mathematics

MDPI

Article

Discrete Mutation Hopfield Neural Network in Propositional Satisfiability

Mohd Shareduwan Mohd Kasihmuddin [1], Mohd. Asyraf Mansor [2,*], Md Faisal Md Basir [3] and Saratha Sathasivam [1]

[1] School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800 USM, Malaysia; shareduwan@usm.my (M.S.M.K.); saratha@usm.my (S.S.)
[2] School of Distance Education, Universiti Sains Malaysia, Penang 11800 USM, Malaysia
[3] Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, Johor Bahru 81310 UTM, Johor, Malaysia; mfaisalmbasir@utm.my
* Correspondence: asyrafman@usm.my; Tel.: +60-46535906

Received: 21 September 2019; Accepted: 16 October 2019; Published: 19 November 2019

Abstract: The dynamic behaviours of an artificial neural network (ANN) system are strongly dependent on its network structure. Thus, the output of ANNs has long suffered from a lack of interpretability and variation. This has severely limited the practical usability of the logical rule in the ANN. The work presents an integrated representation of *k*-satisfiability (*k*SAT) in a mutation hopfield neural network (MHNN). Neuron states of the hopfield neural network converge to minimum energy, but the solution produced is confined to the limited number of solution spaces. The MHNN is incorporated with the global search capability of the estimation of distribution algorithms (EDAs), which typically explore various solution spaces. The main purpose is to estimate other possible neuron states that lead to global minimum energy through available output measurements. Furthermore, it is shown that the MHNN can retrieve various neuron states with the lowest minimum energy. Subsequent simulations performed on the MHNN reveal that the approach yields a result that surpasses the conventional hybrid HNN. Furthermore, this study provides a new paradigm in the field of neural networks by overcoming the overfitting issue.

Keywords: Mutation Hopfield Neural Network; Hopfield neural network; *k*-satisfiability

1. Introduction

In recent years, the high dimensional neural network has been developed by researchers in various mathematical fields. Although many optimization approaches have been proposed to achieve a global solution, several problems have arisen associated with overfitting and lack of solution variation. In another development, the artificial neural network (ANN) approach provides feasible solutions to useful optimization problem such as Very Large-Scale Integration (VLSI) [1], pattern recognition [2], image processing [3], classification problems [4] and knowledge discoveries [5]. Inspired by the biological human brain, the Hopfield neural network (HNN) was originally proposed by Hopfield and Tank [6] to solve optimization problems. Due to the simplicity of the network, the HNN has attracted much attention [7,8]. The HNN utilizes a dynamic system in which the possible solution of the HNN will be reduced to a minimum Lyapunov energy function. In this case, if the solution achieves the lowest minimum energy, the solution is likely to be optimal. Although the energy function always converges to minimum energy, neurons oscillate with the same energy and are ultimately trapped in the local minima [9]. The main disadvantages of the HNN are the storage capacity problem [10] and that it is easily trapped to the local minimum solution [11]. Motivated by these weaknesses, researchers have proposed various hybrid systems to increase the accuracy and stability of the HNN. Silva and

Bastos-Filho [7] proposed the hierarchical HNN to reduce the network structure and mitigate the convergence problem of the normal HNN. The proposed hybrid HNN increased the storage capacity issue in solving various practical applications. Yang et al. [12] proposed a hybrid metaheuristic with the HNN to solve the assignment problem. In this study, a hybrid genetic algorithm with HNN managed to overcome the problem constraints by searching for high quality solutions with the minimum possible cost. Jayashree and Kumar [8] proposed a gravitational search algorithm with a genetically optimized HNN to diagnose diabetes patients. The proposed method is able to create a diabetic expert system which consists only of induced feature interpretation. In terms of quality of the neuron states, Zhang et al. [13] proposed an impulsive and switching HNN by using the B-equivalence method. The proposed method has enhanced global stability by altering the state of the neuron. In addition, gradient descent learning has been used by Kobayashi [14] in a quaternionic HNN. In his work, proposed gradient descent learning incorporated with an activation function outperformed a projection rule in noise tolerance in a computer simulation. The common denominator in these studies is the implementation of the optimal learning method in the HNN.

The estimation distribution algorithm (EDA) has been applied to many optimization problems. This algorithm generally chooses some high fitness solution from the current population to form the parent population. Hence, the EDA implements a probability model from the parent population and the next generation is sampled from this probabilistic model [15,16]. The EDA has been widely introduced in various practical applications, such as the job scheduling problem [17], the economic dispatch problem [18], traffic modelling [19] and species modelling [20]. Wang [21] proposed an interesting hybrid HNN with the EDA. In the proposed hybrid network, the neuron state that achieved local minimum energy is perturbed based on the EDA. The act of perturbation for the neuron state will generate the new starting point for the HNN and reduce the possible local minimum solution. Hu et al. [22] proposed the mutation hopfield neural network (MHNN) to solve max-cut and aircraft landing schedule (ALS) problems. In this work, the proposed model utilized the EDA to explore other solution spaces that fulfil the given cost function. Unfortunately, most EDAs have focused on finding the solution to the problem instead of creating a learning model for the problem.

Representing the nondeterministic polynomial time (NP) problems by reducing them to a proportional satisfiability (SAT) logical rule is a powerful strategy that is widely used in a range of research disciplines and to tackle various industrial problems. Recently, SAT representation has been proposed in mathematical formulation [23,24], model checking [25], protein bioinformatics [26], social networking [27], disease detection [28], ANN [29] and many industrial automations [30]. In order to create an intelligent system, SAT is an important language in propositional logic. The introduction of SAT as propositional logic in the ANN is a step towards understanding human intelligence. Propositional logic has been embedded in the HNN as a single intelligent unit [31]. The proposed method was further implemented with Horn clauses [32]. The developed model is able to achieve more than 90% global minimum energy. In another development, the proposed model utilized the unbounded McCulloch–Pitts (MCP) neuron to retrieve the final state of neurons. Although the MCP neuron helps the network to converge to global minima, it was shown by some researchers [33] that the MCP neuron is prone to a number of weaknesses, such as computational burdening and lack of efficiency when the complexity of the logic program increased. Sathasivam [34] upgraded the MCP neuron by incorporating the effective relaxation method in generating the optimum final neuron states. Such realization lead Sathasivam [35] to develop a stochastic energy distribution method in determining the neuron state. This proposed approach reduces the neuron oscillations during the retrieval phase deployed by the HNN. Mansor and Sathasivam [36] and Kasihmuddin and Sathasivam [37] proposed unipolar and bipolar activation functions, respectively, to reduce neuron oscillation during the retrieval phase. Yoon and Lee [38] utilized the sub-planner algorithm in reducing the local minima, which indicated the possibility of escaping the local minima in the HNN. Velavan et al. [39] constructed a feasible method using mean field theory (MFT) to reduce the local minima in the HNN. Alzaeemi and Sathasivam [40] used the kernel method in the HNN as a systematic classification

of the state of neurons. The proposed method manages to achieve 99% accuracy in terms of global minimum energy. To this end, the implementation of the mentioned work only manages to retrieve the correct and redundant neuron states.

Unfortunately, there has been no recent effort to discover other possible neuron states in the solution space. A detailed comparison of different retrieval phases may reveal better possibilities in terms of theoretical assumptions, modelling frameworks and computational processes. Thus, the contributions of the present paper are: (1) A novel mutation hopfield neural network (MHNN) is proposed by implementing the EDA into the Hopfield neural network. (2) Implementation of propositional satisfiability in the MHNN and determination of the effectiveness of the retrieval properties of the proposed MHNN. (3) A comprehensive comparison of the MHNN with five established HNN models with two different retrieval models. By constructing an effective ANN work model, the proposed network will be beneficial in finding the correct approximate solution for various mathematical formulations, such as Runge–Kutta, the Lie group integrator and control theory. Our results showed that the proposed MHNN displays the best performance in terms of propositional satisfiability compared to other well-established existing work.

2. Propositional Satisfiability

Satisfiability (SAT) can be defined as a logical rule that comprises clauses containing variables. The three components of general SAT problems are summarized as follows:

1. Consist of a set of l variables, $x_1, x_2, \ldots, x_l$, where $x_i \in \{1, -1\}$. All the variables in the clause will be connected by logical OR $(\vee)$.
2. A set of literals. A literal is a variable or a negation of variable.
3. A set of $\boldsymbol{n}$ distinct clauses, $C_1, C_2, \ldots, C_n$. Each clause consists of only literals combined by logical AND $(\wedge)$.

Generalized k-satisfiability (kSAT) has demonstrated the ability to represent real life applications. Thus, [41] has noted that generalized kSAT can be reduced to maximum satisfiability (MAX2SAT), and the 2SAT logical rule is a step toward exploring an ANN based on the MAX2SAT logical rule. The general formula of 2SAT is given as:

$$P_{2SAT} = \wedge_{i=1}^{n} C \tag{1}$$

where P_{2SAT} is a 2-satisfiability logical rule that consists of clause C_i given as:

$$C_i = \vee_{j=1}^{2} (x_{ij}, y_{ij}) \tag{2}$$

with l variables and $\boldsymbol{n}$ clauses denoted by $F_k(n, m)$. In this case, F_k is a Conjunctive Normal Form (CNF) formula where the clauses are chosen uniformly, independently and without replacement among all $2^k \begin{pmatrix} n \\ k \end{pmatrix}$ non-trivial clauses of length k. Note that variable $\boldsymbol{x}$ occurs in a clause, if the clause contains either $\boldsymbol{x}$ or $\neg x$. A mapping $\alpha : V(F_k) \rightarrow \{-1, 1\}$ is called logical interpretation. If α maps all variables to a Boolean value, it is considered complete. A true literal is a literal that evaluates true under the given interpretation. A given clause is considered satisfied if it has at least one true literal under the given interpretation and falsified if it has no true literal under the given interpretation. The comprehensive summary of the implementation of propositional satisfiability is given in Figure 1.

