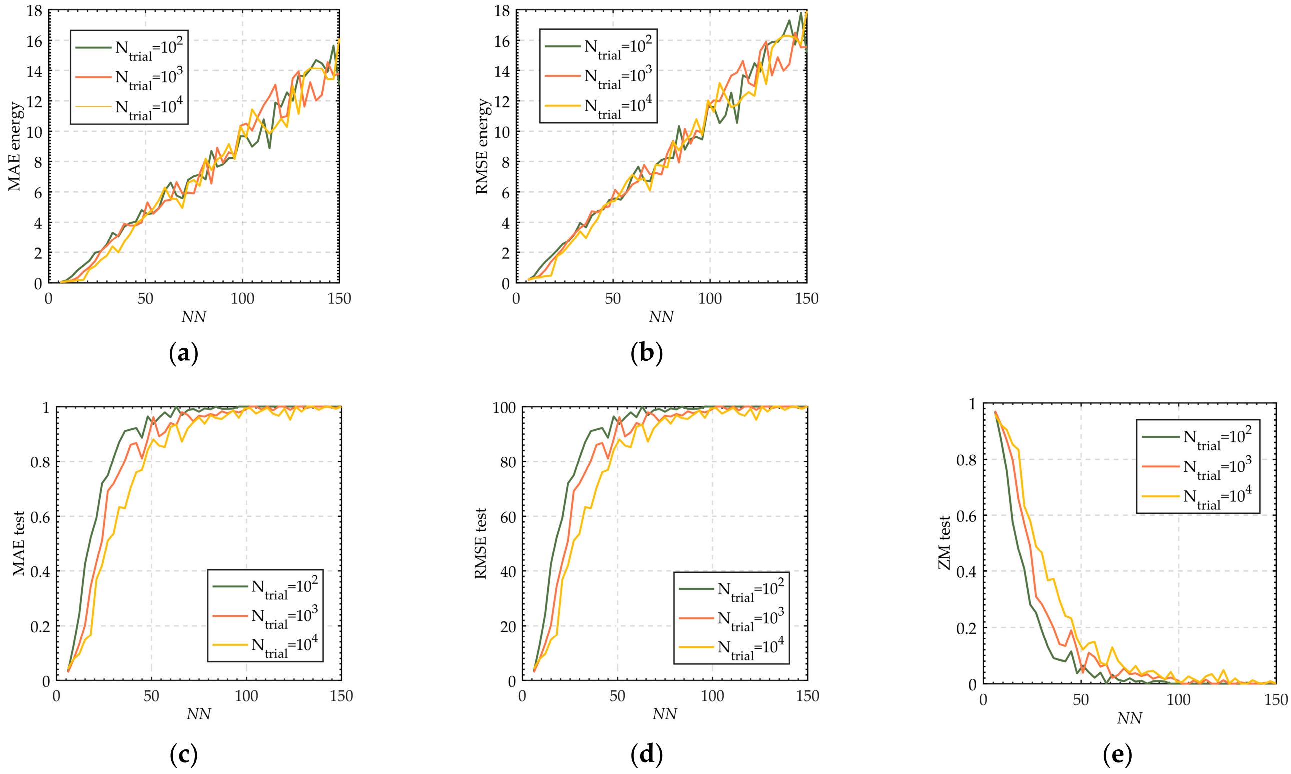
Figure 9. Error performance for (**a**) MAE_{learn}, (**b**) $RMSE_{learn}$, (**c**) $MAPE_{learn}$, (**d**) MAE_{weight}, and (**e**) $RMSE_{weight}$.

Figure 9d,e illustrate the effect of the N_{trial} on the errors MAE_{weight} and $RMSE_{weight}$ for the GRAN3SAT model. At $6 \leq NN \leq 63$, the value of MAE_{weight} and $RMSE_{weight}$ are the highest when N_{trial} is the lowest. This is because although the cost function of the neuron in GRAN3SAT is $\chi_{P_{GRAN3SAT}} \neq 0$, some fragment of the clause of $P_{GRAN3SAT}$ was satisfied and obtained the optimal synaptic weight. In this case, more N_{trial} will provide more solution space for the ES to find near optimal neuron fitness. Therefore, MAE_{weight} and $RMSE_{weight}$ can be reduced. Based on Table 14, at $NN > 63$, the values of MAE_{weight} and $RMSE_{weight}$ gradually stabilize, and it is difficult to satisfy the clause in Equations (4)–(6). This shows that the ES does not contribute to driving the neuron towards global maxima. It is worth noting that simply increasing the number of N_{trial} will increase both computational time and capacity of the GRAN3SAT model. On the contrary, the effect of high N_{trial} seems obvious if GRAN3SAT became systematic SAT, except that the clause produced is only a first-order clause. In order to remediate the situation, the learning options of GRAN3SAT can be improved by combining other meta-heuristic algorithms, such as the Grey Wolf Optimization (GWO) [31], Genetic algorithm (GA) [32], Election Algorithm (EA) [33], and Particle Swarm Optimization (PSO) [34].

