



Research article

Major 3 Satisfiability logic in Discrete Hopfield Neural Network integrated with multi-objective Election Algorithm

Muhammad Aqmar Fiqhi Roslan¹, Nur Ezlin Zamri², Mohd. Asyraf Mansor^{2,*} and Mohd Shareduwan Mohd Kasihmuddin¹

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

² School of Distance Education, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

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

Abstract: Discrete Hopfield Neural Network is widely used in solving various optimization problems and logic mining. Boolean algebras are used to govern the Discrete Hopfield Neural Network to produce final neuron states that possess a global minimum energy solution. Non-systematic satisfiability logic is popular due to the flexibility that it provides to the logical structure compared to systematic satisfiability. Hence, this study proposed a non-systematic majority logic named Major 3 Satisfiability logic that will be embedded in the Discrete Hopfield Neural Network. The model will be integrated with an evolutionary algorithm which is the multi-objective Election Algorithm in the training phase to increase the optimality of the learning process of the model. Higher content addressable memory is proposed rather than one to extend the measure of this work capability. The model will be compared with different order logical combinations $k = 3,2$, $k = 3,2,1$ and $k = 3,1$. The performance of those logical combinations will be measured by Mean Absolute Error, Global Minimum Energy, Total Neuron Variation, Jaccard Similarity Index and Gower and Legendre Similarity Index. The results show that $k = 3,2$ has the best overall performance due to its advantage of having the highest chances for the clauses to be satisfied and the absence of the first-order logic. Since it is also a non-systematic logical structure, it gains the highest diversity value during the learning phase.

Keywords: Discrete Hopfield Neural Network; non-systematic logic; flexible logical rule; Major 3 Satisfiability logic; Election Algorithm

1. Introduction

The rapid development of data processing has increased the development of Artificial Intelligence (AI). AI systems have been rapidly created and deployed in practically all types of sectors over the last several years by emulating human features like problem-solving, learning, perception, comprehension, reasoning and awareness of surroundings. Artificial neural networks (ANNs) have received the most attention among the common and significant AI techniques because of their capacity to handle large amounts of data and anticipate outcomes [1]. ANNs are a network of neuron-like units that is optimized by a series of training. ANNs models are best used for logic mining and prediction for big data problems and ANNs are a form of AI. ANNs are mathematical models inspired by the biological processes of the human brain [2]. If the brain is the operating system for humans, then ANNs are the operating system for machines. NNs are a decision-making system that goes through a training process based on a set of rules and logic. Like a brain, ANNs consist of neurons that are connected by a synaptic weight. This synaptic weight carries crucial information for ANNs to perform. [3] presented the Hopfield Neural Network (HNN) a form of ANNs as a way to address a combinatorial problem. In the work, the computing capacity and speed of collective analog networks of neurons in solving optimization problems have been quantitatively demonstrated. In searching for the best solution, a very good answer computed in a quick enough period to be employed in the choice of suitable action is more important than a nominally improved “best” solution. This is especially true in perception and pattern recognition tasks in biology and robotics [3]. Wan Abdullah method [4] is used in this study since the study capitalizes on the energy minimization of the Lyapunov energy function and the cost function. To accelerate the convergence property of the logic programming of the Hopfield Neural Network, [5] propose a relaxation method. This relaxation method can stabilize the high neuron oscillations. In an exploration of searching for a very good answer computed in a quick enough period, HNN has evolved throughout time due to countless improvization that has been made by AI researchers. Scholars have been applying logical structure in HNN as a tool to navigate HNN for obtaining a better final solution. Satisfiability logic (SAT) is a popular logical structure that can be applied in HNN to counter this problem [6]. SAT also complies with the Wan Abdullah method, which caters to bipolar representation [4]. This is important, as it will convey the data that is provided into mathematical information [7]. [8] has proposed a study that utilizes k SAT in Discrete Hopfield Neural Network (DHNN). The logical structures used are Horn-Satisfiability (HORN-SAT), 2 Satisfiability (2SAT) and 3 Satisfiability (3SAT) logic. Moreover, [9] are proposing 3SAT logic that will be aided by an Artificial Immune System (AIS) in solving the 3SAT problem. [10] also use the 3SAT logical structure to represent the entries of Amazon Employees Resources Access (AERA). There are also a few works that highlight the 3SAT clauses in a non-systematic environment. [11] propose higher-order Random 3 Satisfiability (RAN3SAT), where the 3SAT clauses are generated randomly, including other types of clauses. This logical structure is able to increase the flexibility of the logical structure. Next, inspired by the work of [12] that captures the majority concept in terms of literal, [5] proposes a new novel non-systematic logic named Major 2 Satisfiability (MAJ2SAT) logic. The logical structure features the 2SAT clauses as the majority clauses in a logical string. This logical structure emphasizes the majority ratio of 2SAT clauses with other types of clauses. [9] utilize 3SAT clauses in a systematic environment, while [11] and [5] utilize 3SAT clauses in a non-systematic environment. While MAJ2SAT and RAN3SAT have high flexibility, systematic 3SAT still has the highest chances for clauses to be satisfied. Therefore, inspired by [9], [11] and [5] a new variant of non-systematic logic named Major

3 Satisfiability (MAJ3SAT) logic is proposed in this study majoring the 3SAT clauses.

In improving the learning process, metaheuristics have been applied in this study to compensate for the high computational load that will occur in the learning phase when higher neuron numbers (N) are applied [13]. It is said that Exhaustive Search (ES) is not optimal for delivering the best performance since the computational cost is increasing [8]. Hence, due to the likelihood of discovering the HNN's ideal cost function value reducing to zero, learning algorithms are needed to aid the model produces final neuron states with an optimal state. The metaheuristic is used to maximize the logical rule's fitness so that the cost function can be minimized effectively. Election Algorithm (EA) is a known metaheuristic that is proposed by [14]. It is inspired by the social-politic phenomenon of the presidential election. EA is an iterative population algorithm that works with a solution of people, each of whom is either a candidate or voter and EA is an evolutionary and swarm-based algorithm [7]. [15] then modified the benchmark EA. The work modified the party formation stage, added chaotic positive advertisement, and included a migration operator to create a novel Chaotic Election Algorithm (CEA). It is said that CEA is able to increase the population's diversity and prevents early convergences. Next, [16] becomes a pioneering work for applying EA in the non-systematic logic-Discrete Hopfield Neural Network model. EA is able to improve the learning capability of the model. [7] then use the same method to deal with non-systematic higher-order Random k Satisfiability logic. EA is used to compensate for the high computational cost during the learning phase. [6] then proposed a multi-objective Hybrid Election Algorithm (HEA) in a non-systematic logic DHNN model. The proposed HEA will broaden the logical rule while maximizing the fitness of a logical string to boost the DHNN's storage capacity. [6] and [7] then inspired this work to propose a multi-objective EA as the learning algorithm that will be integrated into the learning phase of DHNN. It is the same EA as [7] but with multi-objective functions. As the number of neurons that hold pattern rises, the capacity problems in the discrete neural network get worse [17]. While attempting to increase the storage capacity of DHNN, accuracy and diversity are the aims that need to be noted. Therefore, the diversity phase will be considered after the learning process to ensure a better solutions profile since this model is considering higher Content Addressable Memory (CAM). This attempt is inspired by [6] in expanding the model's storage capacity while avoiding over-fitting solutions. The new contributions of our work are as follows:

- 1) To formulate a new non-systematic logical structure named Major 3 Satisfiability logic where 3SAT is the majority clauses in the logical string with different logical combinations, k and embedded it in Discrete Hopfield Neural Network to increase the chances of obtaining more correct synaptic weight.
- 2) To imply a multi-objective Election Algorithm in the learning phase of Major 3 Satisfiability logic to enhance the learning process by capitalizing the explore and exploit mechanism and replacing the Exhaustive Search.
- 3) To formulate multi-objective functions that can optimize the solutions' fitness and diversity while expanding the storage capacity.
- 4) To evaluate the compatibility and the behavior of the proposed model with different logical combinations in terms of learning error, diversity error, testing error, energy management, global minimum energy and neuron variations.

This model will be generated randomly by computer and utilize simulated data. The same α as proposed by [5] is used to control the ratio of the 3SAT clauses and the other clauses. This study consists of the introduction of the work in Section 1 and the motivation of the work in Section 2. The

detailed explanation of MAJ3SAT and DHNN will be covered in Sections 3 and 4, respectively. The algorithm of the multi-objective Election Algorithm is explained in Section 5 and the experimental setup and parameter control can be seen in Section 6. Section 7 will cover the discussions of the model's performance and Section 8 will conclude the work. This model is suitable for classification and forecasting which can be extended in the logic mining model [18]. The likelihood that the best-induced logic will be found by logic mining to represent the datasets is increased by the best logical rule [19].

2. Motivation

2.1. Boolean logical structure

The boolean logical structure is needed to represent information in a logical state. SAT is a logical structure that is formed in the Conjunctive Normal Form (CNF). The Boolean logical structure consists of a set of literals that are connected by the OR operator (\cup) and a set of clauses that are connected by the AND operator (\cap) [5]. The logical structure provides a pattern behavior that maps to DHNN to generate initial and final neuron states. It is hard for DHNN to achieve global minimum energy as no guide can drive the neuron states to global minimum energy [20]. [8] and [9] proposed systematic satisfiability logic, which is effective in producing global minimum energy. However, this kind of logic lacks diversity and flexibility in terms of logical interpretations. Therefore, [11] proposed a study that utilized higher-order non-systematic logic that can tend to this problem. The proposed Random k -Satisfiability (RANkSAT) logic is able to achieve minimum cost function while having variation of final neuron states. Another variant of non-systematic logic is the logical structure that is proposed by [5]. This study featured the majority of the 2SAT clauses to provide higher chances of obtaining satisfying interpretations. Although without using any metaheuristics, this logical structure manages to obtain a higher number of final neuron states that can escape sub-optimal conditions. Inspired by this work and the work of [9] that features the 3SAT clauses, MAJ3SAT logic is proposed. The major characteristic that 3SAT provides will increase the chances of obtaining correct interpretations clauses and higher correct synaptic weights can be generated.

2.2. Metaheuristic

The metaheuristic is a learning algorithm that is developed in order to minimize the cost function. The aim of applying metaheuristics in the learning phase is to minimize the cost function as minimum as possible to obtain a more correct synaptic weight. Metaheuristics can also guide the network to produce better outcomes [21]. [8] has used the Genetic Algorithm (GA) as the learning algorithm in the paper. GA is able to minimize the cost function of a systematic logical structure. In the same year also, the author proposed Artificial Bee Colony (ABC) a learning algorithm that take inspired by the colony of bees. Then, in working with metaheuristic algorithms, [7] utilized EA in the learning phase of DHNN embed with RANkSAT logic. It is proven that EA works well with non-systematic logic to minimize the cost function. [6] proposed a hybrid EA with an extra objective function in the paper. Inspired by [7] and [6], this study then proposed multi-objective EA, utilizing the same EA as [7] and integrating an additional objective function, which is the diversity function. The diversity function is proposed to obtain diverse neurons during the learning phase since this study will consider 5 CAM to

increase the storage capacity. It is also to avoid an over-fitting solution that may be generated.