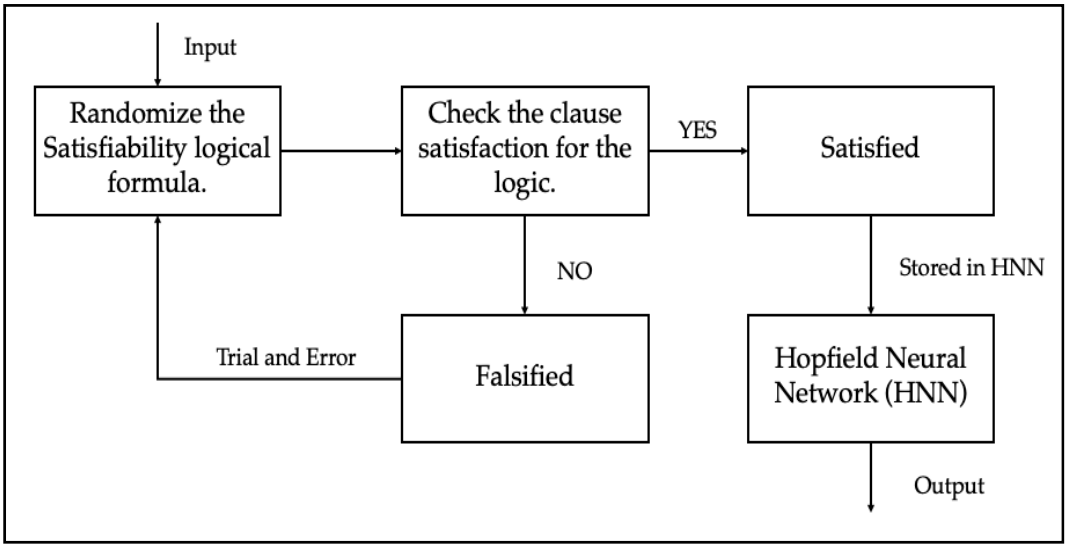

Figure 1. Summary of propositional satisfiability.

3. Discrete Hopfield Neural Network

The usage of the Hopfield neural network (HNN) in solving various NP problems was proposed by Hopfield and Tank [6]. The first task of the newly introduced HNN demonstrated the computational power in solving travelling salesman and circuit problems. The HNN comprises N interconnected neurons, each described by an Ising spin variable [42,43]. Conventional neuron update in the HNN is as follows:

$$S_i = \begin{cases} 1 & \text{if } \sum_j W_{ij} S_j > \psi_i \\ -1 & \text{Otherwise} \end{cases} \tag{3}$$

where W_{ij} is the synaptic weight from unit j to i, S_j is the state of unit j and ψ_i is the pre-defined threshold of unit i. Several studies [43] have defined $\psi_i = 0$ to ensure the energy of the network decreases monotonically. Each time a neuron is connected with W_{ij}, the value of the connection will be stored in an interconnection matrix where $W^{(2)} = \left[W_{ij}^{(2)}\right]_{n\times n}$ and $W^{(1)} = \left[W_{i}^{(1)}\right]_{n\times n}$ for N dimensional column vectors $\psi = (\psi_1, \psi_2, \psi_3, \ldots, \psi_N)^T$. As indicated in [44], the constraint of matrix $W^{(1)}$ and $W^{(2)}$ does not permit self neuron connection, $W_{ii}^{(2)} = W_{jj}^{(2)} = W_{kk}^{(2)} = 0$, and symmetrical neuron connection, $W_{ij}^{(2)} = W_{ji}^{(2)}$. The simple features of the HNN, such as fault tolerance [6] and content addressable memory [11], make it suitable for integration of propositional satisfiability. The HNN utilizes the usage of logical rules to instruct the behaviour of the network using the synaptic weight (neuron connection). In this case, the logical formula consists of variables represented in terms of N neurons. The implementation of the 2SAT logic rule in the HNN is abbreviated HNN-2SAT and the primary aim of the network is to reduce logical inconsistencies by minimizing the cost function of the network. The cost function E_P of the logic rule in the HNN is given by:

$$E_{P_{2SAT}} = \sum_{i=1}^{NC} \prod_{j=1}^{NV} L_{ij} \tag{4}$$

where NC and NV denote the number of clauses and variables, respectively. The inconsistencies of logic clause L_{ij} is given as follows:

$$L_{ij} = \begin{cases} \frac{1}{2}(1 - S_x), & if\ \neg x \\ \frac{1}{2}(1 + S_x), & Otherwise \end{cases} \tag{5}$$

The synaptic weight represents the connection between the variable and the clauses in the logical formula. In order to calculate the synaptic weight of the HNN, Abdullah [41] described the most straightforward measure by comparing $E_{P_{2SAT}}$ with the final energy function $H_{P_{2SAT}}$. Denoting the state of neuron for each variable in the logic rule S_i at time t as $S_i(t)$, the local field of the network can be represented as follows:

$$h_i(t) = \sum_{j=1, i \neq j \neq k}^{N} W_{ij}^{(2)} S_j + W_i^{(1)} \tag{6}$$

$$S_i(t+1) = \begin{cases} 1, & \sum\limits_{j=1, i \neq j}^{N} W_{ij}^{(2)} S_j + W_i^{(1)} \geq 0 \\ -1, & \sum\limits_{j=1, i \neq j}^{N} W_{ij}^{(2)} S_j + W_i^{(1)} < 0 \end{cases} \tag{7}$$

Equations (6) and (7) guarantee the energy will decrease monotonically with the network. The final energy of the HNN is given by:

$$H_{P_{2SAT}} = -\frac{1}{2} \sum_{i=1, i \neq j}^{N} \sum_{j=1, i \neq j}^{N} W_{ij}^{(2)} S_i S_j - \sum_{i=1}^{N} W_i^{(1)} S_j \tag{8}$$

In this case, the updating rule proposed in [6] is as follows.

Theorem 1. *Let W be a symmetric matrix with a non-negative diagonal. The neurons in the discrete HNN $N = (W, T)$ that operate in asynchronous mode will always converge to a stable state. Under this circumstance, the HNN is considered stable.*

The synaptic weight of HNN-2SAT is always symmetrical. Hence the final neuron state of the proposed HNN will converge to minimum energy. In this approach, the energy value of each logical rule embedded to the HNN, $H_{P_{2SAT}}$ is used to separate local minimum and global minimum solutions. Global minimum energy, $H_{P_{2SAT}}^{\min}$, can be pre-calculated because the total magnitude of the energy that corresponds to any logic clause is always a constant [35,45]. The implementation of the discrete Hopfield neural network is summarized in the block diagram as shown in Figure 2.

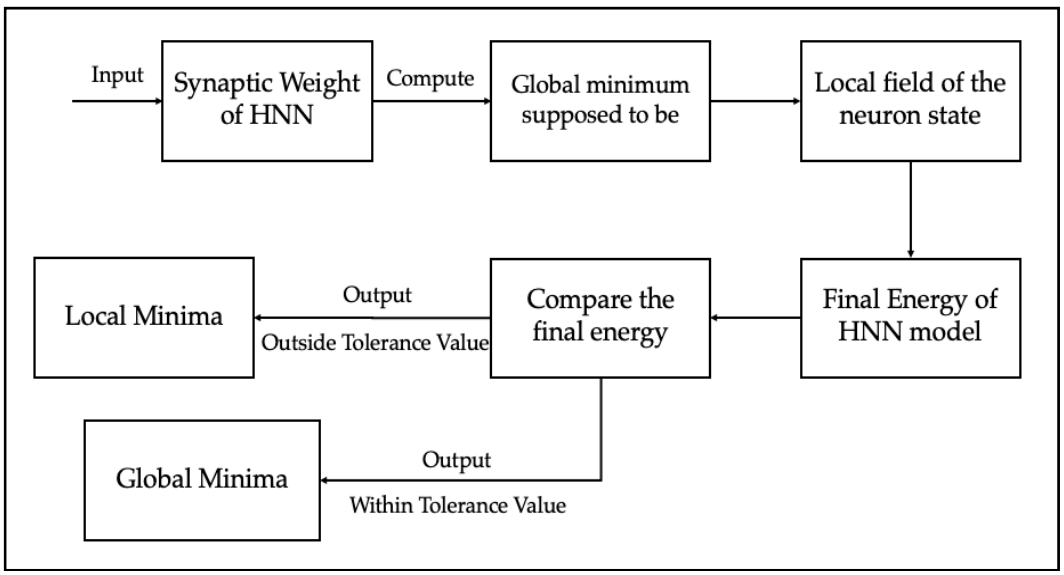

Figure 2. Summary of discrete Hopfield neural network (HNN).

4. Mutation Hopfield Neural Network

The conventional HNN has a high chance of being trapped in local minima as the number of neurons involved increases. For decades, several scholars have incorporated genetic algorithm (GA) into the HNN [8,46,47] to increase the searching capability of the HNN. However, the usage of genetic algorithm during the retrieval phase of the HNN is considered inefficient due to the complexity of several operators, such as crossover and mutation. These two operators require more generations/iterations before the apparent improvement can take place [22]. Several studies indicate [48–50] that GAs entail a higher complexity, leading to premature convergence. Recently, the EDA was proposed to avoid premature convergence of the system [16,51,52]. The EDA utilizes a probability model to learn and sampling to optimize the entire swarm. In this case, the crossover and mutation operators are not required in the EDA. The complete implementation of the EDA in the HNN is as follows:

Step 1: The input of the HNN, $S_i = (S_{i1}, S_{i2}, \ldots\ldots, S_{iN})$ is randomly initialized.

Step 2: The output $h_i = (h_1, h_2, h_3, \ldots, h_N)$ of the HNN for every S_i, $i = 1, 2, \ldots, n$ is computed using Equation (6). When the HNN converges to a single stable state, the output of the HNN will be mutated with a univariate marginal Gaussian distribution probability model [52].

$$h_i^{M_i} = \frac{1}{\sqrt{2\pi{\sigma_i}^2}} \exp\left(-\frac{(h_i - \mu_i)^2}{2{\sigma_i}^2}\right) \tag{9}$$

where μ and σ_i^2 are defined as:

$$\mu_i = \frac{1}{N}\sum_{i=1}^{N} h_i \tag{10}$$

$$\sigma_i^2 = \frac{1}{N-1}\sum_{i=1}^{N} (h_i - \mu_i)^2 \tag{11}$$

The mutation creates a minor perturbation to the neuron and helps the neuron to escape the current stable state of the HNN.