Figure 10a,b illustrate the effect of the N_{trial} on the MAE_{energy} and $RMSE_{energy}$ for the GRAN3SAT model. Based on these figures, there is a small difference in the energy error between L_{min} (minimum energy) and L_f (final energy) as a whole, showing a trend of decreasing energy error as the N_{trial} value increases. This is closely related to the optimal synaptic weights obtained by GRAN3SAT during the learning phase. We confirm our findings by simulating the linear fitting of the MAE_{energy} with different values of N_{trial} (refer to Table 15). Overall, the MAE_{energy} and $RMSE_{energy}$ show a steady upward trend with the increase of NN. This is due to the increased numbers of third-order clauses and second-order clauses, resulting in slow network convergence.

Table 14. The values of MAE_{learn} and MAE_{weight} for GRAN3SAT.

NN	**6**	**21**	**63**	**75**	**105**
Learning error	MAE_{learn}	MAE_{learn}	MAE_{learn}	MAE_{learn}	MAE_{learn}
$N_{trial} = 10^2$	0.640	4.061	14.595	16.417	21.321
$N_{trial} = 10^3$	0.767	3.817	13.373	14.413	23.293
$N_{trial} = 10^4$	0.741	4.220	13.596	16.239	25.710
Weight error	MAE_{weight}	MAE_{weight}	MAE_{weight}	MAE_{weight}	MAE_{weight}
$N_{trial} = 10^2$	0	0.126	0.256	0.246	0.232
$N_{trial} = 10^3$	0	0.067	0.217	0.225	0.243
$N_{trial} = 10^4$	0	0.051	0.219	0.222	0.247



(**a**) (**b**)

(**c**) (**d**) (**e**)

Figure 10. Error performance for (**a**) MAE_{energy}, (**b**) $RMSE_{energy}$, (**c**) MAE_{test}, (**d**) $RMSE_{test}$, and (**e**) ZM_{test}.

Table 15. The slope of the linear fitting for MAE_{energy} .

N_{trial}	$N_{trial} = 10^2$	$N_{trial} = 10^3$	$N_{trial} = 10^4$
The slope of MAE_{energy}	0.102	0.102	0.106

Figure 10c–e and Table 16 demonstrate the effect of N_{trial} on the MAE_{test}, $RMSE_{test}$, and ZM_{test} for the GRAN3SAT model. As the N_{trial} increased from $N_{trial} = 10^2$ to $N_{trial} = 10^3$, the sharp increase of MAE_{test} for $6 < NN < 63$ to $6 < NN < 87$ was about 42%. On the other hand, as the N_{trial} increased from $N_{trial} = 10^3$ to $N_{trial} = 10^4$, the sharp increase of MAE_{test} for $6 < NN < 87$ to $6 < NN < 93$ was about 7%. This indicates that the larger the value of N_{trial}, the more likely for GRAN3SAT to obtain more global minimum solutions. However, an extremely high value of N_{trial} may increase the number of global solutions, but at the expense of high computational complexity. Therefore, although $N_{trial} = 10000$ has the best performance in terms of the MAE_{test}, $RMSE_{test}$, and ZM_{test}, it is not the best choice for GRAN3SAT efficiency.

Table 16. The ranges of MAE_{test} for different N_{trial}.

Trial	Interval	MAE_{test}	Interval	MAE_{test}
$N_{trial} = 10^2$	$6 < NN < 63$	0.030–0.990	$NN \geq 63$	0.990–1.000
$N_{trial} = 10^3$	$6 < NN < 87$	0.028–0.982	$NN \geq 87$	0.982–1.000
$N_{trial} = 10^4$	$6 < NN < 93$	0.042–0.985	$NN \geq 93$	0.985–1.000

Figure 11 demonstrates the $S_{Jaccard}$ of the proposed GRAN3SAT model with different values of N_{trial}. Note that the bar graph represents the change of the $S_{Jaccard}$ with the NN, and that the curve represents the changing trend of the $S_{Jaccard}$ with different N_{trial}. Figure 11 shows a downward trend of $S_{Jaccard}$ because the number of final states that achieved global minimum solutions reduced dramatically as the number of NN increased. Interestingly, the final neuron state for the proposed GRAN3SAT model has the best performance when $N_{trial} = 100$. This further indicates that the increase in N_{trial} does not further improve the diversity of the final neuron state. Another interesting point of view is the property of $P_{GRAN3SAT}$ that can be transformed into systematic logical rules. When this transformation occurs, fewer N_{trial} are required to obtain the lower $S_{Jaccard}$. This is an obvious exception if and only if the $P_{GRAN3SAT}$ was transformed into a first-order clause, as indicated in Equation (6). In this case, it will be very difficult for us to examine the behavior of the GRAN3SAT model as a result of the absence of solutions that can achieve global minima ratios.