3. Major 3 Satisfiability logic (MAJ3SAT)

Major 3 Satisfiability logic is a novel variant of a non-systematic logic represented in CNF. The logical string consists of clauses with different orders of clauses (3SAT, 2SAT and first-order logic). This type of satisfiability logic provides flexibility for the user interface. MAJ3SAT features the 3SAT clauses as the majority clauses in the logical string. The number of the 3SAT clauses per string must be more than the other SAT. This logical structure will lose its majority feature if the number of 3SAT clauses is the same or lesser than the other SAT. The majority of terms in the logical structure can increase the variation and the solution generated [5]. The logical string consists of non-redundant literal to obtain the correct synaptic weight. The general formula of MAJ3SAT is presented in Eqs (1–3) for each possible logical combination. k represent the logical combination of the order of the clauses and n , m and r are the total number of 3SAT, 2SAT and first-order logic clauses respectively. The logical structure of MAJ3SAT is presented in a few components which are:

- a) A set of variables $c_1^*, c_2^*, c_3^*, \dots, c_y^*$.
- b) A set of literals c_y^* or the negation of it, $\neg c_y^*$.
- c) A set of 3SAT clauses $C_1^{(3)}, C_2^{(3)}, C_3^{(3)}, \dots, C_n^{(3)}$ where $C_i^{(3)} = (a_i \vee b_i \vee c_i), i \in \mathbb{N}$.
- d) A set of 2SAT clauses $C_1^{(2)}, C_2^{(2)}, C_3^{(2)}, \dots, C_m^{(2)}$ where $C_i^{(2)} = (d_i \vee e_i), i \in \mathbb{N}$.
- e) A set of first-order logic clauses $C_1^{(1)}, C_2^{(1)}, C_3^{(1)}, \dots, C_r^{(1)}$ where $C_i^{(1)} = (f_i), i \in \mathbb{N}$.

where \mathbb{N} is natural numbers, i is the index number and the bracket number (1), (2), (3) is the order of clauses.

$$(k = 3,2,1): \quad \varphi_{MAJ3SAT} = \bigwedge_{i=1}^n C_i^{(3)} \bigwedge_{i=1}^m C_i^{(2)} \bigwedge_{i=1}^r C_i^{(1)}. \quad (1)$$

$$(k = 3,2): \quad \varphi_{MAJ3SAT} = \bigwedge_{i=1}^n C_i^{(3)} \bigwedge_{i=1}^m C_i^{(2)}. \quad (2)$$

$$(k = 3,1): \quad \varphi_{MAJ3SAT} = \bigwedge_{i=1}^n C_i^{(3)} \bigwedge_{i=1}^r C_i^{(1)}. \quad (3)$$

In capitalizing majority terms, the ratio of the clauses is one of the aspects that we need to focus on. Hence, α is proposed to represent the ratio of 3SAT clauses to the total number of clauses. α needs to be between 0.5 and 1 (exclusive) and $n > m$ and r . α cannot include 0.5, as it will destroy the majority terms. α also cannot include 1 as the logical structure will overlap with systematic satisfiability (k SAT) that is proposed by [11]. Even if the logical structure will be generated randomly, the number of clauses is pre-determined. The total number of 2SAT clauses and first-order logic cannot exceed the 3SAT clauses. Furthermore, the reason α is proposed is to avoid bias, as the different logical combinations will give different neuron numbers. In some cases, the same α can have a different number of clauses. Therefore, the total number of clauses (TC) is proposed. TC will fix the number of clauses that can exist in a particular logical string. Equation (4) presented the equation for α .

$$\alpha = \frac{n}{n+m+r} \quad \text{where } \alpha \in (0.5,1). \quad (4)$$

The numerator is the total number of 3SAT clauses whereas the denominator is TC . TC is used to

provide a domain for α . TC can be presented in an equation where $TC = n + m + r$. Each TC will possess all possible α in the logical string. Each literal in the clauses is represented in bipolar representation which indicates that $c_i^* = \{1, -1\}$ [5]. The advanced feature that $\varphi_{MAJ3SAT}$ possessed can be utilized in the logic mining field. Equations (5)–(7) show the example of $\varphi_{MAJ3SAT}$ logical structure for $\alpha = 0.57$, $\alpha = 0.60$ and $\alpha = 0.75$ respectively. Equation 5 utilize $n = 4, m = 2$, and $r = 1$. While Eq 6 utilize $n = 3, m = 2$, and $r = 0$ and Eq 7 utilize $n = 3, m = 0$, and $r = 1$.

$$\begin{aligned} \varphi_{MAJ3SAT} = & (\neg a_1 \vee b_1 \vee c_1) \wedge (\neg a_2 \vee \neg b_2 \vee c_2) \wedge (a_3 \vee b_3 \vee \neg c_3) \\ & \wedge (a_4 \vee b_4 \vee \neg c_4) \wedge (\neg d_1 \vee \neg e_1) \wedge (d_2 \vee e_2) \wedge (f_1). \end{aligned} \quad (5)$$

$$\begin{aligned} \varphi_{MAJ3SAT} = & (\neg a_1 \vee b_1 \vee c_1) \wedge (\neg a_2 \vee \neg b_2 \vee c_2) \wedge (a_3 \vee b_3 \vee \neg c_3) \\ & \wedge (\neg d_1 \vee \neg e_1) \wedge (d_2 \vee e_2). \end{aligned} \quad (6)$$

$$\begin{aligned} \varphi_{MAJ3SAT} = & (a_1 \vee \neg b_1 \vee c_1) \wedge (\neg a_2 \vee \neg b_2 \vee \neg c_2) \wedge (a_3 \vee b_3 \vee c_3) \\ & \wedge (f_1). \end{aligned} \quad (7)$$

As we know, the proposed $\varphi_{MAJ3SAT}$ is a non-systematic logic. This logic is proposed based on MAJ2SAT that is proposed by [5]. MAJ2SAT features the majority of the 2SAT clauses while MAJ3SAT will feature the majority of the 3SAT clauses. This crucial concept also differentiates MAJ3SAT from RAN3SAT which is proposed by [7] where the number of clauses per string is randomized but needs to have at least one 3SAT clause. To be added, this logic is predicted to be able to provide higher satisfied interpretations.

4. $\varphi_{MAJ3SAT}$ in Discrete Hopfield Neural Network

A Discrete Hopfield Neural Network is a feedback network made up of interconnected neurons that are modeled after the human brain system. DHNN acts as a decision-making system for a computer. This neural network has no hidden layers [22]. As a result, the input will go directly to the output. Second, during the testing phase, the synaptic weight is stored in DHNN's efficient content addressable memory [5]. Finally, DHNN updates the neurons asynchronously [16], which means they are not updated at the same time each cycle. Fourth, DHNN's neural network caters to bipolar representation which is a logical translation of true and false [11]. 1 represents the true value and -1 represents the false value connected by the logic gate. $\varphi_{MAJ3SAT} = (1 \vee -1 \vee 1) \wedge (-1 \vee -1 \vee -1) \wedge (1 \vee 1 \vee 1) \wedge (1)$ is an example of the truth value of the logical string. DHNN utilizes synaptic weight to carry the information from the learning phase to the testing phase. The synaptic weight W is always symmetric, $W_{ij}^{(2)} = W_{ji}^{(2)}$ and the synaptic weight has no self-looping, $W_{ii}^{(2)} = W_{jj}^{(2)} = W_{kk}^{(2)} = W_{iii}^{(3)} = W_{jjj}^{(3)} = W_{kkk}^{(3)} = 0$. For your information, HNN suffers from the lack of symbolic rule that governs the neural network [7]. The global minimum energy for HNN is numerically challenging to acquire due to the lack of an effective symbolic rule [6]. Hence, that is why $\varphi_{MAJ3SAT}$ is proposed in DHNN to fuse both domains to form DHNN-MAJ3SAT.

4.1. Learning phase

The aim of the learning phase is to obtain the correct synaptic weight by minimizing the cost function which is why we apply $\varphi_{MAJ3SAT}$ in DHNN. The fully satisfied interpretation of the logical structure will lead to the cost function, $E_{\varphi_{MAJ3SAT}} = 0$. Which then will lead to correct solution generation during the testing phase. The synaptic weight will be obtained by comparing the cost function, Eq (8) and the Lyapunov energy function, Eq (10) where q_i is the literal state.

$$E_{\varphi_{MAJ3SAT}} = \frac{1}{8} \sum_{i=1}^n \left(\prod_{j=1}^3 q_i^{(3)} \right) + \frac{1}{4} \sum_{i=1}^m \left(\prod_{j=1}^2 q_i^{(2)} \right) + \frac{1}{2} \sum_{i=1}^r \left(\prod_{j=1}^1 q_i^{(1)} \right). \quad (8)$$

$$\{q_i^{(3)}, q_i^{(2)}, q_i^{(1)}\} = \begin{cases} \frac{1}{2}(1 - S_n)_{q_i}, & \text{if } q_i, \\ \frac{1}{2}(1 + S_n)_{q_i}, & \text{otherwise.} \end{cases} \quad (9)$$

$$H_{\varphi_{MAJ3SAT}} = -\frac{1}{3} \sum_{p=1, p \neq q \neq s}^N \sum_{q=1, q \neq p \neq s}^N \sum_{s=1, s \neq p \neq q}^N W_{pqs}^{(3)} S_p S_q S_s - \frac{1}{2} \sum_{p=1, p \neq q}^N \sum_{q=1, p \neq q}^N W_{pq}^{(2)} S_p S_q - \sum_{p=1}^N W_p^{(1)} S_p. \quad (10)$$

This method is proposed by Wan Abdullah (1992). n , m and r are the total number of clauses of each clause. $W_{pqs}^{(3)}$, $W_{pq}^{(2)}$, $W_p^{(1)}$ are the synaptic weight that will be generated and S_p , S_q , S_s are the neuron states. Equations (8)–(10) will be employed with $\varphi_{MAJ3SAT}$ with the aim that the cost function can be minimized. The most desired value is the cost function equal to 0 [5]. The energy profile of this model is depending on the effectiveness of the learning phase [5]. While the energy of the neuron is computed again by Eq (10), the neurons will undergo a diversity state where the benchmark diversity of the neuron will be analyzed. This is because we aimed to analyze the diversity of the solution produced with the benchmark solution during the learning phase. These new aspects that are proposed in DHNN will let the user know how diversified the solution is produced when applying higher storage capacity.

$$d_i = \sum_{i=1}^{NN} p_i, \quad (11)$$

$$p_i = \begin{cases} 1, & S_i \neq S_i^{max}, \\ 0, & S_i = S_i^{max}. \end{cases} \quad (12)$$

$$\frac{d_i}{NN} \geq tol_d. \quad (13)$$

Equations (11)–(13) will be computed as the benchmark diversity of the solution and NN is the literals number. p_i is the scoring scheme. p_i will be 1 if the targeted variables are different from benchmark variables and 0 otherwise. d_i will store the value of the summation of p_i and will be evaluated. d_i will be divided with neuron numbers, NN to obtain the diversity ratio. The solution is considered to diversify from the benchmark state if it exceeds the diversity tolerance value, tol_d . For

example, let $\varphi_{MAJ3SAT} = (a_1 \vee \neg b_1 \vee c_1) \wedge (\neg a_2 \vee \neg b_2 \vee \neg c_2) \wedge (a_3 \vee b_3 \vee c_3) \wedge (f_1)$. Then the benchmark states should be $\varphi_{MAJ3SAT} = (1 \vee -1 \vee 1) \wedge (-1 \vee -1 \vee -1) \wedge (1 \vee 1 \vee 1) \wedge (1)$. Next, the obtained neuron is $\varphi_{MAJ3SAT} = (-1 \vee -1 \vee 1) \wedge (-1 \vee 1 \vee -1) \wedge (1 \vee 1 \vee -1) \wedge (1)$. From this p_i will score the condition either S_i equal to S_i^{max} or different. The obtained neuron has 3 different variables from the benchmark solution causing $d_i = \sum_{i=1}^{NN} p_i = 3$. The neuron will be considered diversified if it satisfies Eq (13). For example, $\frac{d_i}{NN} = \frac{3}{10} = 0.3 \geq tol_d$.