Step 3: Obtain the new state $S_i^{M_i}$ by using $h_i^{M_i}$. Check the solution fitness using the following equation:

$$f_i = \sum_{i=1}^{NC} C_i \tag{12}$$

where C_i is defined as follows:

$$C_i = \begin{cases} 1, & True \\ 0, & False \end{cases} \tag{13}$$

The best solution of $S_i^{M_i}$ will be updated.

Step 4: New output $S_i^{M_i}$ based on h_i^M and $S_i^{M_i}$ will be retained.

Step 5: Steps 2–5 are repeated until termination criteria are satisfied. The best individual is chosen.

In this paper, the implementation of the EDA in the HNN is named the MHNN. The MHNN will be embedded with the logic rules stated in Equations (1) and (2). The primary aim of the MHNN is to increase the variation of logic rules produced by neurons and explore more global minimum energy in the search space.

5. HNN Model Performance Evaluation

To test the effectiveness of the proposed method, the performance of all HNNs will be evaluated based on error analysis, energy analysis and similarity analysis of the retrieval neurons. The equations

for root mean squared error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) analysis are as follows:

$$\text{RMSE} = \sum_{i=1}^{NC} \sqrt{\frac{1}{n}\left(H_{P_{2SAT}}^{\min} - H_{P_{2SAT}}\right)^2} \tag{14}$$

$$\text{MAE} = \sum_{i=1}^{NC} \frac{1}{n}\left|H_{P_{2SAT}}^{\min} - H_{P_{2SAT}}\right| \tag{15}$$

$$\text{MAPE} = \sum_{i=1}^{NC} \frac{100}{n} \frac{\left|H_{P_{2SAT}}^{\min} - H_{P_{2SAT}}\right|}{\left|H_{P_{2SAT}}^{\min}\right|} \tag{16}$$

$$Zm = \frac{1}{tc}\sum_{i=1}^{NN} N_{H_{P_{2SAT}}} \tag{17}$$

where the global minimum energy H_i^P is given in Equation (6) and $\left|H_{P_{2SAT}}^{\min} - H_{P_{2SAT}}\right| \leq \partial$. In general, the bipolar state of the retrieved neuron represents the interpretation of the logic rule. For similarity analysis, the key component of analyzing the final state of the neuron is from comparing the retrieved state with a benchmark neuron state. The benchmark state is defined as the ideal neuron state retrieved from the HNN model. The benchmark neuron state is given as follows:

$$S_i^{\max} = \begin{cases} 1 & , A \\ -1 & , \neg A \end{cases} \tag{18}$$

where A and $\neg A$ are positive and negative literals in the logic clause, respectively. Based on Equation (17), if the logic rule reads $P = (A \vee \neg B \vee \neg C) \wedge (D \vee E \vee \neg F)$, the benchmark state of the neuron is given as $S_A^{\max} = 1$, $S_B^{\max} = -1$, $S_C^{\max} = -1$, $S_D^{\max} = 1$, $S_E^{\max} = 1$, $S_F^{\max} = -1$ or $S_i^{\max} = (1, -1, -1, 1, 1, -1)$. Notably, the final energy of $S_i^{\max}$ is always a global minimum solution or $\left|H_{P_{S_i^{\max}}} - H_{P_{S_i^{\max}}}^{\min}\right| \leq \partial$ [22]. ∂ is a tolerance value for energy difference in the HNN. By using a benchmark state, the variation of the HNN model is formulated as follows:

$$V = \sum_{i=0}^{\lambda} F_i \tag{19}$$

$$F_i = \begin{cases} 1 & , S_i \neq S_i^{\max} \\ 0 & , S_i = S_i^{\max} \end{cases} \tag{20}$$

where λ is a total solution produced by the HNN model and F_i is a scoring mechanism to evaluate the difference between the benchmark state and the final state of neuron. Since most of the neuron states retrieved in the HNN achieve global minimum energy [53], $S_i^{\max}$ is a perfect benchmark state in comparing the final states of different HNN models. In this section, the final neuron state retrieved that corresponds to the 2SAT logic rule will be analyzed using similarity metrics. Despite its mathematical simplicity, several similarity metrics were implemented to identify which HNN model maximized the value of V. In particular, instead of comparing logic with logic, the comparison will be made based on the individual neuron state. Hence, the general comparison between benchmark state and the final neuron state is as follows:

$$C_{S_i^{\max} S_i} = \left\{ \left(S_i^{\max}, S_i\right) \middle| i = 1, 2, \ldots, n \right\} \tag{21}$$

The further specifications of the variables are defined as follows:

l is the number of $(S_i^{\max}, S_i)$ where both elements have the value 1 in $C_{S_i^{\max} S_i}$;

m is the number of $(S_i^{\max}, S_i)$ where $S_i^{\max}$ is 1 and S_i is -1 in $C_{S_i^{\max} S_i}$;
n is the number of $(S_i^{\max}, S_i)$ where $S_i^{\max}$ is -1 and S_i is 1 in $C_{S_i^{\max} S_i}$;
o is the number of $(S_i^{\max}, S_i)$ where both elements have the value -1 in $C_{S_i^{\max} S_i}$.

Note that the size of the neuron string is given as $n = l + m + n + o$. Using the above information, the similarity coefficient for all HNN models is summarized in Table 1.

Table 1. Similarity coefficient for neuron state.

No	Similarity Coefficient	Similarity Representation $(S_i^{\max}, S_i)$
1	Jaccard's Index [54]	$J(S_i^{\max}, S_i) = \frac{l}{l+m+n}$
2	Sokal Sneath 2 [55]	$SS(S_i^{\max}, S_i) = \frac{l}{l+2(m+n)}$
3	Dice [56]	$D(S_i^{\max}, S_i) = \frac{2l}{2l+m+n}$

6. Simulation

To further investigate the performance of the proposed model for propositional logic, the proposed MHNN was compared with the standard Hopfield neural network (HNN), the kernel Hopfield neural network (KHNN), the hybrid Boltzman Hopfield neural network (BHNN) and the Hopfield neural network incorporated with mean field theory (MFTHNN). The simulation dynamic is divided into two parts. Firstly, the performance of all hybrid models will be implemented without internal noise on the retrieval phase. In this case, the HNN will retrieve the final state of neurons by following the network dynamic proposed by Sathasivam [34] and Mansor et al. [35]. The full flowchart of HNN is enclosed in the Appendix A, specifically in Figure A1. S_i will be updated using Equations (6) and (7) and the final energy will be computed. Secondly, each HNN model will be incorporated with a noise function [57] given as follows:

$$h_i(t+1) = h_i(t) + \beta(t) \tag{22}$$

where h_i is a local field of the network and $\beta(t)$ is the noise function incorporated in every HNN model. In this experiment, the noise function will be implemented for the range $\beta(t) \in [-0.05, 0.05]$. In both simulations, the quality of the retrieval phase for the HNN model will be tested based on RMSE, MAE and MAPE. Similarity analysis will be conducted to identify different global minimum solutions produced by the HNN model. The HNN with a different initial neuron state might contribute to the biasedness of the retrieval state because the network simply memorizes the final state without producing a new state [58]. In this respect, possible positive and negative bias can be reduced by generating all the neuon states randomly:

$$(S_1, S_2, S_3, \ldots\ldots, S_N) = \begin{cases} 1 & , rand(0,1) < 0.5 \\ -1 & , otherwise \end{cases} \tag{23}$$

where the states of true and false are given as 1 and -1, respectively. In this case, the simulated dataset will be obtained by generating random clauses and literals for each 2SAT logic rule. The parameters for each HNN model are listed in Tables 2–6.

Table 2. List of Parameters in MHNN.

Parameter	Parameter Value
Neuron Combination	100
Tolerance Value (∂)	0.001
Number of Learning (Ω)	100
No_Neuron String	100
Selection_Rate	0.1
Mutation Rate	0.01

Table 3. List of Parameters in HNN [49].

Parameter	Parameter Value
Neuron Combination	100
Tolerance Value (∂)	0.001
Number of Learning (Ω)	100
No_Neuron String	100
Selection_Rate	0.1

Table 4. List of Parameters in KHNN [38].

Parameter	Parameter Value
Neuron Combination	100
Tolerance Value (∂)	0.001
Number of Learning (Ω)	100
No_Neuron String	100
Selection_Rate	0.1
Type of Kernel	Linear Kernel

Table 5. List of Parameters in BHNN [34].

Parameter	Parameter Value
Neuron Combination	100
Tolerance Value (∂)	0.001
Number of Learning (Ω)	100
No_Neuron String	100
Selection_Rate	0.1
Temperature *(T)*	70

Table 6. List of Parameters in MFTHNN [37].

Parameter	Parameter Value
Neuron Combination	100
Tolerance Value (∂)	0.001
Number of Learning (Ω)	100
No_Neuron String	100
Selection_Rate	0.1
Temperature *(T)*	70
Activation Function	Hyperbolic Tangent (HTAF)

The mutation rate in our proposed model, MHNN, plays a prominent role in determining the nature of final neuron states. If the mutation rate is more than 0.1, the neuron tends to converge to one state only, $S_i \rightarrow a$, $a = \pm 1$. Meanwhile, if the mutation rate is between 0.05 and 0.1, the neuron will undergo state oscillation that will produce local minima solutions $E_{P_{2SAT}} \neq 0$ or $\left| H_{P_{S_i^{\max}}} - H^{\min}_{P_{S_i^{\max}}} \right| \leq \partial$. On the contrary, the effect of the mutation in the HNN can be seen to be significant when the mutation rate is within 0.01 until 0.04. Thus, the ideal mutation rate was chosen to be 0.01 to effectively investigate the impact of a mutation in the HNN. The choice of mutation rate has good agreement with [22]. A non-common parameter such as T was utilized in the BHNN and MFTHNN as the simulated annealing effect takes place in both models. Theoretically, if we select the temperature, $T > 75$, the neuron states tend to oscillate and affect the final states of the neuron attained after the simulation. On the contrary, if $T < 70$, the effect of simulated annealing will vanish, and the network will be reduced to the ordinary HNN, as proposed by [34]. Therefore, the value of T was selected according to [35] for the BHNN and [39] for the MFTHNN. According to Table 4, the linear kernel is applied due to the good agreement with the logic programming problem as outlined in the work of [40]. Based on Table 6, the

hyperbolic tangent (HTAF) was selected due to the differentiable nature of the function and the ability to establish the non-linear relationship among the neuron connections [42]. The comparison among the MHNN, HNN, KHNN, BHNN, and MFTHNN was made on the same basis, which is utilized in propositional satisfiability problems. Thus, the important parameters, including the non-common parameter, must comply with the previous studies dealing with the similar problem.