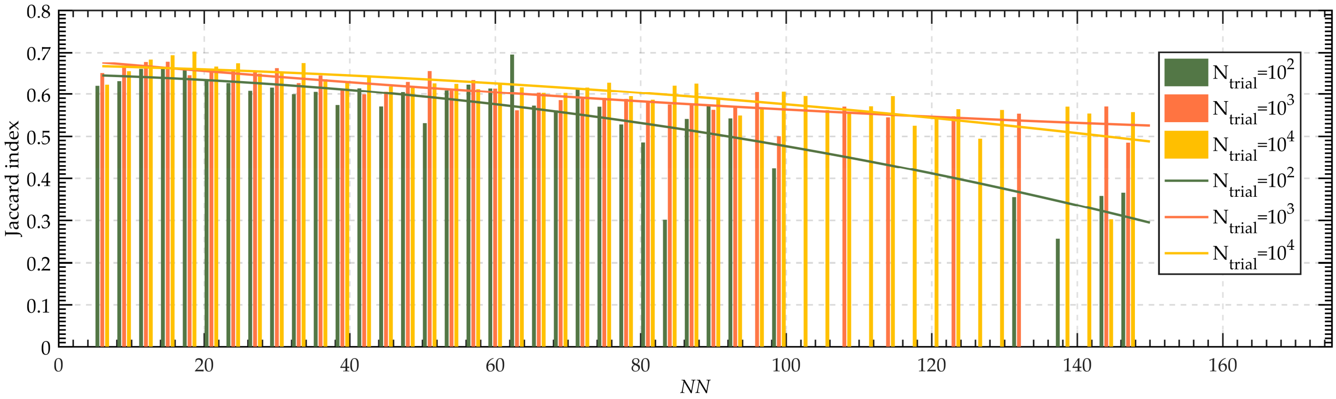


Figure 11. $S_{Jaccard}$ for GRAN3SAT for different values of N_{trial}.

In summary, optimal GRAN3SAT can be obtained by increasing the number of N_{trial} so that the learning error can be further reduced. When the error remains low, the final neuron state of the GRAN3SAT model will be optimized and will converge to a global minimum solution. The increase in N_{trial} must not be large until it increases the computational complexity of the proposed GRAN3SAT. By optimizing the learning phase and the retrieval phase of the GRAN3SAT model, diversity of the $S_{Jaccard}$ can be increased. By considering all the factors proposed in the previous results, $N_{trial} = 100$ was chosen as an optimal value for our proposed GRAN3SAT model. The use of several meta-heuristic algorithms, such as those provided by [35,36], are crucial in constructing a more optimal GRAN3SAT model.

5.4. The Effect of Relaxation in GRAN3SAT

The purpose of this section is to explore the effect of DHNN relaxation on GRAN3SAT performance. This analysis was inspired by the research of Sathasivam [8] where a relaxation phase during the retrieval phase is vital to ensure the DHNN will converge to an optimal final neuron state. According to that paper, without appropriate relaxation, the DHNN will not display the behavior of the logical rule due to some neurons becoming trapped in local minima solutions. In this section, we will discuss the effect of the relaxation towards GRAN3SAT according to various numbers of relaxations. Figure 12 illustrates the effectiveness of relaxation for $1 \le r \le 5$ in terms of MAE_{energy}, $RMSE_{energy}$, MAE_{test}, and $RMSE_{test}$ in GRAN3SAT.