4.2. Testing phase

Most of the neurons still possess unstable energy management due to high oscillations. To stabilize this, the Sathasivam relaxation phase will be employed in this study to relax the neuron and reduce its oscillation [5]. The final neuron states S_i can be generated by the local field equation, Eq (14) and the Hyperbolic Tangent Activation Function (HTAF) will squash the neuron energy value, Eq (15).

$$h_{\varphi_{MAJ3SAT}} = \sum_{s=1, s \neq q}^N \sum_{q=1, q \neq s}^N W_{pqs}^{(3)} S_q S_s + \sum_{q=1, p \neq q}^N W_{pq}^{(2)} S_q + W_p^{(1)}, \quad (14)$$

$$\tanh(h_\varphi) = \frac{e^{h_\varphi} - e^{-h_\varphi}}{e^{h_\varphi} + e^{-h_\varphi}}, \quad (15)$$

$$S_i = \begin{cases} 1, & \tanh(h_\varphi) \geq 0, \\ -1, & \tanh(h_\varphi) < 0. \end{cases} \quad (16)$$

The model then will filter the final neurons by Eq (17) after it has achieved final energy where $H_{\varphi_{MAJ3SAT}}$ is the final energy and $H_{\varphi_{MAJ3SAT}}^{min}$ is the minimum energy. The one that is stable and able to converge its energy to achieve global minimum energy will be considered as the solution. Therefore, for an effective testing phase, an optimum learning phase needs to be achieved. That is why Election Algorithm is proposed in the learning phase of DHNN.

$$\left| H_{\varphi_{MAJ3SAT}} - H_{\varphi_{MAJ3SAT}}^{min} \right| < tol. \quad (17)$$

The study by [5], [7] and [11] has proposed non-systematic logic in DHNN. It is shown that non-systematic logic able to generate more satisfied interpretations. This study utilizes HTAF as it is able to provide more optimal solutions [23]. Also, to enhance the learning process, this study will consider Election Algorithm (EA) as a metaheuristic that will be integrated into the learning phase. EA can compensate for the high computational cost during the learning phase [7].

Figure 1 represents the topology of the network where each box represents the connection between neurons in 3SAT, 2SAT and first-order logic clauses. The red dot is the synaptic weight of the clauses. This model is starting with the logic phase to set up a few parameters. The dashed line represents $k = 3, 2, 1$, the thin line represents $k = 3, 2$ and the bold line represents $k = 3, 1$. The network model will then proceed with stages in DHNN until the energy of the neurons is filtered. The passed neurons will consider having global minimum energy and otherwise will have local minimum energy.

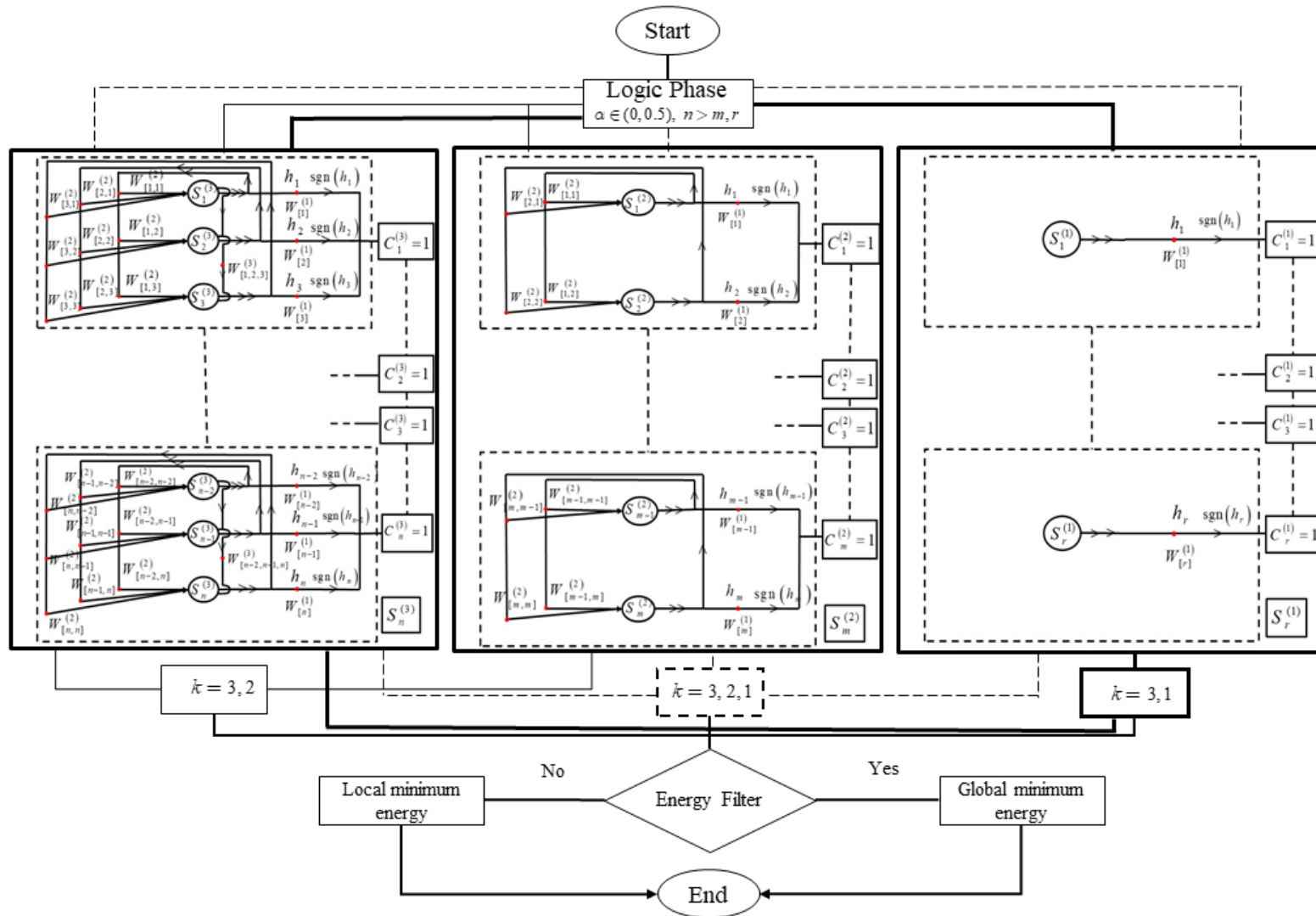


Figure 1. Topology of DHNN-MAJ3SAT.

5. Multi-objective Election Algorithm

The learning phase of this model can be improved by integrating metaheuristics. Election Algorithm will be able to maximize the fitness of each neuron which then leads to the minimization of the cost function. Election Algorithm is an evolutionary algorithm and a swarm intelligence algorithm [7]. EA replicates the behavior of candidates and voters in a presidential election process. EA divides the solution space into a few partitions and will be swarming the party based on the operators that it possessed. Each individual in EA is the possible solution that will be optimized by local and global operators. The state of the neuron also will be undergoing state flipping to create evolution in solution searching. This will increase the convergence rate of the solution search. EA is needed because the existence of 2SAT and first-order logic will lower the chances to obtain satisfied interpretations. Especially, if neuron numbers are increasing, it provides a computational burden to the model [8]. The three main operators of EA that cover local and global search operators are positive advertisement, negative advertisement and coalition. Each individual of $\varphi_{MAJ3SAT}$ is represented in a bipolar state $S_i = \{1, -1\}$ where $i = 1, 2, 3, \dots, N_{pop}$.

Stage 1: Initialization

A population of a random possible solution of $\varphi_{MAJ3SAT}$ is initialized where each individual is a voter or a candidate. The fitness of each individual will be quantified by the objective function, Eq (18) and will pick the candidates with the highest fitness for the optimization problem.

$$f_{L_i} = \sum_{i=1}^n C_i^{(3)} + \sum_{i=1}^m C_i^{(2)} + \sum_{i=1}^r C_i^{(1)}. \quad (18)$$

$$C_i^{(k)} = \begin{cases} 1, & \text{satisfied} \\ 0, & \text{otherwise} \end{cases}, k = 3, 2, 1. \quad (19)$$

Stage 2: Forming parties

N_{pop} will be divided into partitions of a party, N_{party} . The number of parties that are proposed in this study can be observed in Table 2. N_j represent the total number of individuals per party.

$$N_j = \frac{N_{pop}}{N_{party}}, \text{ where } j=1, 2, 3, 4. \quad (20)$$

After candidates are selected, EA will compute the similarity of belief between candidates L_j and voters v_j^i . The similarity of belief is presented in distance form in Eq (21).

$$\text{dist}(f_{L_j}, f_{v_j^i}) = f_{L_j} - f_{v_j^i}. \quad (21)$$

Stage 3: Positive Advertisement

The candidates that have been elected will start to advertise their campaign and will try to attract and influence the voters from their party. The candidates will try to attract as many voters as possible to have the majority voting. The number of voters that their decision-making will be influenced will be represented by Eq (22). σ^p is the positive advertisement rate and $\omega_{v_j^i}$ is the eligibility distance coefficient. Based on the effect of the candidates on the voters in Eq (23), the voters will undergo state flipping represented by Eq (24) and $S_{v_j^i}$ is the number of literals that will be flipped.

$$N_s = \sigma^p N_j, \quad \sigma^p \in [0,0.5]. \quad (22)$$

$$\omega_{v_i^j} = \frac{1}{\text{dist}(f_{L_j}, f_{v_i^j}) + 1}. \quad (23)$$

$$S_{v_i^j} = N_c \omega_{v_i^j}. \quad (24)$$

where N_c is the total number of literals per individual. Next, the fitness of each voter will be updated by Eq (18) and the candidate will be replaced if the updated voters have higher fitness than the candidate.

Stage 4: Negative advertisement

In this stage, the candidates will try to influence the decision-making of other voters from another party. This global operator will try to expand the solution space by interacting with another party with its party. The voters that are influenced by the candidate's advertisement will be represented by Eq (25).

$$N_{v_i^*} = \sigma^n (N_j - N_s), \quad \sigma^n \in [0,0.5]. \quad (25)$$

$$\text{dist}(f_{L_j}, f_{v_i^*}) = f_{L_j} - f_{v_i^*}. \quad (26)$$

$$\omega_{v_i^*} = \frac{1}{\text{dist}(f_{L_j}, f_{v_i^*}) + 1}. \quad (27)$$

$$S_{v_i^*} = N_c \omega_{v_i^*}. \quad (28)$$

σ^n is the negative advertisement rate, $\omega_{v_i^*}$ is the eligibility distance coefficient and $S_{v_i^*}$ represents the number of variables that will undergo state flipping. The voters that are influenced will undergo state flipping, Eq (28). The candidate will be replaced if the updated voters have higher fitness.