All HNN models were implemented in Dev C++ Version 5.11 in Windows 10 (Bloodshed), using an Intel Core i3 with 1.7 GHz processor. To make an impartial comparison, all HNN models were terminated after being executed for not more than 24 h. All models were stopped after the computation time reached 24 h. In order to make an overall comparison, the original exhaustive search method was deployed during the learning phase of the HNN model. The learning model remained constant and the results only measured the influence of the retrieval property.

7. Results and Discussion

In this experiment, the retrieval property for the MHNN model against the other existing HNN models in the literature will be investigated. The simulation will be divided into two parts. First, the restricted learning model of propositional logic [59] will be used to examine the accuracy and stability of the network. Secondly, a non-restricted learning model will be used to examine the quality of the solution produced by the HNN models [34]. This is an interesting question since relying on a single learning model does not provide an actual performance measure of the proposed model. Hence, the main contribution of our work is to show the effectiveness of the MHNN in outperforming the existing HNN models.

In most studies of the HNN, the quality of the neuron state is not well assessed since the focus is to attain the final state. For instance, the studies of Sathasivam [35] and Velavan et al. [39] achieved global minimum energy 98% of the time but the quality of the final neuron state was not effectively measured. In this section, learning iteration will be restricted to $NH \leq \Omega$. In this case, learning in the HNN will terminated when $NH = \Omega$. Hence, all the HNN models exhibit the same learning phase via the Wan Abdullah method [31]. According to Figures 3–6, for $0 \leq NN \leq 60$, the MHNN is the best model in terms of Zm, RMSE, MAE and MAPE. Although the MHNN underwent restricted iteration during the learning phase, the MHNN is still able to locate a state that leads to global minimum energy $\left| H_{P_{S_i^{\max}}} - H^{\min}_{P_{S_i^{\max}}} \right| \leq \partial$. In this case, the MHNN only requires a fragment of correct synaptic weight to retrieve the optimal final state during the learning phase. A similar perturbation strategy was utilized by Hu et al. [22] in solving the max-cut problem. On the other hand, the EDA in the MHNN creates minor neuron oscillations and retrieves the state independently, although the network trained the suboptimal synaptic weight. Hence, the synergistic property of the learning phase of the HNN and the EDA reduces the number of local minima. The neuron perturbation reduces the effect of the suboptimal synaptic weight during the learning phase. Other HNN models, such as the BHNN [35] and MFTHNN [40], reduced the number of local minima but failed to achieve optimal global minimum energy as the number of clauses increased. According to Figures 4–6, the conventional HNN has the highest error at $NN = 60$ because the network retrieved the suboptimal synaptic weight. Structurally, the conventional HNN has no second optimization layer and focuses solely on the MCP neuron in retrieving the final state of neurons [60]. Similar to the BHNN and MFT, the suboptimal synaptic weight reduces the effectiveness of the Boltzmann machine in retrieving the correct final state. In addition, MHNN is minimally affected when more negative or positive noise is added to the retrieval phase. The EDA in the MHNN independently locates several possible neuron states that leads to a global minimum solution. In addition, at $NN \geq 20$, the conventional HNN failed to obtain at least 10% global minimum energy and the learned synaptic weight has no effect on the retrieval phase. Although the BHNN and MFTHNN utilized energy to overcome the barrier of local minima, the energy from the temperature increment does not compensate for the incorrect synaptic weight. Although the direction of the Boltzmann machine in the BHNN and MFTHNN can be improved, as seen in other

work such as [61], it can be argued that it will create more unnecessary free parameters. Additional optimization algorithms, such as a metaheuristics algorithm, is required to find the optimal value of the free parameter. In addition, low values of *Zm* (refer to Figure 3) accompanied by high values of RMSE, MAE and MAPE (refer Figures 4–6) were recorded in the KHNN. Unfortunately, the kernel function in the KHNN is largely dependent on the quality of the neuron state retrieved. In fact, at $NN = 20$, more than 50% of the states retrieved were trapped at local minima. Similar to other HNN models, a suboptimal synaptic weight also reduces the effectiveness of the kernel function in the KHNN by retrieving the final state that contributes to inconsistent interpretation. As can be seen, the MHNN increases the fault tolerance property of the conventional HNN model and is comparatively different from other metaheuristics algorithms, such as the genetic algorithm and artificial bee colony, that focus on gradual solution improvement. Gradual solution improvement requires several layers of optimization, such as local and global search operators. The implementation of multiple layers in the metaheuristic algorithm will increase the complexity of the HNN. Additional random parameters are required to avoid premature convergence. In addition, this simulation only takes into account the effect of the retrieval property of HNN models due to incorrect synaptic weight. In this context, all HNN models utilized a conventional exhaustive search method during the learning phase and computation time for all HNN model was not considered.

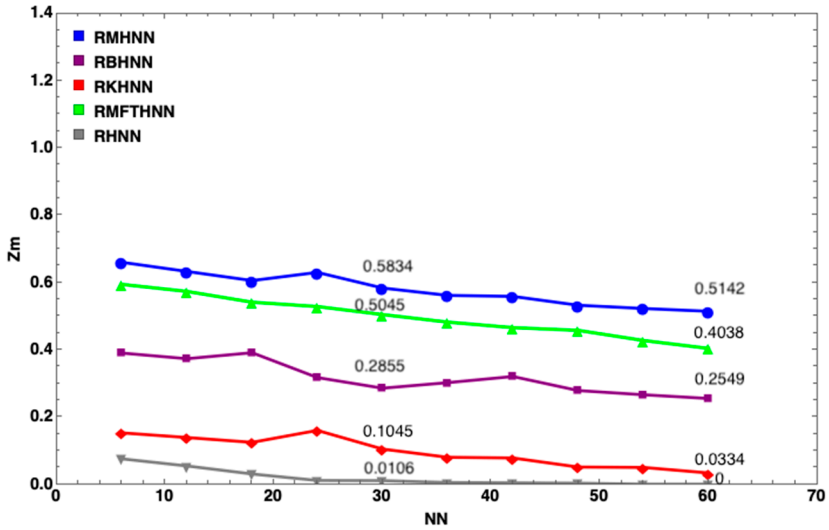

Figure 3. Global minima ratio (*Zm*) of HNN models in restricted learning.

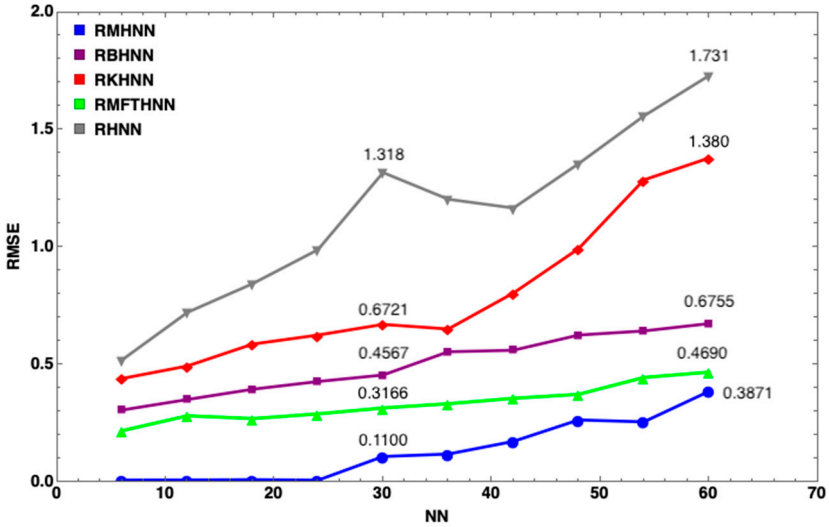

Figure 4. RMSE of HNN models in restricted learning.

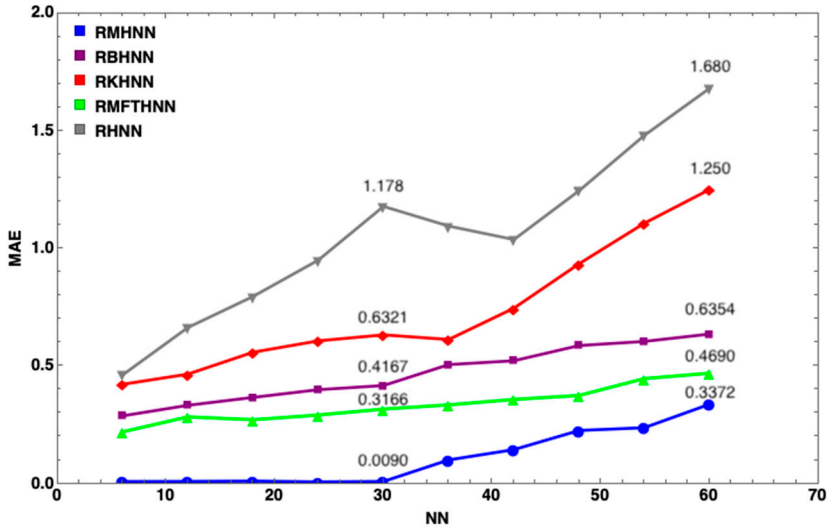

Figure 5. MAE of HNN models in restricted learning.

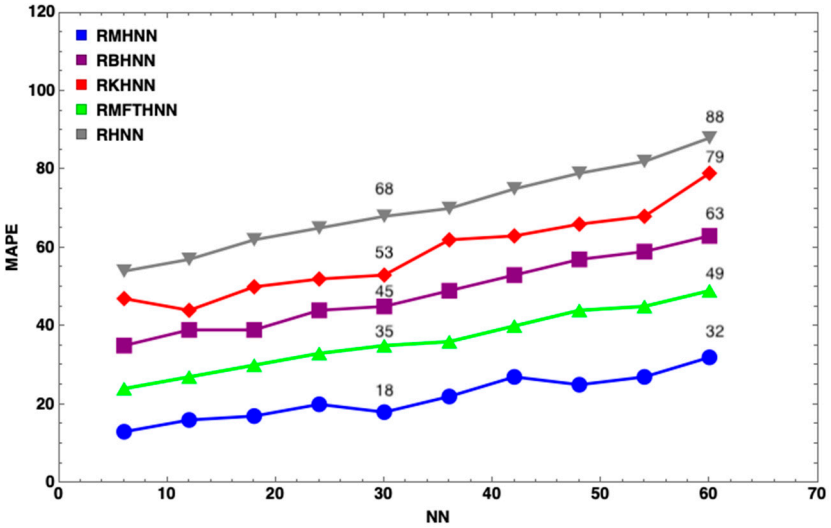

Figure 6. MAPE of HNN models in restricted learning.