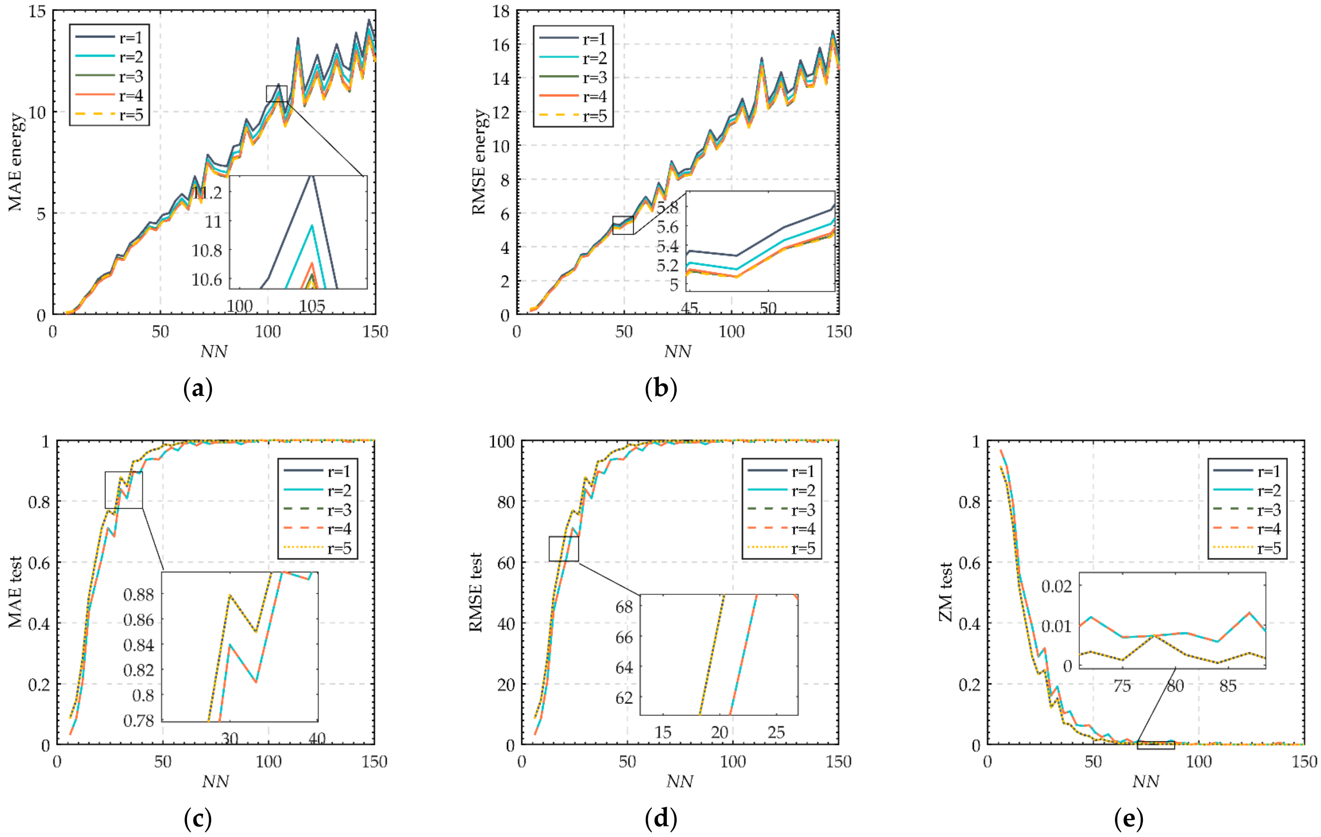


Figure 12. Error performance for (**a**) MAE_{energy}, (**b**) $RMSE_{energy}$, (**c**) MAE_{test}, (**d**) $RMSE_{test}$, and (**e**) ZM_{test}.

As shown in Figure 12a,b, the values of MAE_{energy} and $RMSE_{energy}$ were the highest when a minimum number of relaxations $(r = 1)$ was implemented. This is a result of the poor convergence property toward global minimum energy, even if a short relaxation was implemented. This is after considering the effect of the random weight (due to an ineffective learning phase), and the structure of $P_{GRAN3SAT}$ formulation in the DHNN. However, the MAE_{energy} and $RMSE_{energy}$ improve as the number of r is increased because the local field in Equations (14) and (15) will move the state of the neuron to the global solution. Despite the final energy differences, the numbers of final states (Figure 12c,d) that were trapped in local minimum solutions were almost the same for the case of $r = 2$ and $r = 4$. This shows that the energy distribution of the Lyapunov energy function for GRAN3SAT is stable and optimal. Interestingly, the performance of the global minima ratio (refer to Figure 12) is $ZM(r = 2, 4) > ZM(r = 1, 3, 5)$. These data show that there is a phenomenon of mutual conversion between the local solution and the global solution during the retrieval phase of the DHNN. Since the DHNN requires more computational iterations at $r = 4$, we can conclude that $r = 2$ is the optimal number of relaxations for GRAN3SAT.

Figure 13 represents the $S_{Jaccard}$ of the proposed GRAN3SAT model with different numbers of neurons. Note that the bar graph represents the change of the $S_{Jaccard}$ with the NN, and that the curve represents the changing trend of the $S_{Jaccard}$ when $1 \le r \le 5$. As shown in Figure 13, $S_{Jaccard}$ was reported to obtain the highest value when $r = 1$. Interestingly, at $NN \le 54$ the value of $S_{Jaccard}$ for $r = 2$ and $r = 4$ is the same. This pattern has been reported for $r = 3$ and $r = 5$ (refer to Table 17). Such a result is due to the regular alternation or repetition of the neuron states in the clauses during the relaxation process. On the other hand, the regularity of the value for $S_{Jaccard}$ is due to relaxation, which is not obvious for $2 \leq r \leq 5$ at $54 < NN < 102$. This indicates that the final neuron states of GRAN3SAT are consistently similar to each other. Unfortunately, the value of $S_{Jaccard}$ cannot be determined at $NN > 102$ since there is no final neuron state that achieves global minima solutions. Thus, the trend of the second-order fitting curve of the $S_{Jaccard}$ has diverged. This is a result of an ineffective learning phase of GRAN3SAT which leads to random generation of synaptic weight as the number of neurons increases. Thus, the diversity of the global solution cannot be examined. In summary, the proposed GRAN3SAT model obtained the most diversified final state when $r = 2$. The higher value of relaxation $r > 2$ is not favored by GRAN3SAT due to the high computational iterations involved, despite delivering the same value for $S_{Jaccard}$.