Stage 5: Coalition

The different parties will cooperate and form allies. After the two parties have united, new candidates will be randomly picked. The effect of the candidate on the party will influence all the voters. Therefore, the similarity of belief and the eligibility distance coefficient will be computed the same as in Eq (26) and Eq (27), respectively. The number of variables that will undergo state flipping will be computed by Eq (28). If voters have the highest fitness, the voters will be elected as the new candidate and will proceed with election day. If not, the old candidate will proceed to compete on election day.

Stage 6: Election day

The candidates will fight for their spots. The candidate with the highest fitness will win the election and be elected as the winner. However, rather than choosing only one, all the candidates and voters that manage to achieve the highest fitness will be considered the optimal solution for this model. If there is no individual with the highest fitness, the iteration will continue until the termination criteria are met.

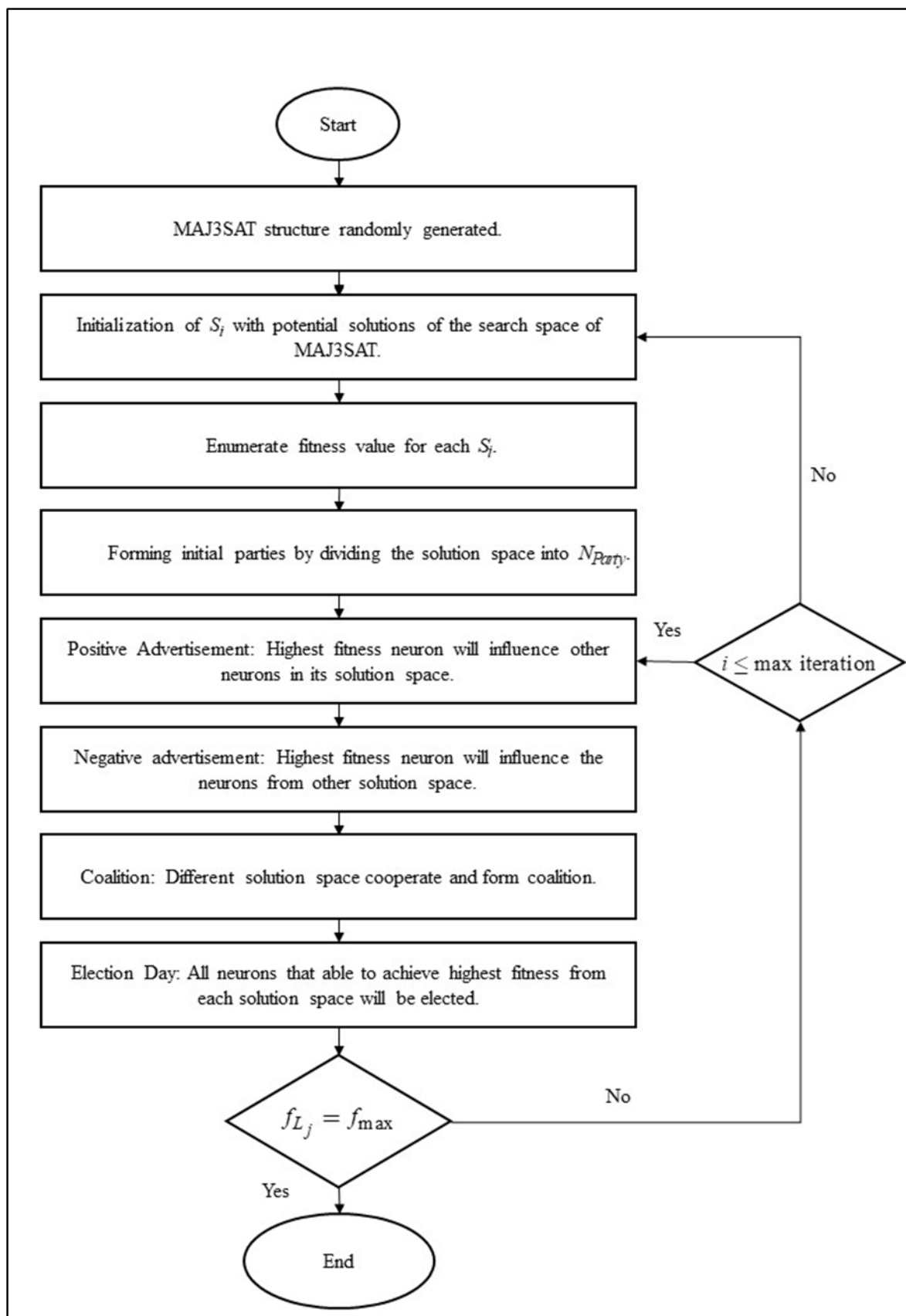


Figure 2. Flowchart of proposed EA.

Algorithm 1: Pseudo-Code of the proposed EA

```

1   Generate initial population  $N_{pop}$ 
2    $while(i < max \text{ iteration}) \text{ or } (f_{L_j} < f_{max}0)$ 
3   Forming initial parties  $N_j$  by Eq (20)
4    $for(j \leq N_{party}), do$ 
5   Evaluate the fitness of each individual by Eq (18)
6   Evaluate the similarity of belief between voters and candidates by Eq (21)
7    $end$ 
8   {Positive Advertisement}
9   Evaluate the number of influenced voters by Eq (22)
10   $for(i \leq N_s), do$ 
11  Evaluate the reasonable effect from the candidate,  $\omega_{v_j^i}$  by Eq (23)
12  Evaluate the number of state flipping by Eq (24)
13  Update the fitness of the neuron by Eq (18)
14   $if (f_{v_j^i} > f_{L_j})$ 
15  Assign  $v_j^i$  as candidate
16   $else$ 
17  Remain  $L_j$ 
18   $end$ 
19  {Negative Advertisement}
20  Evaluate the number of influenced voters by Eq (25)
21   $for(i \leq N_{v_i^*}), do$ 
22  Evaluate the reasonable effect from the candidate,  $\omega_{v_i^*}$  by Eq (27)
23  Evaluate the number of states flipping by Eq (28)
24  Update the fitness of the neuron by Eq (18)
25   $if (f_{v_i^*} > f_{L_j})$ 
26  Assign  $v_i^*$  as candidate
27   $else$ 
28  Remain  $L_j$ 
29   $end$ 
30  {Coalition}
31   $for(i \leq N_j + N_k), do$ 
32  Evaluate the reasonable effect from the candidate,  $\omega_{v_i^*}$  by Eq (27)
33  Evaluate the number of states flipping by Eq (28)
34  Update the fitness of the neuron by Eq (18)
35   $if (f_{v_i^*} > f_{L_j})$ 
36  Assign  $v_i^*$  as candidate
37   $else$ 
38  Remain  $L_j$ 
39   $end$ 
40   $end while$ 
41   $return$  output

```

[16] has proposed EA in the learning phase of DHNN to deal with the computational cost issues. The study by [7] also uses EA to tolerate the computational cost of a non-systematic logic-DHNN model. Then, the study by [6] proposed a hybrid EA that possessed few objective functions to work with in DHNN. Therefore, this study will imply the same EA structure that is proposed by [7] with an improvement that is inspired by [6]. After Election Day, the model needs to undergo a diversity phase which possessed another objective function. The multi-objective EA can ensure the optimality of the training of the neurons and the quality of the solutions produced. Algorithm 1 pictures the pseudo-code of the proposed multi-objective EA and its flow can be seen in Figure 2.

6. Experimental setup

The experimental setup and parameter control of this model's framework will be listed in this section. The performance of this model will be analyzed based on the listed parameters. This experiment will observe the synaptic weight management, minimization of the cost function, testing capability, energy profile, diversity performance and neuron variation. The framework's parameters are as follows:

6.1. Simulation datasets

This section describes the setup for simulation controls that are used in this model. The simulations were computed by Dev C++ Version 5.11 running on 12GB RAM Intel Core i5 7th Generation occupied with 64-bit Windows 10. The time threshold for the computational is 24h and the program will be terminated if it exceeds the time threshold.

The simulated data of the logical structure are generated randomly based on $\varphi_{MAJ3SAT}$ parameter settings to avoid biases. This model considers small size TC ($TC = 10$) and large size TC ($TC = 20$). Each TC will cover three types of logical combinations $k = 3,2$, $k = 3,2,1$ and $k = 3,1$. Each k will have a logical structure for $\alpha \in (0.5,1)$. For $k = 3,2,1$ the ratio of clauses between 2SAT and first-order logic are randomized also to avoid logical biases. This model works with a simulated data program to observe the performance of the model. A simulated or synthetic dataset is data that is randomly generated by a compiler in bipolar representation $\{1, -1\}$ [16]. The usefulness of the model then can be seen when integrated with real-life datasets.

6.2. Parameter control

The neuron combination used, ε is 100 and the number of learning is taken from [5], 10000. The relaxation rate for the Sathasivam relaxation phase is 3 taken from [5]. Worth to mention, that this study will utilize bipolar neuron states since bipolar neuron states help to converge to global minimum energy. It is said also that bipolar converges faster than the binary states [5]. This is due to the existence of zero states that will delete some important information like synaptic weight. This will slow down the convergence process. The operated synaptic weight will always be symmetric, $W_{ij}^{(2)} = W_{ji}^{(2)}$ and the synaptic weight has no self-looping, $W_{ii}^{(2)} = W_{jj}^{(2)} = W_{kk}^{(2)} = W_{ii}^{(3)} = W_{jj}^{(3)} = W_{kk}^{(3)} = 0$. The learning process of this model will be handled by EA [16] which will be integrated into the learning

phase. The number of strings that will be considered during Election Day is the maximum which is different from [7]. The parameter settings will be presented in Tables 1 and 2.

Table 1. Experimental settings for DHNN-MAJ3SATEA.

Parameter	Value
Tolerance value, tol	0.001 [16]
Neuron combination, ε	100
Number of learning, ω	10000 [5]
Number of trials, ν	100 [16]
Number of strings	Max
Alpha, α	(0.5,1) [5]
Order of clauses, k	$k = 3,2,1$ $k = 3,2$ $k = 3,1$
Diversity tolerance value, tol_d	0.1
Type of diversity	Benchmark
Learning iterations	100
Threshold time simulation	24 hours [24]
Threshold constraint of DHNN	0
Relaxation rate	3 [5]
Activation function	Hyperbolic Tangent Activation Function [8]
Learning method	Election Algorithm [16]
Initialization of neuron states in the learning and testing phase	Random

Table 2. Parameter settings for Election Algorithm [16]

Parameter	Value
Number of populations, N_{Pop}	120
Number of parties, N_{Party}	4
Positive advertisement rate, σ^p	0.1
Negative advertisement rate, σ^n	0.1
Candidate selection	Highest fitness (random)
Type of voter's attraction	Random
Type of state flipping	Random
Number of strings on election day	Max

The parameters remain static while doing this experiment. The same parameters are used for each α for each TC . To summarize, most of the parameters are taken from [5,6,7,11,16,24] to ensure the producibility of the model. It has also proven that the listed parameters in Tables 1 and 2 can sync with the DHNN model effectively in producing solutions.