In this section, non-restricted learning will be implemented in all HNN models. Learning will iterate until the consistent interpretation is obtained $NH \rightarrow (E_P = 0)$. Note that the similarity indexes only evaluate the neuron state that achieves the global minimum solution. High similarity values signify low variation value in generating the final state of the neuron for the HNN model. According to Figures 7–11, the MHNN has the lowest index value for Jaccard, Socal Sneath 2 and Dice compared to other HNN models in a given benchmark state. The lower value of similarity index for the MHNN was supported by a high value of variability. With the same amount of global minimum energy, the MHNN generated more different neuron final states compared to other HNN models. Figures 11–16 illustrate the performance error of HNN models in terms of RMSE, MAE and MAPE with and without $\beta(t)$. It can be observed from these figures that the learning error (in RMSE, MAE and MAPE) increase rapidly as the number of neuron exceeds 10. The conventional HNN has the worst performance, as 83% of the final states produced contain $E_P \neq 0$. A similar pattern is reported in energy analysis for all HNN models (in Zm). It is apparent that an improvement in Zm is achieved using the MHNN in comparison with other HNN model. For instance, the Zm value for the BHNN, MFTHNN and KHNN reduced dramatically to leass than 50% as the number of $NN \geq 20$. In this case, the retrieval power of other HNN models reduces because the introduction of $\beta(t)$ as an extra bias to Equation (6) increases the probability of the local field to achieve the suboptimal state. According to Figure 17, the conventional HNN has the lowest

variability value compared to other HNN models. The Boltzmann component in both the BHNN and MFTHNN showed similar retrieval patterns which are relatively low compared to the MHNN. $\beta(t)$ is observed to reduce the effectiveness of the BHNN and MFTHNN (low Zm value) and the global solution retrieved has lower variation value. In this case, the BHNN and MFTHNN were only effective if the desired solution space is small. A similar variation pattern is reported in the KHNN. Although several studies [58] showed the beneficial effect of the noise in an ANN, $\beta(t)$ contributed a minimal impact in retrieving more global solutions in HNN models. On the other hand, all the existing HNN models failed to achieve a 70% variation value and explored less than 50% of neuron configurations. The energy penalty for the evaluation of the HNN model is given in Figure 16, where the global minimum energy will be penalized if the state is $S_i^{\max} = S_i$. From Figure 18, it is clear that:

1. The MHNN has the lowest energy penalty value compared to other HNN models
2. With the same number of neurons, such as $NN = 60$, the energy penalty of the HNN has the largest value, followed by the KHNN, BHNN and HNN, indicating that the EDA has the significant effect on the performance of the MHNN.
3. $\beta(t)$ has little impact on the MHNN in terms of the energy penalty.

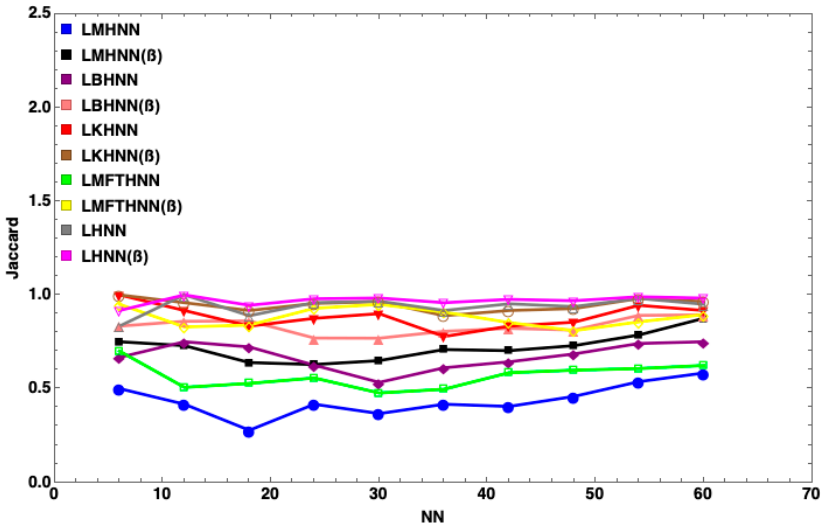

Figure 7. Jaccard Index of HNN models in non-restricted learning.

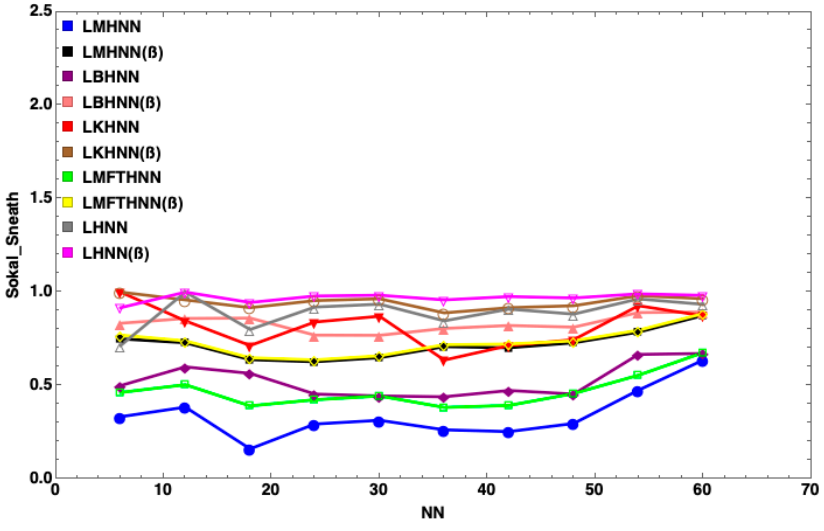

Figure 8. Sokal Sneath Index of HNN models in non-restricted learning.

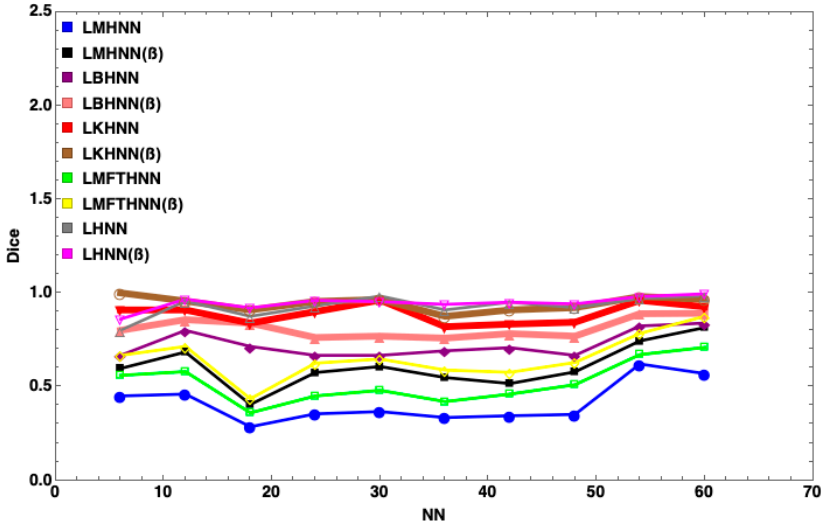

Figure 9. Dice Index of HNN models in non-restricted learning.

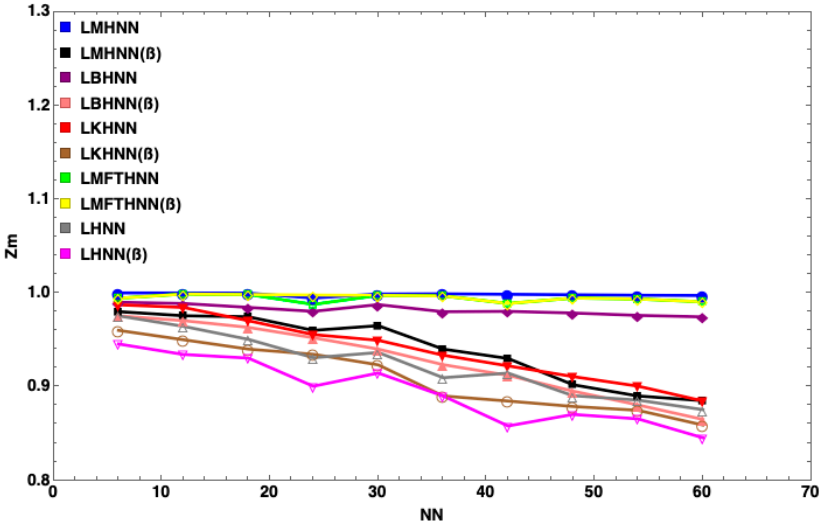

Figure 10. Global minima ratio (*Zm*) of HNN models in non-restricted learning.

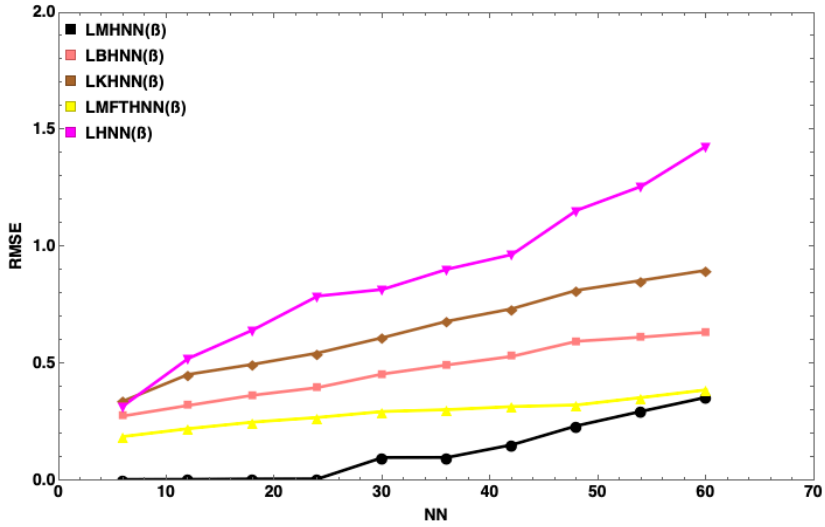

Figure 11. RMSE of HNN models (with noise) in non-restricted learning.