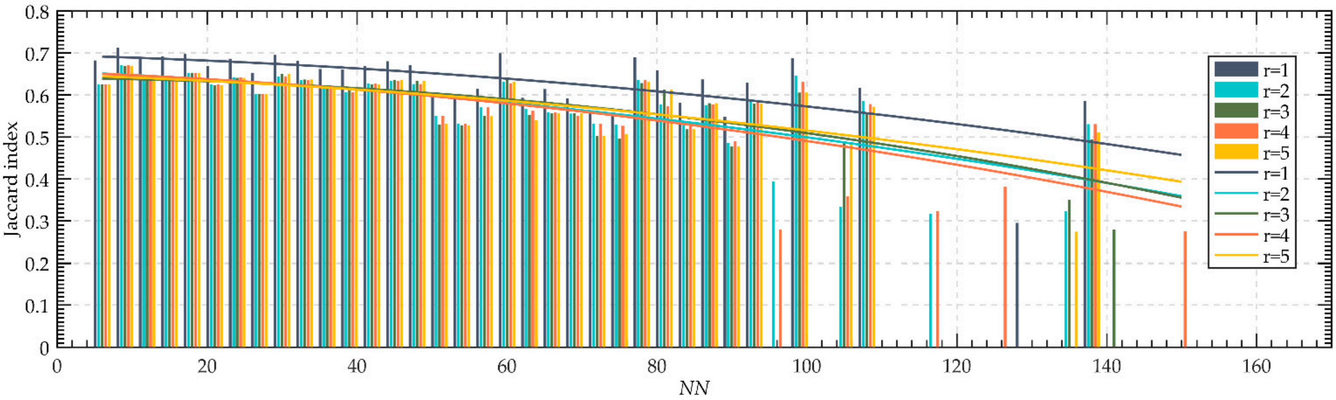


Figure 13. $S_{Jaccard}$ for GRAN3SAT for different values of r.

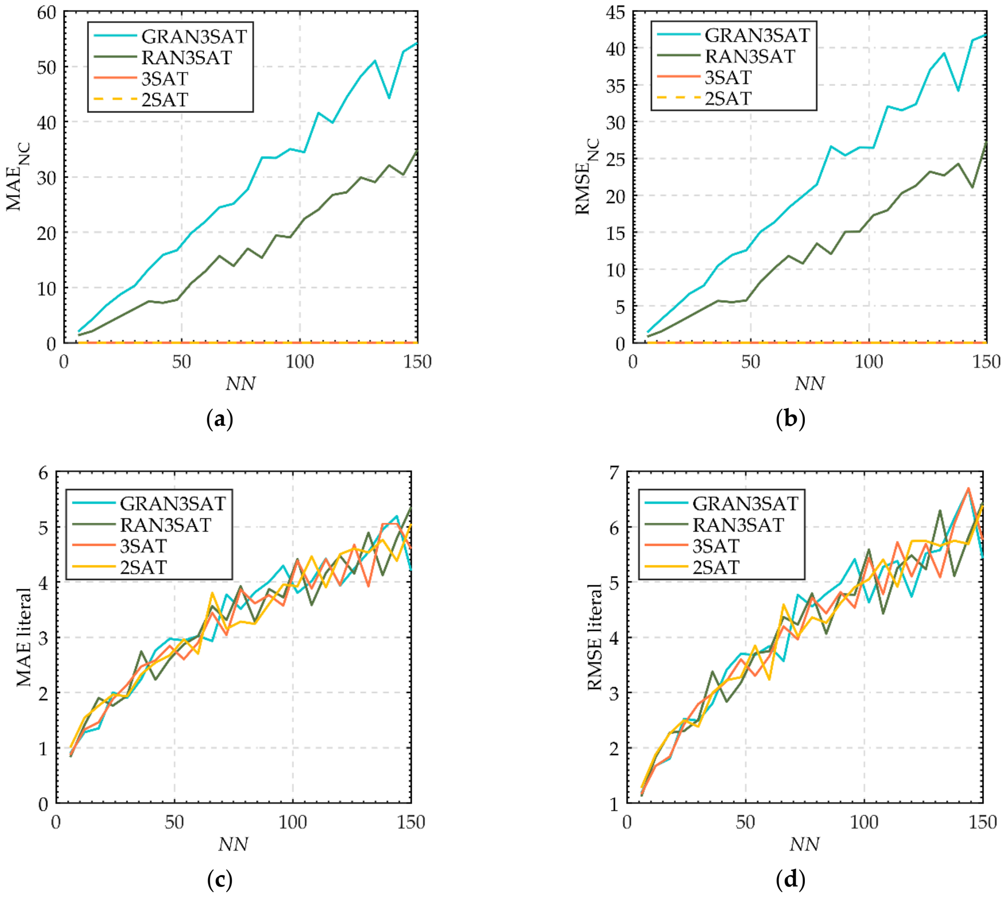
Table 17. $S_{Jaccard}$ under different relaxations and numbers of neurons (NN). "-" represents all the final states of the GRAN3SAT model trapped in local minimum solutions.

NN	6	18	30	42	54	66
$r = 1$	0.682	0.697	0.695	0.668	0.593	0.613
$r = 2$	0.625	0.652	0.644	0.627	0.531	0.558
$r = 3$	0.624	0.652	0.650	0.624	0.527	0.555
$r = 4$	0.625	0.652	0.644	0.627	0.531	0.558
$r = 5$	0.624	0.652	0.650	0.624	0.527	0.556
NN	78	90	102	114	126	138
$r = 1$	0.690	0.548	-	-	-	0.585
$r = 2$	0.635	0.485	-	-	-	0.530
$r = 3$	0.628	0.477	-	-	-	0.494
$r = 4$	0.635	0.489	-	-	0.382	0.531
$r = 5$	0.630	0.477	-	-	-	0.510

5.5. Comparisons with the Existing Work

The purpose of this section is to analyze the flexibility of the proposed GRAN3SAT model from the perspectives of the changes for both numbers of clauses and literals. The flexibility of the proposed model will be compared with $P_{RAN3SAT}$ [15], P_{3SAT} [20], and P_{2SAT} [9] in terms of error analysis. Since it is difficult for us to compare the effectiveness of the GRAN3SAT model, we propose two perspectives of the error analysis. First, we will evaluate the flexibility of the GRAN3SAT model from the perspective of clause. In this case, higher values of error indicate greater flexibility of the clause. In this case, MAE_{NC} and $RMSE_{NC}$ are the chosen metrics to evaluate the error of the number of clauses and the average numbers of clauses, respectively. Secondly, we will evaluate the flexibility of the proposed GRAN3SAT model from the perspective of literals. In this case, higher values of error indicate greater flexibility of the literals in the proposed logical rule compared to the existing work.