6.3. Performance evaluation metrics

The performance of the model will be examined by six main categories: synaptic weight management, minimization of the cost function, testing capability, energy profile, diversity and neuron

variation. $MAE_{Learning}$, $MAE_{Diversity}$, $MAE_{Testing}$, MAE_{Energy} , Z_m , total neuron variation (TV), Jaccard Similarity Index (JSI) and Gower and Legendre Similarity Index (GLSI) are the eight performance metrics that will be used in estimating the quality of the model performance. $MAE_{Learning}$ will be estimating the mean absolute error (MAE) of the learning phase. Next, $MAE_{Diversity}$ will evaluate the performance of diversity in the learning phase using MAE. $MAE_{Testing}$ is an error analysis that will evaluate the performance of the testing phase. For energy analysis in the testing phase, MAE_{Energy} will evaluate the energy management's performance. Global minimum ratio (Z_m) is also evaluated in energy analysis. Meanwhile, TV , JSI and GLSI will be computed in similarity analysis, analyzing the neuron variation. Table 3 lists some of the parameters that will be used in the performance metrics formula.

Table 3. List of parameters in performance analysis.

Parameter	Value
ω	Number of learning
ε	Neuron combination
ν	Number of trials
$Z_{\varphi_{MAJ3SAT}}$	Number of global minimum solution
$L_{\varphi_{MAJ3SAT}}$	Number of local minimum solution
f_i	Current fitness of the solution
f_{max}	Maximum solution
$H_{\varphi_{MAJ3SAT}}$	Final energy
$H_{\varphi_{MAJ3SAT}}^{min}$	Minimum energy

6.3.1. Assessment for learning phase

Mean absolute error

MAE is an average model-performance metric. MAE summing the magnitude of the error without considering the square of the error will avoid escalating the error. The general equation of MAE is presented by Eq (29). A best estimate is one where these errors are at their minimum [25].

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i|. \quad (29)$$

where n is the number of errors and e_i is the error. By this formula the MAE used for this study is formulated as below:

$$MAE_{Learning} = \sum_{i=1}^{\omega} \frac{1}{\omega} |f_{max} - f_i|. \quad (30)$$

$$MAE_{Diversity} = \sum_{i=1}^{\omega} \frac{1}{\omega} |S_i^{max} - S_i|. \quad (31)$$

$$MAE_{Testing} = \sum_{i=1}^{\omega} \frac{1}{\varepsilon \nu} \left| Z_{\varphi_{MAJ3SAT}} - L_{\varphi_{MAJ3SAT}} \right|. \quad (32)$$

$$MAE_{Energy} = \sum_{i=1}^{\omega} \frac{1}{\varepsilon \nu} \left| H_{\varphi_{MAJ3SAT}} - H_{\varphi_{MAJ3SAT}}^{min} \right|. \quad (33)$$

6.3.2. Assessment for testing phase

Global minimum ratio (Z_m)

Z_m evaluate the global minimum energy solution. It evaluates the ratio of the final neuron states that attained global minimum energy and the final neuron states that attained local minimum energy. The formula Z_m is:

$$Z_m = \frac{1}{\varepsilon V} \sum_{i=1}^{\omega} Z_{\varphi_{MAJ3SAT}}. \quad (34)$$

Total neuron variation and similarity index

The retrieved final neuron states are compared with the benchmark solution and according to [5] the benchmark solution is presented below:

$$S_i^{max} = \begin{cases} 1, & \text{when } a_1, \\ -1, & \text{when } \neg a_1. \end{cases} \quad (35)$$

a_1 and $\neg a_1$ are the positive and negative states of the literals in $\varphi_{MAJ3SAT}$ clauses. The formula for TV is represented by a sum of a scoring mechanism F_i .

$$TV = \sum_{i=0}^{\beta} F_i. \quad (36)$$

$$F_i = \begin{cases} 1, & S_i \neq S_i^{max}, \\ 0, & S_i = S_i^{max}. \end{cases} \quad (37)$$

β represent the total solution produced by the model. For further exploration of analyzing the relationship between final neuron states and the benchmark states, this study will consider the similarity index as the performance metric. Table 4 presents the formula formulated for JSI and GLSI and Table 5 is the neuron states for the benchmark and target solution that will be considered in the formula.

Table 4. Similarity Index Formula.

Similarity Index	Formula
Jaccard [5]	$JSI = \frac{l}{l + m + n}$
Gower and Legendre [7]	$GLSI = \frac{lo}{l + 0.5(m + n) + o}$

Table 5. Neuron states for benchmark and target solution.

Parameter	S_i^{max}	S_i
l	1	1
m	1	-1
n	-1	1
o	-1	-1

7. Results and discussion

In this section, the performance of different structures of DHNN-MAJ3SAT will be examined and

analyzed. The simulated data sets are generated based on the parameter control. The evaluation will be divided into 2 main parts which are the learning phase and the testing phase.

7.1. Learning phase

This section will be focused on the capability of the models with different logical combinations of cost function minimization and synaptic weight management. [26] ANNs are a network of neuron-like units that is optimized by a series of training. Therefore, this section will estimate the diversity of the neuron generated based on the higher CAM used.

7.1.1. Learning error analysis

This study will use $MAE_{Learning}$ for analyzing the learning error of DHNN-MAJ3SATEA. The lower $MAE_{Learning}$ value depicts the optimality of the model [10]. The learning error shown is based on $\omega = 10000$.

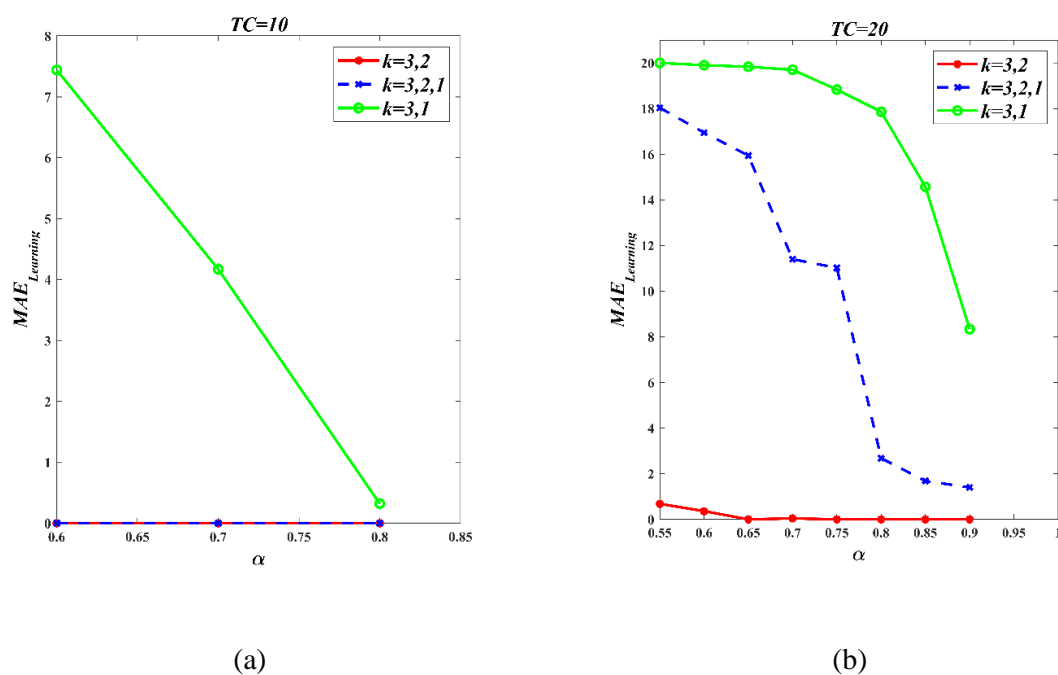


Figure 3. Learning error analysis for (a) $TC = 10$ and (b) $TC = 20$ with different k .

Note that the optimum learning phase provides a low error to the model performance due to fewer learning iterations. Figure 3 shows that $k = 3,2$ manage to obtain the best learning process. This is due to the natural characteristics of $\varphi_{MAJ3SAT}$ that have a higher probability to be satisfied with the absence of first-order logic causing more correct synaptic weight generation. The logical structure for MAJ3SAT for $k = 3,2$ does not need to obey the restriction that first-order logic will cause. Since first-order logic has the lowest way possible to be satisfied. Due to these characteristics, other k such as $k = 3,2,1$ and $k = 3,1$ have higher MAE. The logical combinations for $k = 3,1$ suffered the most

since for lower NN it was not able to achieve 0 error compare to the others because $k = 3,1$ clause has the lowest likelihood of achieving $E_{\varphi_{MAJ3SAT}} = 0$ than those in other orders of clauses due to its limitations to obtain satisfied interpretations. The $MAE_{Learning}$ for higher NN ($TC = 20$) shows an increasing trend. DHNN-MAJ3SATEA suffers due to high neuron numbers. It then provides more computational cost, resulting in a higher number of iterations of the model to obtain the correct synaptic weight. For $TC = 10$ the logical combination $k = 3,2,1$ able to have the same performance as $k = 3,2$. This is due to the 2SAT clauses in the logical string of $k = 3,2,1$. It provides a shield as it can counters back the drawback that first-order logic has provided. The Wan Abdullah method succeeds in generating more correct optimal synaptic weights. Table 6 represents the Friedman test that has been conducted for $k = 3,2$, $k = 3,2,1$ and $k = 3,1$. The Chi-Square value is $\chi^2 = 16$ and the degree of freedom is $df = 2$ for MAE learning. The null hypothesis is that there is no significant relation between different k . Considering $\alpha_0 = 0.05$ the null hypothesis is rejected stresses the importance of comparing the different logical combinations as it validates different environments for the model.

Table 6. Friedman test for learning between different k .

α	$k = 3, 2$	$k = 3, 2, 1$	$k = 3, 1$
0.55	0.68	18.0325	19.9999
0.60	0.36	16.9428	19.8998
0.65	0	15.939	19.8312
0.70	0.04	11.3987	19.6976
0.75	0	11.0198	18.8296
0.80	0	2.68	17.8595
0.85	0	1.68	14.5703
0.90	0	1.4	8.3333
Chi-Square, χ^2	16		
p – Value	0.0003355		
Accept/Reject H_0	Reject H_0		

7.1.2. Diversity error analysis

Figure 4 describes the error of diversity generated in the learning phase. The targeted neuron is considered diverse from the benchmark solution if it exceeds the diversity tolerance value, $tol_d = 0.1$. Repetitive neurons while using 5 CAM will degrade the model optimality. The aim is to gain different 5 CAM to be analyzed. This section takes place in the learning phase. The decreasing of MAE throughout α conveys that a higher number of diverse solutions are generated. The logical patterns that MAJ3SAT provided were able to increase the number of satisfied interpretations. A higher satisfied interpretation leads the model to succeed in generating more satisfied neurons with different states. For instance, for $TC = 10$ both $k = 3,2$ and $k = 3,2,1$ able to achieve 0 for all α . Both k manage to generate all solutions that own at least 10% different literals state from S_i^{max} . The disadvantage that first-order logic possesses in the logical structure can be covered by the 3SAT and 2SAT clauses causing more satisfied clauses with the help of EA. Figure 4 indicates that $k = 3,1$ have the least diversity value (highest $MAE_{Diversity}$). Thus, it has the least number of diverse neurons. It shows that

the majority of the 3SAT clauses also were not able to shield the logical string due to the existence of the first-order logic. The logical string starts to lose the majority patterns and disrupt the synaptic weight management and induce less diverse neurons. Table 7 shows the Friedman test that has been conducted for $k = 3,2$, $k = 3,2,1$ and $k = 3,1$ for MAE Diversity. The Chi-Square value is $\chi^2 = 16$ and the degree of freedom is $df = 2$. While using $\alpha_0 = 0.05$, the null hypothesis is rejected. The fact that the null hypothesis is rejected emphasizes how important it is to compare various logical arrangements. The different logical combinations are important in investigating the model's behavior in a higher storage capacity environment.