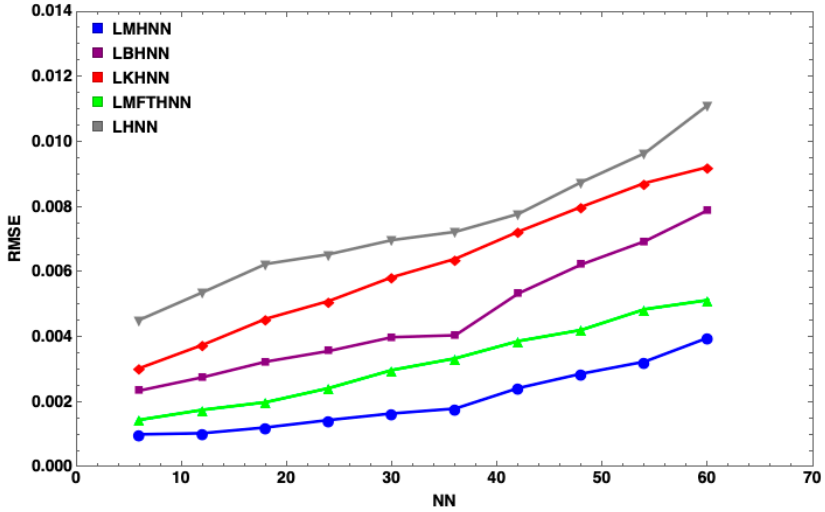

Figure 12. RMSE of HNN models (without noise) in non-restricted learning.

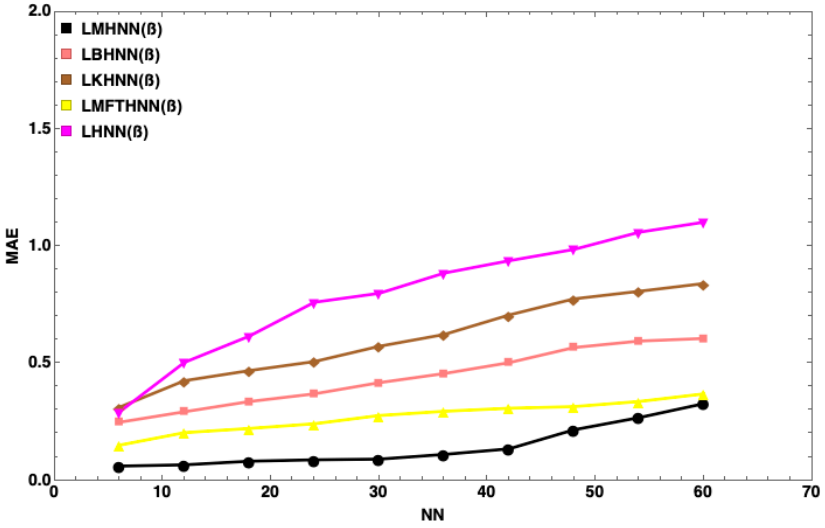

Figure 13. MAE of HNN models (with noise) in non-restricted learning.

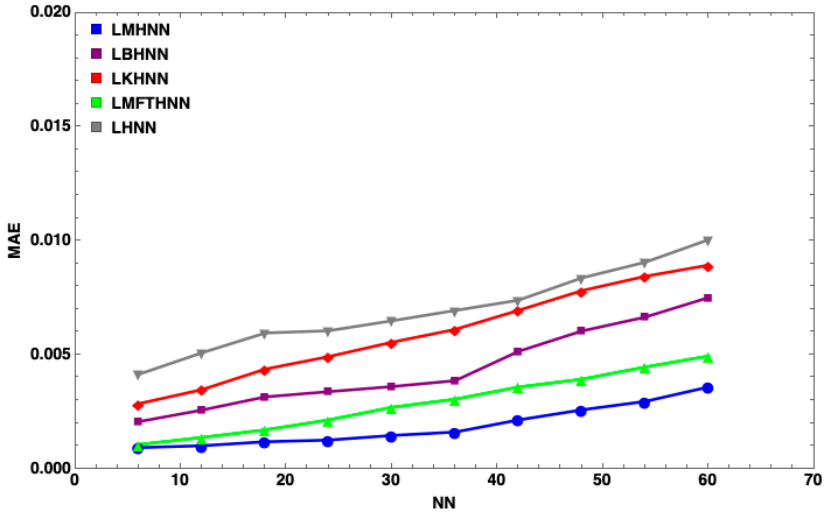

Figure 14. MAE of HNN models (without noise) in non-restricted learning.

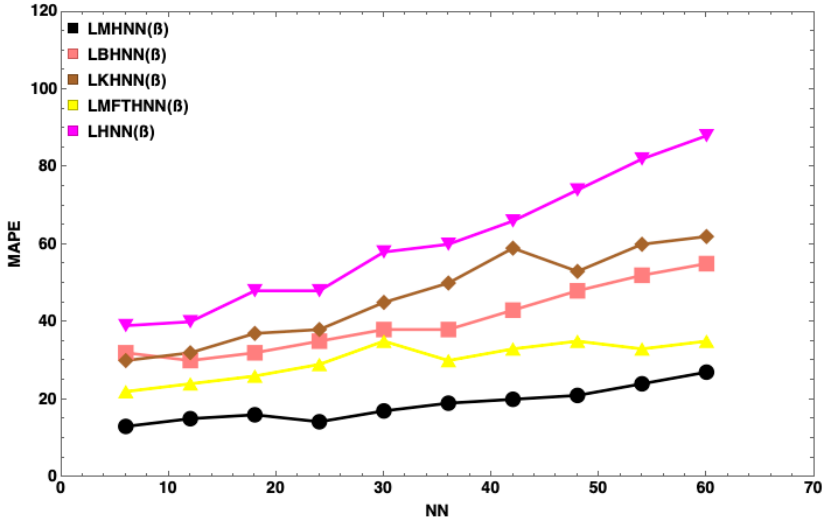

Figure 15. MAPE of HNN models (with noise) in non-restricted learning.

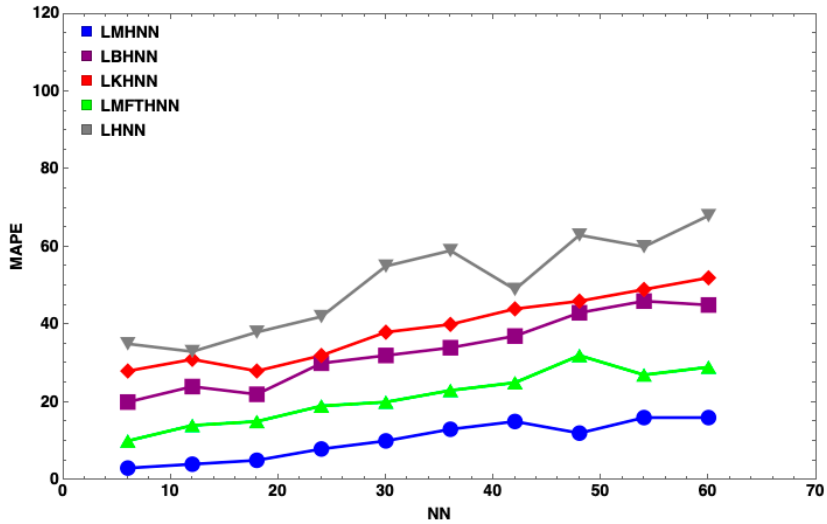

Figure 16. MAPE of HNN models (without noise) in non-restricted learning.

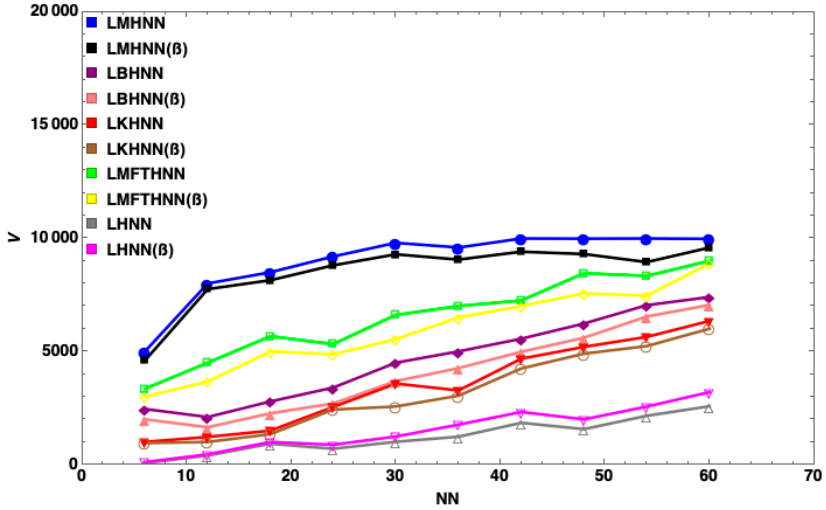

Figure 17. Variability of HNN models in non-restricted learning.