According to Figure 14a,b, the MAE_{NC} and $RMSE_{NC}$ for GRAN3SAT are reported to outperform all the existing networks. This shows that GRAN3SAT acquires good flexibility in producing randomized clausal arrangements which interchangeably create first, second, and third-order logic. On the other hand, $P_{RAN3SAT}$ shows a competitive flexibility resulting from a property of the nonsystematic logical rule compared to systematic logic (P_{3SAT} and P_{2SAT}). Although the $P_{RAN3SAT}$ shows absolute differences and high deviation from the number of clauses, random arrangement of GRAN3SAT was shown to cover more solution sets and possible logical interpretations. This is indicated by the highest values of MAE_{NC} and $RMSE_{NC}$ achieved by GRAN3SAT. Thus, GRAN3SAT is expected to provide more possible synaptic weight compared to RAN3SAT. As expected, systematic logical rule (P_{3SAT} and P_{2SAT}) has the lowest flexibility error due to the constant number of clauses for all values of NN. In other words, interpretation from the systematic logical rule is more rigid and will never produce different sets of logical formulation. We also observe that the probability for our proposed GRAN3SAT model to be reduced to systematic logical rules, such as P_{3SAT} and P_{2SAT}, will approach zero as the number of neurons increases. This shows that GRAN3SAT has a higher tendency towards nonsystematic logical rules compared to the other existing systematic logical rules. This behavior suggests that GRAN3SAT has more possible sets of the synaptic weight which will make the proposed DHNN retrieve more neuron states. As shown in Figure 14c,d, the values of $MAE_{literal}$ and $RMSE_{literal}$ for GRAN3SAT are not significantly different compared to those of existing work. Since the probability of obtaining positive and negative literals is similar for all types of logic, GRAN3SAT shares the same characteristics as RAN3SAT in terms of literals. Thus, we can conclude that the proposed GRAN3SAT model capitalizes the feature of the existing nonsystematic logical rules despite having a totally random clausal arrangement.



(**a**) (**b**) (**c**) (**d**)

Figure 14. Flexibility error analysis of the proposed GRAN3SAT model compared with RAN3SAT [15], 3SAT [20], and 2SAT [9]: (**a**) MAE_{NC}, (**b**) $RMSE_{NC}$, (**c**) $MAE_{literal}$, and (**d**) $RMSE_{literal}$.

6. Conclusions

Creating DHNN with flexible final neuron state is imperative in the field of machine learning. By formulating the most flexible logical rule, the neuron structure of the conventional DHNN will exhibit larger storage capacity and has the ability to explore high dimensional problem. In this paper, we propose a novel logical rule namely $P_{GRAN3SAT}$, by randomly generating the first-order, second-order, and third-order clauses. By incorporating third-order clauses in the $P_{GRAN3SAT}$, the potential diversity in the learning and retrieval phases of the logic increased. The proposed $P_{GRAN3SAT}$ was implemented into the DHNN (GRAN3SAT) by minimizing the cost function of the network. In that regard, the minimized cost function will help us in computing the optimal synaptic weight of the proposed GRAN3SAT. The efficiency and robustness of the proposed GRAN3SAT model was verified by using four extensive simulations, such as different orders of clauses, different proportions between positive and negative literals, and differing numbers of iterations and learning trials. Based on the simulations, the optimal value for each parameter in GRAN3SAT was reported. Next, the proposed GRAN3SAT model was compared with state-of-the-art methods, such as RAN3SAT, 3SAT, and 2SAT. The simulation reported greater flexibility of the proposed GRAN3SAT method compared with other existing works.

The flexible architecture of the proposed GRAN3SAT model provides an alternative insightinto the possible random dynamics for the application of real-life bioinformatic problems. For instance, the proposed GRAN3SAT model can be embedded into logic mining, which extracts the best logical rule that classifies single nucleotide polymorphisms (SNPs) inside known genes associated with Alzheimer's disease. This can lead to the discovery of the optimal logic mining incorporated with GRAN3SAT, which has the ability to classify and predict.