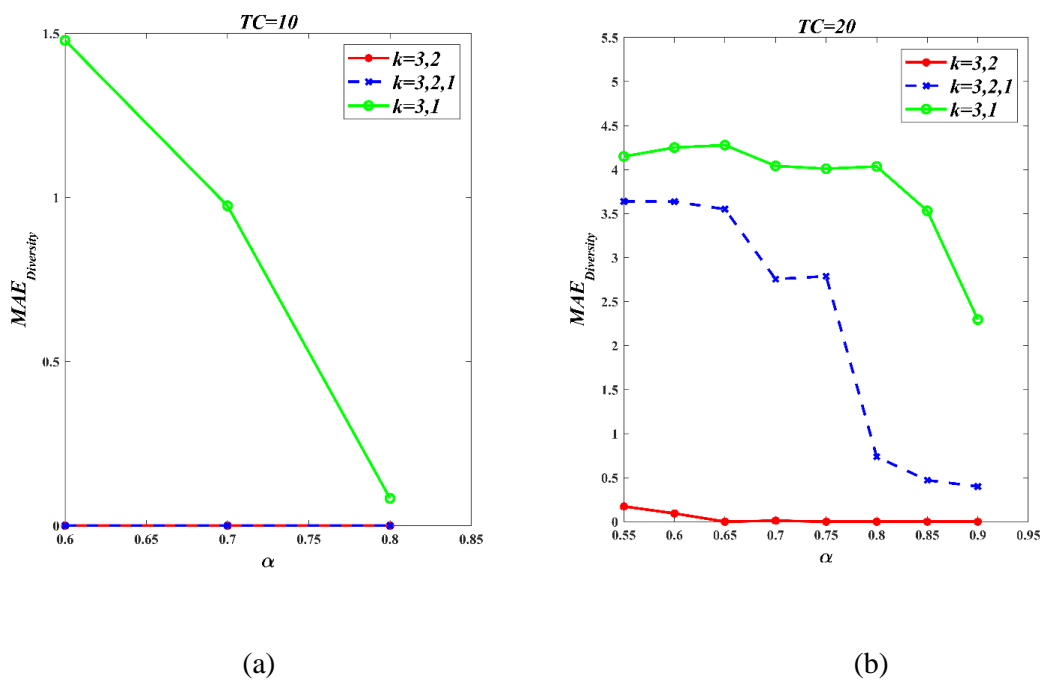


Figure 4. Diversity error analysis for (a) $TC = 10$ and (b) $TC = 20$ with different k .

Table 7. Friedman test for Diversity between different k .

α	$k = 3,2$	$k = 3,2,1$	$k = 3,1$
0.55	0.1734	3.6378	4.1496
0.60	0.0936	3.6358	4.2504
0.65	0	3.55	4.278
0.70	0.0108	2.756	4.0416
0.75	0	2.7878	4.01
0.80	0	0.737	4.0352
0.85	0	0.4704	3.5316
0.90	0	0.399	2.2960
Chi-Square, χ^2	16		
p – Value	0.0003355		
Accept/Reject H_0	Reject H_0		

7.2. Testing phase

This section discusses the performance of this model with different k in the testing phase. The testing capability, energy profile and neuron variation are considered with the aim that DHNN-MAJ3SATEA is able to achieve global minimum energy.

7.2.1. Testing analysis

Figure 6 demonstrates the performance of the testing phase in DHNN-MAJ3SATEA. It tells the value of neurons that are able to escape local minimum energy. Since $k = 3,2$ has the best learning phase, it also will have the best testing phase. $k = 3,2$ able to produce the most global minimum solution. The majority patterns that $\varphi_{MAJ3SAT}$ possessed succeeded in improving the model and generating more global minimum energy solutions even though NN is rising. This is due to optimal energy management for $k = 3,2$. Also, HTAF manages to effectively squash the energy of the neurons due to its steep sigmoid function. Even if the first-order logic is present in $k = 3,2,1$, the generated neurons can induce global minima states. The correct pattern of $\varphi_{MAJ3SAT}$ can generate more correct final neuron states. The relaxation phase, local field and HTAF seem able to stabilize the neuron oscillation providing chances for the neuron to escape suboptimal conditions. The energy of the neuron for $k = 3,1$ is the highest. This is because the disruption in obtaining a satisfied interpretation has affected the energy management for this logical structure causing high oscillating neurons. The neurons are not able to maintain within the tolerance area causing them to trap in suboptimal conditions. Note that as α increases the NN will also increase. Therefore, based on the study [8], higher NN will cause a computational burden to the model dropping the model's performance. However, in this study, the performance of the model is improving with the help of HTAF effectively improving the solution and decreasing the number of neurons that suffers from the suboptimal condition. This causes lower testing energy errors due to better energy management during the testing phase. The higher the ratio, the more neurons are able to acquire global minimum energy. As expected for $TC = 10$, $k = 3,2$ and $k = 3,2,1$ able to achieve $Z_m = 1$ for all α . Note here for all cases the Z_m are increasing as α increases. The majority patterns that $\varphi_{MAJ3SAT}$ possessed succeeded in improving the model and generating more global minimum energy solutions even though NN is rising. The local field did not suffer from the computational burden and is able to generate optimal final neuron states. Figure 7 presents the energy error analysis for different $TC = 10$ and $TC = 20$ with different k . The energy error analysis is based on 5 CAM. For $0.6 \leq \alpha \leq 0.8$ in $TC = 10$, $k = 3,2$ and $k = 3,2,1$ able to achieve $MAE_{Energy} = 0$. This indicates that all of the neurons are able to achieve global minimum energy thanks to the relaxation operators that this model proposed. It also shows that the local field tends to generate optimal neuron states. This highlights the capability of $\varphi_{MAJ3SAT}$ when the majority ratio is increasing, more final neuron states are able to achieve global minimum energy. We can prove this with $k = 3,1$. The model loses its efficiency a little bit as the TC number is doubled. The computational burden that higher NN has produced is affecting the neuron stabilizing process. Therefore, more neurons are trapped in a suboptimal energy state. This can be seen for $k = 3,2,1$ and $k = 3,1$. In addition, the Friedman test analysis for global minimum energy is displayed in Table 8. This Friedman test shows that the Chi-Square for $k = 3,2$, $k = 3,2,1$ and $k = 3,1$ for top 1 until 5 is 12, 16, 16, 15.0625, 13.5625 respectively and the degree of freedom is $df = 2$. The null hypothesis is rejected for all top considering $\alpha_0 = 0.05$. This again highlighted the significance of analyzing different k in this study.

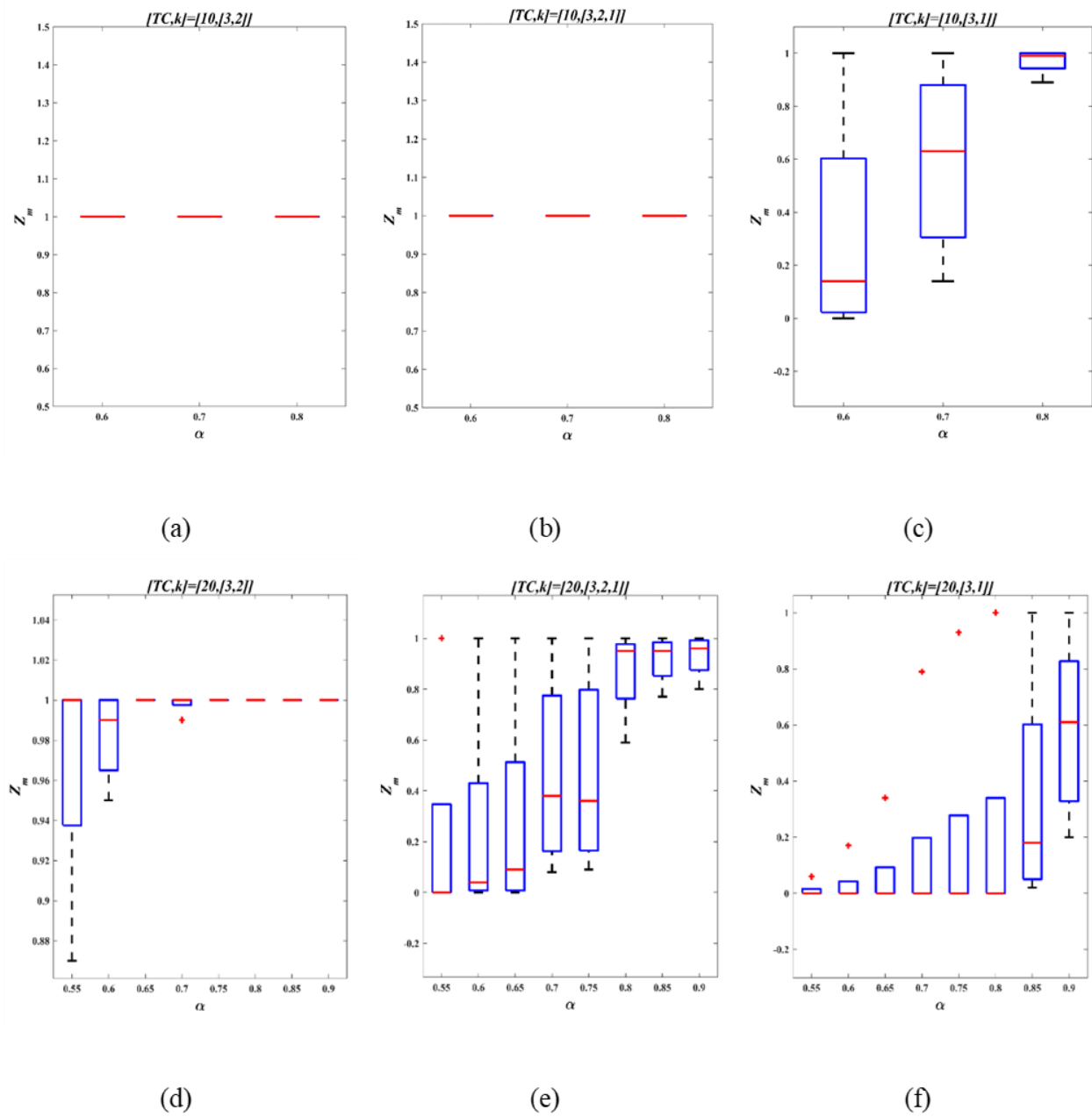


Figure 5. Global minima ratio, Z_m for (a) $[TC, k] = [10, [3, 2]]$, (b) $[TC, k] = [10, [3, 2, 1]]$, (c) $[TC, k] = [10, [3, 1]]$, (d) $[TC, k] = [20, [3, 2]]$, (e) $[TC, k] = [20, [3, 2, 1]]$, (f) $[TC, k] = [20, [3, 1]]$.