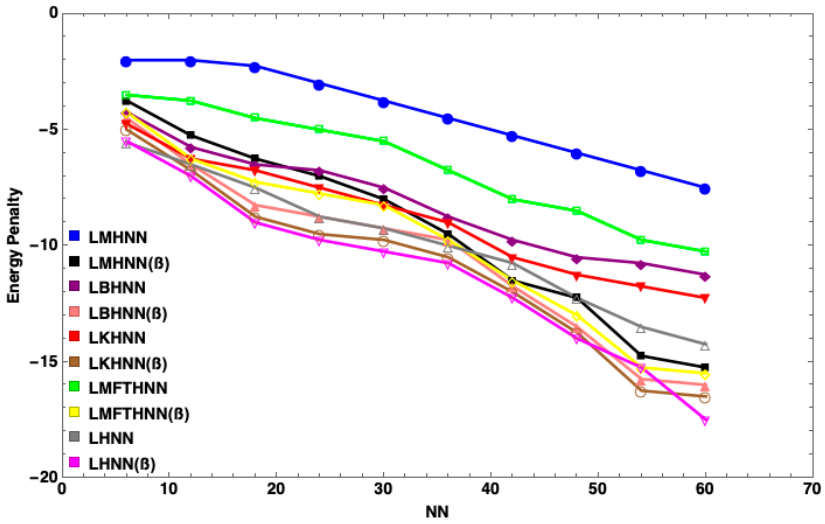

Figure 18. Energy penalty of HNN models in non-restricted learning.

The limitation of the MHNN is the computation time due to the complexity of the learning phase. Metaheuristics and accelerating algorithms, such as in [29,62], are required to reduce the learning complexity. Since all HNN models used the same learning model to find the correct interpretation, computation time is not a significant factor. In the case of $NN \geq 60$, the MHNN was trapped in a trial and error state; this phenomenon was explained in [53]. In addition, this simulation only limits the initial state to random initial points. All consistent initial states do not play a significant role in comparing the effectiveness of the proposed network. The Euclidean distance is not favored in calculating the similarity index because it is not effective in high dimensional data. For instance, both Euclidean distances for $(1, 1, 1, -1)$ and $(1, -1, 1, 1)$ with respect to $S_i^{\max} = (1, 1, -1, 1)$ are $2\sqrt{2}$. This observation lacks a consensus between HNN models with different logical rules. In addition, other established logical rules, such as MAXSAT [29], MinSAT [63] and HornSAT [64], must be investigated in depth to further verify the effectiveness of propositional logic in the MHNN.

8. Conclusions

The primary aim of an ANN is to find an optimal solution in an infinite space. Thus, we believe that the output of this study widened the ability of the conventional neural network in various mathematical perspectives, such as complex analysis [65], stability analysis [66] and forecasting analysis [67–70]. In this article, the core solution diversification principle of the EDA was found to be beneficial for ANN optimization tasks. Using this principle, we presented an efficient MHNN based on the beneficial features of the HNN and EDA. In this case, comprehensive coverage via EDA was utilized to optimize the retrieval phase of a regular HNN. The proposed MHNN was tested using both non-restricted and restricted learning models, and the comparison in terms of various performance metrics between the proposed MHNN and other established HNN models was presented. The evaluation of the results showed the superiority of the proposed MHNN model for all performance metrics. Presenting a satisfactory and efficient MHNN model was a challenging task, particularly for a large number of neurons. Therefore, the efficient learning phase of the MHNN is a subject of great importance, and future research needs to be directed to training the learning phase of the MHNN using metaheuristic algorithms, such as the genetic algorithm, artificial bee colony, artificial immune system, and ant colony optimization.

Author Contributions: Conceptualization, M.F.M.B.; methodology, software, validation, S.S.; formal analysis, writing—Original draft preparation, writing—Review and editing, M.A.M. and M.F.M.B.; project administration, funding acquisition, M.S.M.K.

Funding: This research was funded by Universiti Sains Malaysia, grant number 304/PMATHS/6315226 and the APC was funded by Universiti Sains Malaysia.

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

Appendix A

The complete implementation of HNN models is demonstrated as follows:

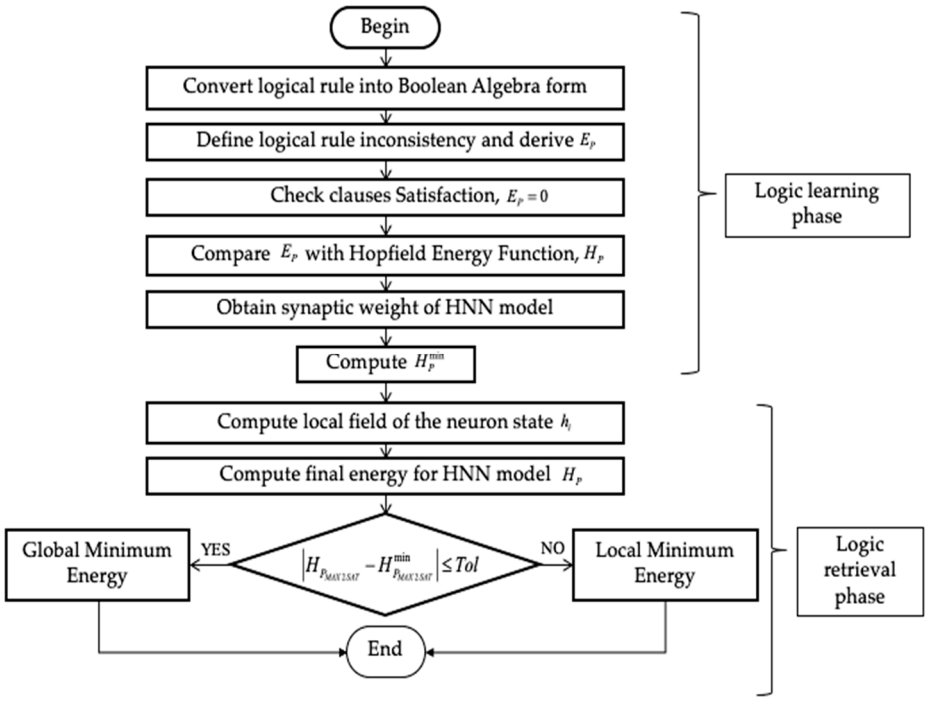

Figure A1. The flowchart of discrete Hopfield neural network.

References

1. Zamanlooy, B.; Mirhassani, M. Mixed-signal VLSI neural network based on continuous valued number system. *Neurocomputing* **2017**, *221*, 15–23. [CrossRef]
2. Fu, Y.; Aldrich, C. Flotation froth image recognition with convolutional neural networks. *Miner. Eng.* **2019**, *132*, 183–190. [CrossRef]
3. Melin, P.; Sanchez, D. Multi-objective optimization for modular granular neural networks applied to pattern recognition. *Inf. Sci.* **2018**, *460*, 594–610. [CrossRef]
4. Turabieh, H.; Mafarja, M.; Li, X. Iterated feature selection algorithms with layered recurrent neural network for software fault prediction. *Expert Syst. Appl.* **2019**, *122*, 27–42. [CrossRef]
5. Grissa, D.; Comte, B.; Petera, M.; Pujos-Guillot, E.; Napoli, A. A hybrid and exploratory approach to knowledge discovery in metabolomic data. *Discret. Appl. Math.* **2019**. [CrossRef]
6. Hopfield, J.J.; Tank, D.W. "Neural" computation of decisions in optimization problems. *Biol. Cybern.* **1985**, *52*, 141–152.
7. Silva, H.O.; Bastos-Filho, C.J. Inter-domain routing for communication networks using Hierarchical Hopfield neural network. *Eng. Appl. Artif. Intell.* **2018**, *70*, 184–198. [CrossRef]
8. Jayashree, J.; Kumar, S.A. Evolutionary Correlated Gravitational Search Algorithm (ECGS) With Genetic Optimized Hopfield Neural Network (GHNN)—A Hybrid Expert System for Diagnosis of Diabetes. *Measurement* **2019**, *145*, 551–558. [CrossRef]
9. Bafghi, M.S.; Zakeri, A.; Ghasemi, Z. Reductive dissolution of manganese in sulfuric acid in the presence of iron metal. *Hydrometallurgy* **2008**, *90*, 207–212. [CrossRef]
10. Yang, J.; Wang, L.; Wang, Y.; Gou, T. A novel memristive Hopfield neural network with application in associative memory. *Neurocomputing* **2017**, *227*, 142–148. [CrossRef]