Author Contributions: Conceptualization and methodology, Y.G. (Yueling Guo); validation, N.A.R.; investigation, W.C.; writing—original draft preparation, Y.G. (Yuan Gao); supervision and funding acquisition, M.S.M.K.; visualization, J.C.; writing—review and editing, M.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Ministry of Higher Education Malaysia for Transdisciplinary Research Grant Scheme (TRGS) with Project Code: TRGS/1/2020/USM/02/3/2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express special thanks to all researchers in the AI Research Development Group for their continued support.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Notation	Explanation
AI	Artificial Intelligence
ANN	Artificial Neural Network
HNN	Hopfield Neural Network
DHNN	Discrete Hopfield Neural Network
SAT	Satisfiability Problem
3SAT	3 Satisfiability
2SAT	2 Satisfiability
$mean(NC_{k-SAT})$	Average number of *k*-order clauses
RAN*k*SAT	Random *k* Satisfiability
GRAN*k*SAT	G-Type Random *k* Satisfiability
RAN3SAT	Random 3 Satisfiability
HTAF	Hyperbolic Tangent Activation Function

RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percent Error
ZM	Ratio of global minimum energy
GRAN3SAT	G-Type Random 3 Satisfiability in DHNN
ES	Exhaustive Search
$P_{GRAN3SAT}$	General formula GRAN3SAT
m_i	Number of the third-order logic clauses
n_i	Number of the second-order logic clauses
k_i	Number of the first-order logic clauses
S_i	State of neuron i
W_{ij}	Synaptic weight between i and j
W_{ijk}	Synaptic weight between three neurons
$\chi_{P_{GRAN3SAT}}$	Cost function of the GRAN3SAT
NN	Number of neurons
NC	Number of clauses
N_{trial}	Number of learning trial
r	Relaxation rate
$rand(m_i, n_i, k_i)$	Number of clauses generated randomly

References

1. Pavlicko, M.; Vojteková, M.; Blažeková, O. Forecasting of Electrical Energy Consumption in Slovakia. *Mathematics* **2022**, *10*, 577. [CrossRef]
2. Xu, W.; Fu, Z.; Xi, Q. Thermal Conductivity Identification in Functionally Graded Materials via a Machine Learning Strategy Based on Singular Boundary Method. *Mathematics* **2022**, *10*, 458. [CrossRef]
3. Tam, V.W.; Butera, A.; Le, K.N.; Da Silva, L.C.; Evangelista, A.C. A prediction model for compressive strength of CO_2 concrete using regression analysis and artificial neural networks. *Constr. Build. Mater.* **2022**, *324*, 126689. [CrossRef]
4. Skrypnik, A.N.; Schelchkov, A.V.; Gortyshov, Y.F.; Popov, I.A. Artificial neural networks application on friction factor and heat transfer coefficients prediction in tubes with inner helical-finning. *Appl. Therm. Eng.* **2022**, *206*, 118049. [CrossRef]
5. Tretiakova, R.; Setukha, A.; Savinkov, R.; Grebennikov, D.; Bocharov, G. Mathematical Modeling of Lymph Node Drainage Function by Neural Network. *Mathematics* **2021**, *9*, 3093. [CrossRef]
6. Hopfield, J.J.; Tank, D.W. "Neural" computation of decisions in optimization problems. *Biol. Cybern.* **1985**, *52*, 141–152. [CrossRef]
7. Abdullah, W.A.T.W. Logic programming on a neural network. *Int. J. Intell. Syst.* **1992**, *7*, 513–519. [CrossRef]
8. Sathasivam, S. Upgrading logic programming in Hopfield network. *Sains Malays.* **2010**, *39*, 115–118.
9. Kasihmuddin, M.S.M.; Mansor, M.A.; Basir, M.F.M.; Sathasivam, S. Discrete mutation Hopfield neural network in propositional satisfiability. *Mathematics* **2019**, *7*, 1133. [CrossRef]
10. Mansor, M.A.; Kasihmuddin, M.S.M.; Sathasivam, S. Artificial Immune System Paradigm in the Hopfield Network for 3-Satisfiability Problem. *Pertanika J. Sci. Technol.* **2017**, *25*, 1173–1188.
11. Alzaeemi, S.; Mansor, M.A.; Kasihmuddin, M.S.M.; Sathasivam, S.; Mamat, M. Radial basis function neural network for 2 satisfiability programming. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *18*, 459–469. [CrossRef]
12. Mansor, M.A.; Jamaludin, S.Z.M.; Kasihmuddin, M.S.M.; Alzaeemi, S.A.; Basir, M.F.M.; Sathasivam, S. Systematic boolean satisfiability programming in radial basis function neural network. *Processes* **2020**, *8*, 214. [CrossRef]
13. Kasihmuddin, M.S.M.; Mansor, M.A.; Sathasivam, S. Discrete Hopfield neural network in restricted maximum k-satisfiability logic programming. *Sains Malays.* **2018**, *47*, 1327–1335. [CrossRef]
14. Sathasivam, S.; Mansor, M.A.; Ismail, A.I.M.; Jamaludin, S.Z.M.; Kasihmuddin, M.S.M.; Mamat, M. Novel Random k Satisfiability for $k \leq 2$ in Hopfield Neural Network. *Sains Malays.* **2020**, *49*, 2847–2857. [CrossRef]
15. Karim, S.A.; Zamri, N.E.; Alway, A.; Kasihmuddin, M.S.M.; Ismail, A.I.M.; Mansor, M.A.; Hassan, N.F.A. Random satisfiability: A higher-order logical approach in discrete Hopfield Neural Network. *IEEE Access* **2021**, *9*, 50831–50845. [CrossRef]
16. Bazuhair, M.M.; Jamaludin, S.Z.M.; Zamri, N.E.; Kasihmuddin, M.S.M.; Mansor, M.A.; Alway, A.; Karim, S.A. Novel Hopfield Neural Network Model with Election Algorithm for Random 3 Satisfiability. *Processes* **2021**, *9*, 1292. [CrossRef]
17. Alway, A.; Zamri, N.E.; Karim, S.A.; Mansor, M.A.; Kasihmuddin, M.S.M.; Bazuhair, M.M. Major 2 Satisfiability Logic in Discrete Hopfield Neural Network. *Int. J. Comput. Math.* **2021**, *99*, 924–948. [CrossRef]
18. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [CrossRef]
19. Sathasivam, S.; Abdullah, W.A.T.W. Logic learning in Hopfield networks. *Mod. Appl. Sci.* **2008**, *2*, 57–63. [CrossRef]
20. Mansor, M.A.; Sathasivam, S. Accelerating activation function for 3-satisfiability logic programming. *Int. J. Intell. Syst. Appl.* **2016**, *10*, 44–50. [CrossRef]
21. Umair, M.; Shen, Y.; Qi, Y.; Zhang, Y.; Ahmad, A.; Pei, H.; Liu, M. Evaluation of the CropSyst Model during Wheat-Maize Rotations on the North China Plain for Identifying Soil Evaporation Losses. *Front. Plant Sci.* **2017**, *8*, 1667. [CrossRef] [PubMed]