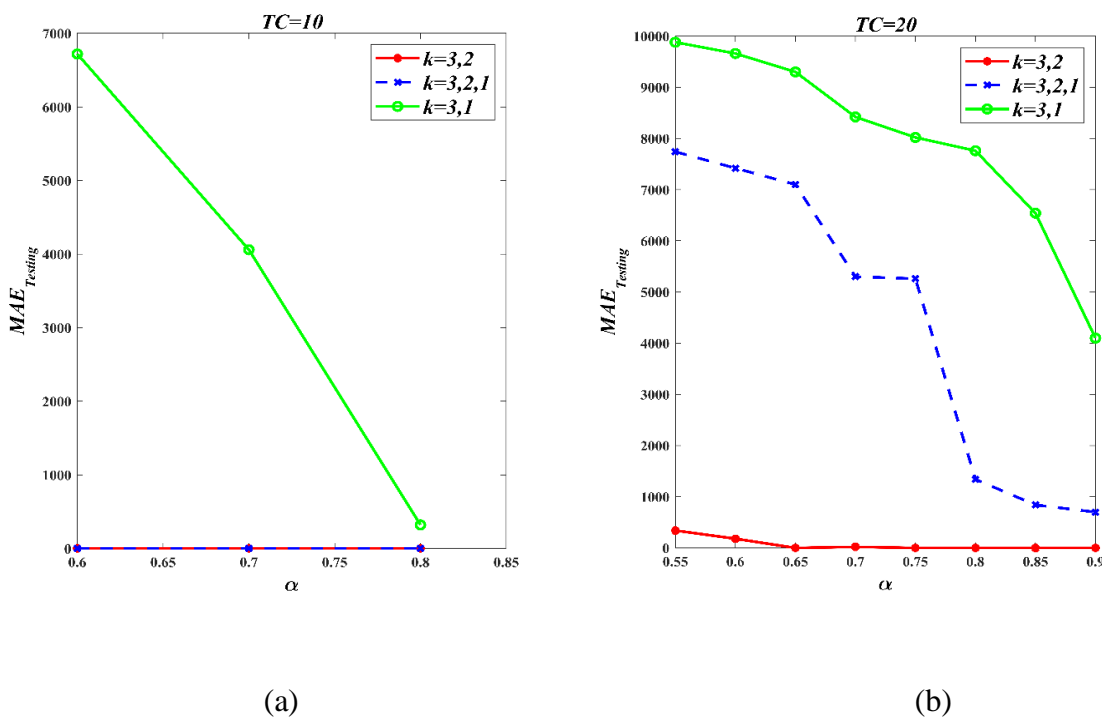


Figure 6. Testing error analysis for (a) $TC = 10$ and (b) $TC = 20$ with different k .

Table 8. Friedman test for Z_m between different k .

Top	Chi-Square, χ^2	p – Value	Accept/Reject H_0
1	12	0.002479	Reject H_0
2	16	0.0003355	Reject H_0
3	16	0.0003355	Reject H_0
4	15.0625	0.0005361	Reject H_0
5	13.5625	0.001135	Reject H_0

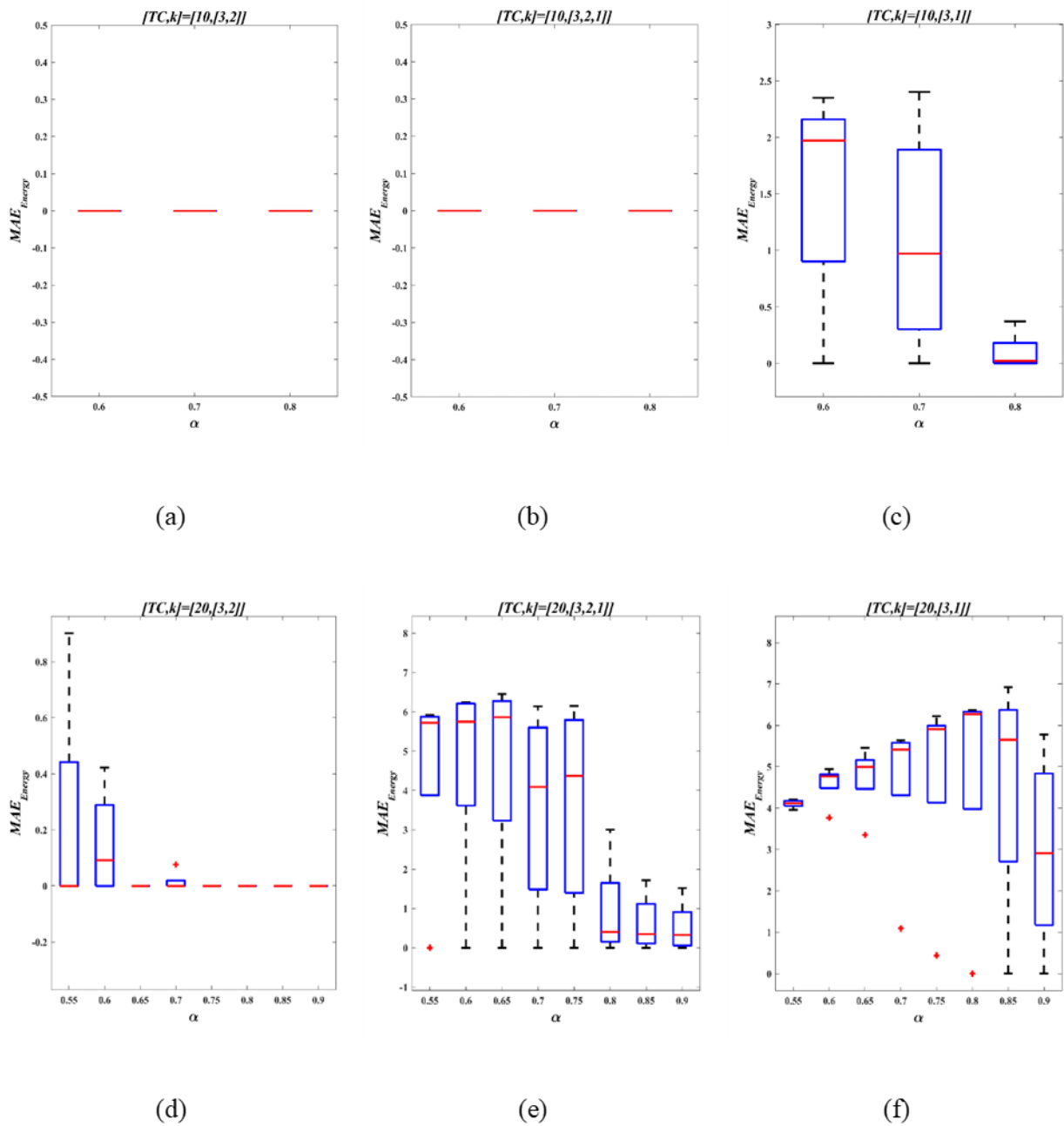


Figure 7. Energy error analysis for (a) $[TC, k] = [10, [3, 2]]$, (b) $[TC, k] = [10, [3, 2, 1]]$, (c) $[TC, k] = [10, [3, 1]]$, (d) $[TC, k] = [20, [3, 2]]$, (e) $[TC, k] = [20, [3, 2, 1]]$, (f) $[TC, k] = [20, [3, 1]]$.

7.2.2. Testing analysis in total variation and similarity analysis

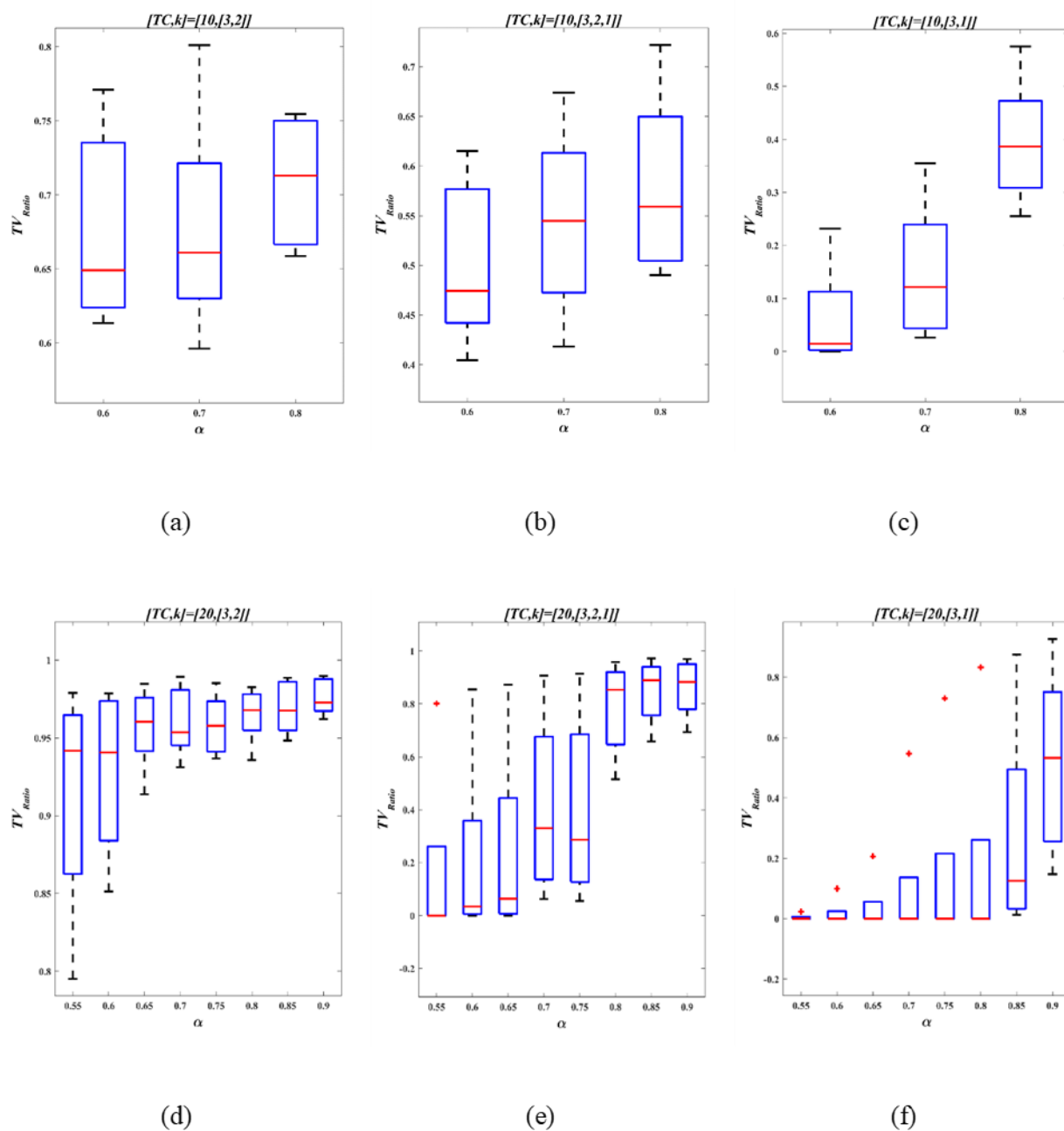
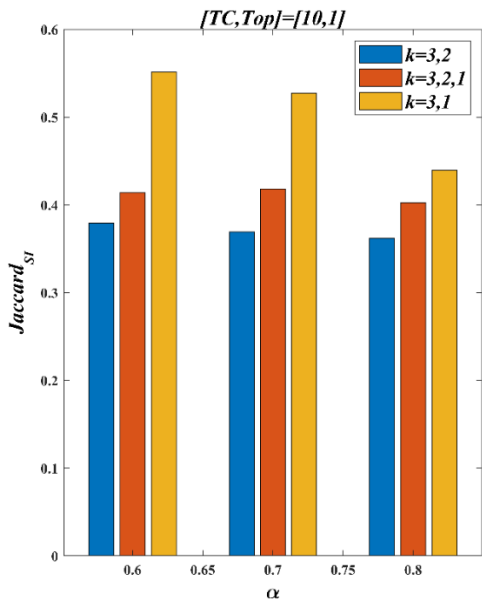
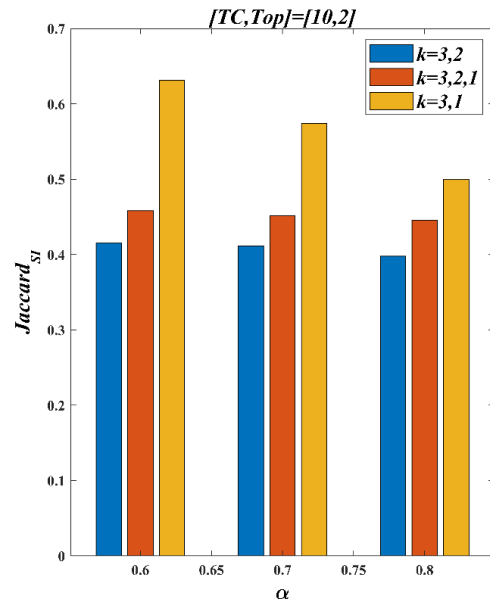