11. Peng, M.; Gupta, N.K.; Armitage, A.F. An investigation into the improvement of local minima of the Hopfield Network. *Neural Netw.* **1996**, *90*, 207–212. [CrossRef]
12. Yang, G.; Wu, S.; Jin, Q.; Xu, J. A hybrid approach based on stochastic competitive Hopfield neural network and efficient genetic algorithm for frequency assignment problem. *Appl. Soft Comput.* **2016**, *39*, 104–116. [CrossRef]
13. Zhang, X.; Li, C.; Huang, T. Hybrid Impulsive and switching Hopfield neural networks with state-dependent impulses. *Neural Netw.* **2017**, *93*, 176–184. [CrossRef] [PubMed]
14. Kobayashi, M. Symmetric quaternionic Hopfield neural networks. *Neurocomputing* **2017**, *227*, 110–114. [CrossRef]
15. Larrañaga, P.; Karshenas, H.; Bielza, C.; Santana, R. A review on probabilistic graphical models in evolutionary computation. *J. Heuristics* **2012**, *18*, 795–819. [CrossRef]
16. Gao, S.; De Silva, C.W. Estimation distribution algorithms on constrained optimization problems. *Appl. Math. Comput.* **2018**, *339*, 323–345. [CrossRef]
17. Zhao, F.; Shao, Z.; Wang, J.; Zhang, C. A hybrid differential evolution and estimation of distributed algorithm based on neighbourhood search for job shop scheduling problem. *Int. J. Prod. Res.* **2016**, *54*, 1039–1060. [CrossRef]
18. Gu, W.; Wu, Y.; Zhang, G. A hybrid Univariate Marginal Distribution Algorithm for dynamic economic dispatch of unites considering valve-point effects and ramp rates. *Int. Trans. Electr. Energy Syst.* **2015**, *25*, 374–392. [CrossRef]
19. Fard, M.R.; Mohaymany, A.S. A copula-based estimation of distribution algorithm for calibration of microscopic traffic models. *Transp. Res. Part C* **2019**, *98*, 449–470. [CrossRef]
20. Gobeyn, S.; Mouton, A.M.; Cord, A.F.; Kaim, A.; Volk, M.; Goethals, P.L. Evolutionary algorithms for species distribution modelling: A review in the context of machine learning. *Ecol. Model.* **2019**, *392*, 179–195. [CrossRef]
21. Wang, J. Hopfield neural network based on estimation of distribution for two-page crossing number problem. *IEEE Trans. Circuits Syst. II* **2008**, *55*, 797–801. [CrossRef]
22. Hu, L.; Sun, F.; Xu, H.; Liu, H.; Zhang, X. Mutation Hopfield neural network and its applications. *Inf. Sci.* **2011**, *181*, 92–105. [CrossRef]
23. Glaßer, C.; Jonsson, P.; Martin, B. Circuit satisfiability and constraint satisfaction around Skolem Arithmetic. *Theor. Comput. Sci.* **2017**, *703*, 18–36. [CrossRef]
24. Budinich, M. The Boolean Satisfiability Problem in Clifford algebra. *Theor. Comput. Sci.* **2019**. [CrossRef]
25. Jensen, L.S.; Kaufmann, I.; Larsen, K.G.; Nielsen, S.M.; Srba, J. Model checking and synthesis for branching multi-weighted logics. *J. Log. Algebraic Methods Program.* **2019**, *105*, 28–46. [CrossRef]
26. Małysiak-Mrozek, B. Uncertainty, imprecision, and many-valued logics in protein bioinformatics. *Math. Biosci.* **2019**, *309*, 143–162. [CrossRef] [PubMed]
27. Christoff, Z.; Hansen, J.U. A logic for diffusion in social networks. *J. Appl. Log.* **2015**, *13*, 48–77. [CrossRef]
28. Xue, C.; Xiao, S.; Ouyang, C.H.; Li, C.C.; Gao, Z.H.; Shen, Z.F.; Wu, Z.S. Inverted mirror image molecular beacon-based three concatenated logic gates to detect p53 tumor suppressor gene. *Anal. Chim. Acta* **2019**, *1051*, 179–186. [CrossRef]
29. 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]
30. Tasca, L.C.; de Freitas, E.P.; Wagner, F.R. Enhanced architecture for programmable logic controllers targeting performance improvements. *Microprocess. Microsyst.* **2018**, *61*, 306–315. [CrossRef]
31. Wan Abdullah, W.A.T. Logic programming on a neural network. *Int. J. Intell. Syst.* **1992**, *7*, 513–519. [CrossRef]
32. Sathasivam, S. First Order Logic in Neuro-Symbolic Integration. *Far East J. Math. Sci.* **2012**, *61*, 213–229.
33. Mansor, M.A.; Sathasivam, S. Accelerating Activation Function for 3-Satisfiability Logic Programming. *Int. J. Intell. Syst. Appl.* **2016**, *8*, 44–50.
34. Sathasivam, S. Upgrading logic programming in Hopfield network. *Sains Malays.* **2010**, *39*, 115–118.
35. Sathasivam, S. Learning Rules Comparison in Neuro-Symbolic Integration. *Int. J. Appl. Phys. Math.* **2011**, *1*, 129–132. [CrossRef]
36. Mansor, M.A.; Sathasivam, S. Performance analysis of activation function in higher order logic programming. *AIP Conf. Proc.* **2016**, 1750.

37. Kasihmuddin, M.S.B.M.; Sathasivam, S. Accelerating activation function in higher order logic programming. *AIP Conf. Proc.* **2016**, 1750.
38. Yoon, H.U.; Lee, D.W. Subplanner Algorithm to Escape from Local Minima for Artificial Potential Function Based Robotic Path Planning. *Int. J. Fuzzy Log. Intell. Syst.* **2018**, *18*, 263–275. [CrossRef]
39. Velavan, M.; Yahya, Z.R.; Abdul Halif, M.N.; Sathasivam, S. Mean field theory in doing logic programming using hopfield network. *Mod. Appl. Sci.* **2016**, *10*, 154. [CrossRef]
40. Alzaeemi, S.A.; Sathasivam, S. Linear kernel Hopfield neural network approach in horn clause programming. *AIP Conf. Proc.* **2018**, *1974*, 020107.
41. Paul, A.; Poloczek, M.; Williamson, D.P. Simple Approximation Algorithms for Balanced MAX 2SAT. *Algorithmica* **2018**, *80*, 995–1012. [CrossRef]
42. Morais, C.V.; Zimmer, F.M.; Magalhaes, S.G. Inverse freezing in the Hopfield fermionic Ising spin glass with a transverse magnetic field. *Phys. Lett. A* **2011**, *375*, 689–697. [CrossRef]
43. Barra, A.; Beccaria, M.; Fachechi, A. A new mechanical approach to handle generalized Hopfield neural networks. *Neural Netw.* **2018**, *106*, 205–222. [CrossRef] [PubMed]
44. Zarco, M.; Froese, T. Self-modeling in Hopfield neural networks with continuous activation function. *Procedia Comput. Sci.* **2018**, *123*, 573–578. [CrossRef]
45. Abdullah, W.A.T.W. The logic of neural networks. *Phys. Lett. A* **1993**, *176*, 202–206. [CrossRef]
46. Kumar, S.; Singh, M.P. Pattern recall analysis of the Hopfield neural network with a genetic algorithm. *Comput. Math. Appl.* **2010**, *60*, 1049–1057. [CrossRef]
47. Salcedo-Sanz, S.; Ortiz-García, E.G.; Pérez-Bellido, Á.M.; Portilla-Figueras, A.; López-Ferreras, F. On the performance of the LP-guided Hopfield network-genetic algorithm. *Comput. Oper. Res.* **2009**, *36*, 2210–2216. [CrossRef]
48. Wu, J.; Long, J.; Liu, M. Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm. *Neurocomputing* **2015**, *148*, 136–142. [CrossRef]
49. Chen, D.; Chen, Q.; Leon, A.S.; Li, R. A genetic algorithm parallel strategy for optimizing the operation of reservoir with multiple eco-environmental objectives. *Water Resour. Manag.* **2016**, *30*, 2127–2142. [CrossRef]
50. García-Martínez, C.; Rodriguez, F.J.; Lozano, M. Genetic Algorithms. *Handb. Heuristics* **2018**, 431–464. [CrossRef]
51. Tian, J.; Hao, X.; Gen, M. A hybrid multi-objective EDA for robust resource constraint project scheduling with uncertainty. *Comput. Ind. Eng.* **2019**, *130*, 317–326. [CrossRef]
52. Fang, H.; Zhou, A.; Zhang, H. Information fusion in offspring generation: A case study in DE and EDA. *Swarm Evol. Comput.* **2018**, *42*, 99–108. [CrossRef]
53. Kasihmuddin, M.S.M.; Mansor, M.A.; Sathasivam, S. Hybrid Genetic Algorithm in the Hopfield Network for Logic Satisfiability Problem. *Pertanika J. Sci. Technol.* **2017**, *1870*, 050001.
54. Bag, S.; Kumar, S.K.; Tiwari, M.K. An efficient recommendation generation using relevant Jaccard similarity. *Inf. Sci.* **2019**, *483*, 53–64. [CrossRef]
55. Pachayappan, M.; Panneerselvam, R. A Comparative Investigation of Similarity Coefficients Applied to the Cell Formation Problem using Hybrid Clustering Algorithms. *Mater. Today: Proc.* **2018**, *5*, 12285–12302. [CrossRef]
56. Cardenas, C.E.; McCarroll, R.E.; Court, L.E.; Elgohari, B.A.; Elhalawani, H.; Fuller, C.D.; Kamal, M.J.; Meheissen, M.A.; Mohamed, A.S.; Rao, A.; et al. Deep learning algorithm for auto-delineation of high-risk oropharyngeal clinical target volumes with built-in dice similarity coefficient parameter optimization function. *Int. J. Radiat. Oncol. Biol. Phys.* **2018**, *101*, 468–478. [CrossRef]
57. Ikemoto, S.; DallaLibera, F.; Hosoda, K. Noise-modulated neural networks as an application of stochastic resonance. *Neurocomputing* **2018**, *277*, 29–37. [CrossRef]
58. Ong, P.; Zainuddin, Z. Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction. *Appl. Soft Comput.* **2019**, *80*, 374–386. [CrossRef]
59. Kasihmuddin, M.S.M.; Mansor, M.A.; Sathasivam, S. Maximum 2 satisfiability logical rule in restrictive learning environment. *AIP Publ.* **2018**, *1974*, 020021.
60. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
61. Cheng, S.; Chen, J.; Wang, L. Information perspective to probabilistic modeling: Boltzmann machines versus born machines. *Entropy* **2018**, *20*, 583. [CrossRef]

62. Mansor, M.A.; Kasihmuddin, M.S.M.; Sathasivam, S. Modified Artificial Immune System Algorithm with Elliot Hopfield Neural Network for 3-Satisfiability Programming. *J. Inform. Math. Sci.* **2019**, *11*, 81–98.
63. Li, K.; Lu, W.; Liang, C.; Wang, B. Intelligence in Tourism Management: A Hybrid FOA-BP Method on Daily Tourism Demand Forecasting with Web Search Data. *Mathematics* **2019**, *7*, 531. [CrossRef]
64. Frosini, A.; Vuillon, L. Tomographic reconstruction of 2-convex polyominoes using dual Horn clauses. *Theor. Comput. Sci.* **2019**, *777*, 329–337. [CrossRef]
65. Shu, J.; Xiong, L.; Wu, T.; Liu, Z. Stability Analysis of Quaternion-Valued Neutral-Type Neural Networks with Time-Varying Delay. *Mathematics* **2019**, *7*, 101. [CrossRef]
66. Yun, B.I. A Neural Network Approximation Based on a Parametric Sigmoidal Function. *Mathematics* **2019**, *7*, 262. [CrossRef]
67. Wu, Z.; Christofides, P.D. Economic Machine-Learning-Based Predictive Control of Nonlinear Systems. *Mathematics* **2019**, *7*, 494. [CrossRef]
68. Kanokoda, T.; Kushitani, Y.; Shimada, M.; Shirakashi, J.I. Gesture Prediction using Wearable Sensing Systems with Neural Networks for Temporal Data Analysis. *Sensors* **2019**, *19*, 710. [CrossRef]
69. Wong, W.; Chee, E.; Li, J.; Wang, X. Recurrent Neural Network-Based Model Predictive Control for Continuous Pharmaceutical Manufacturing. *Mathematics* **2018**, *6*, 242. [CrossRef]
70. Shah, F.; Debnath, L. Wavelet Neural Network Model for Yield Spread Forecasting. *Mathematics* **2017**, *5*, 72. [CrossRef]