22. To, W.M.; Lai, T.M.; Lo, W.C.; Lam, K.H.; Chung, W.L. The growth pattern and fuel life cycle analysis of the electricity consumption of Hong Kong. *Environ. Pollut.* **2012**, *165*, 1–10. [CrossRef]
23. Boobier, S.; Hose, D.R.J.; Blacker, A.J.; Nguyen, B.N. Machine learning with physicochemical relationships: Solubility Prediction in Organic Solvents and Water. *Nat. Commun.* **2020**, *11*, 5753. [CrossRef] [PubMed]
24. Aribowo, W.; Muslim, S. Long-term electricity load forecasting based on cascade forward backpropagation neural network. *J. Telecommun. Electron. Comput. Eng.* **2020**, *12*, 39–44.
25. Aiewsakun, P.; Simmonds, P. The genomic underpinnings of eukaryotic virus taxonomy: Creating a Sequence-Based Framework for Family-Level Virus Classification. *Microbiome* **2018**, *1*, 38. [CrossRef] [PubMed]
26. Loo, L.H.; Lin, H.J.; Steininger, R.J.; Wang, Y.; Wu, L.F.; Altschuler, S.J. An approach for extensibly profiling the molecular states of cellular subpopulations. *Nat. Methods* **2009**, *10*, 759–765. [CrossRef] [PubMed]
27. Gerrard, W.; Bratholm, L.A.; Packer, M.J.; Mulholland, A.J.; Glowacki, D.R.; Butts, C.P. IMPRESSION—Prediction of NMR parameters for 3-dimensional chemical structures using machine learning with near quantum chemical accuracy. *Chem. Sci.* **2020**, *11*, 508–515. [CrossRef]
28. Lee, C.C.; de Gyvez, J.P. Color image processing in a cellular neural-network environment. *IEEE Trans. Neural Netw.* **1996**, *7*, 1086–1098.
29. Aponte, Y.; Atasoy, D.; Sternson, S.M. AGRP neurons are sufficient to orchestrate feeding behavior rapidly and without training. *Nat. Neurosci.* **2011**, *14*, 351–356. [CrossRef]
30. Strait, C.E.; Blanchard, T.C.; Hayden, B.Y. Reward value comparison via mutual inhibition in ventromedial prefrontal cortex. *Neuron* **2014**, *82*, 1357–1366. [CrossRef]
31. Li, Q.; Chen, H.; Huang, H.; Zhao, X.; Cai, Z.; Tong, C.; Liu, W.; Tian, X. An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis. *Comput. Math. Methods Med.* **2017**, *2017*, 9512741. [CrossRef] [PubMed]
32. Tran-Ngoc, H.; Khatir, S.; De Roeck, G.; Bui-Tien, T.; Nguyen-Ngoc, L.; Abdel Wahab, M. Model Updating for Nam O Bridge Using Particle Swarm Optimization Algorithm and Genetic Algorithm. *Sensors* **2018**, *12*, 4131. [CrossRef] [PubMed]
33. Emami, H.; Derakhshan, F. Election algorithm: A New Socio-Politically Inspired Strategy. *AI Commun.* **2015**, *28*, 591–603. [CrossRef]
34. Sun, Y.; Wang, Z.; Van Wyk, B.J. Chaotic Hopfield neural network swarm optimization and its application. *J. Appl. Math.* **2013**, *2013*, 873670. [CrossRef]
35. Muraoka, K.; Chaikittisilp, W.; Okubo, T. Multi-objective de novo molecular design of organic structure-directing agents for zeolites using nature-inspired ant colony optimization. *Chem. Sci.* **2020**, *31*, 8214–8223. [CrossRef]
36. Kalathil, S.; Elias, E. Non-uniform cosine modulated filter banks using meta-heuristic algorithms in CSD space. *J. Adv. Res.* **2015**, *6*, 839–849. [CrossRef]