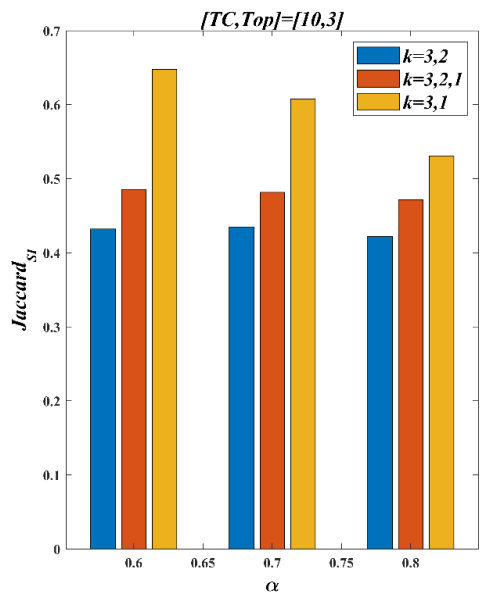
Figure 8. Total Neuron Variation for (a) $[TC, k] = [10, [3, 2]]$, (b) $[TC, k] = [10, [3, 2, 1]]$, (c) $[TC, k] = [10, [3, 1]]$, (d) $[TC, k] = [20, [3, 2]]$, (e) $[TC, k] = [20, [3, 2, 1]]$, (f) $[TC, k] = [20, [3, 1]]$.



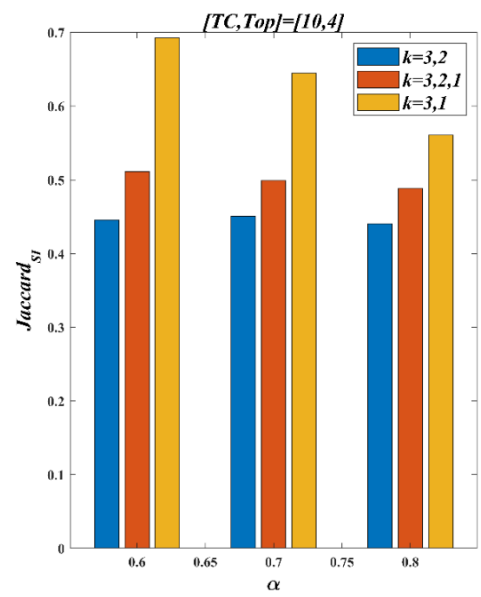
(a)



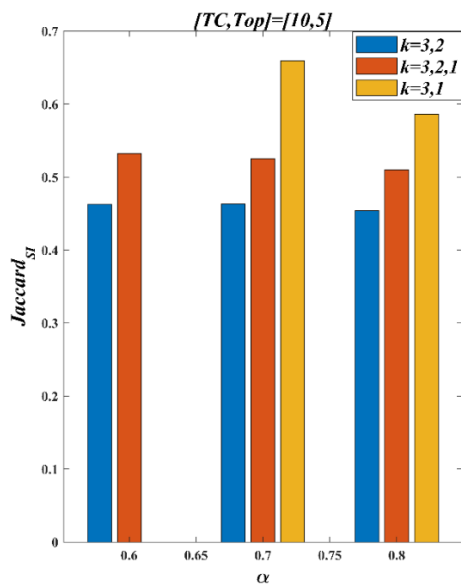
(b)



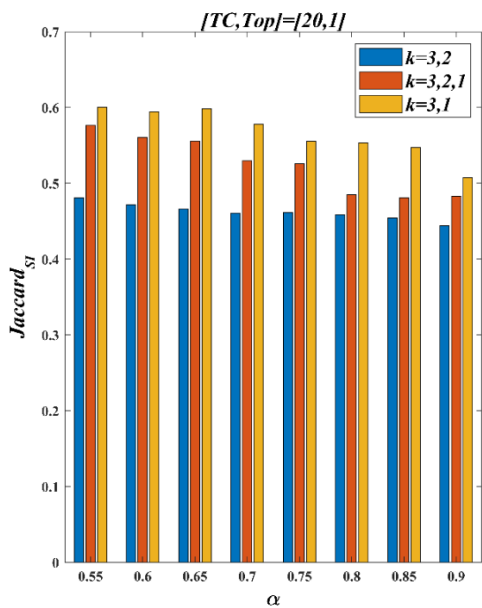
(c)



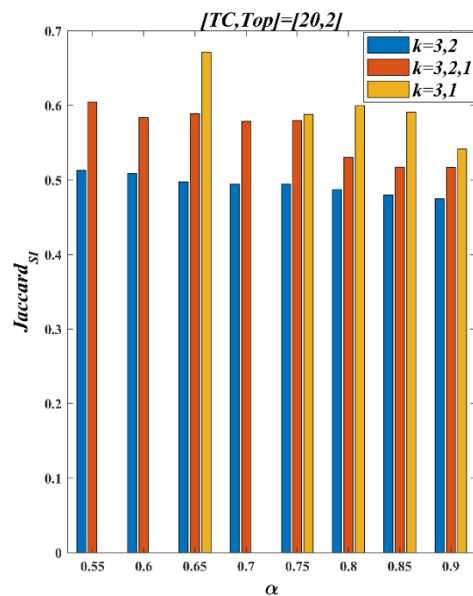
(d)



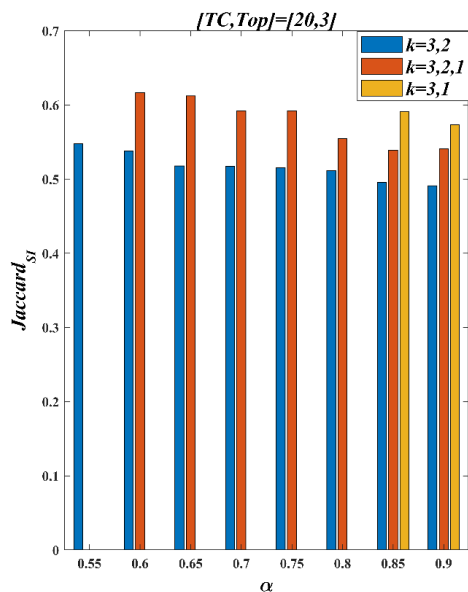
(e)



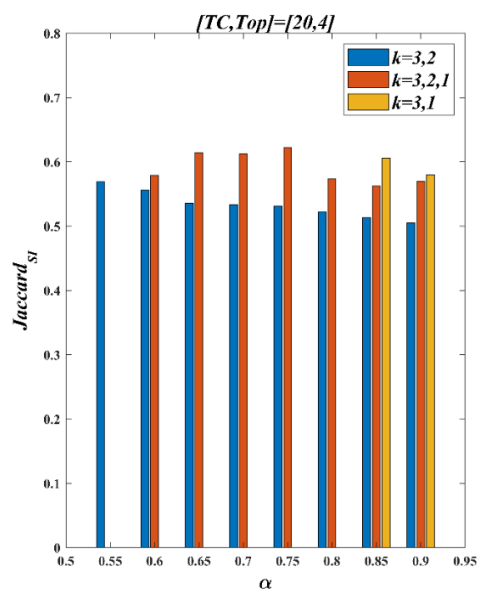
(f)



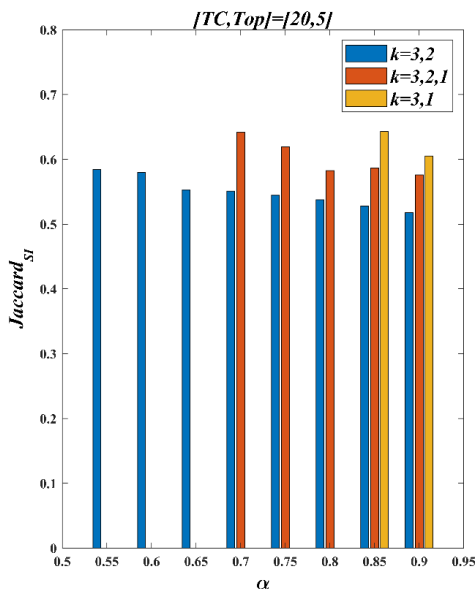
(g)



(h)

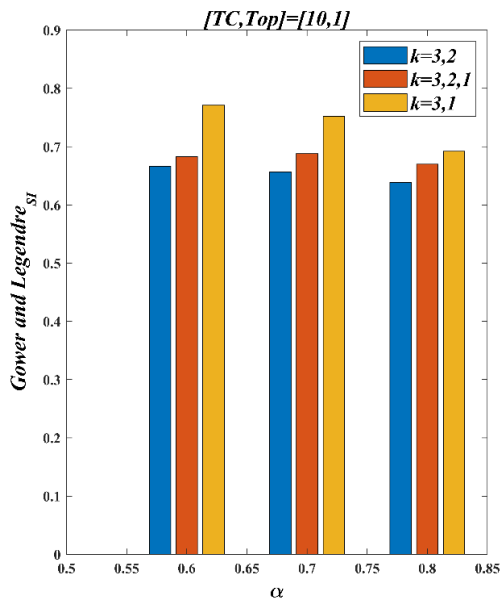


(i)

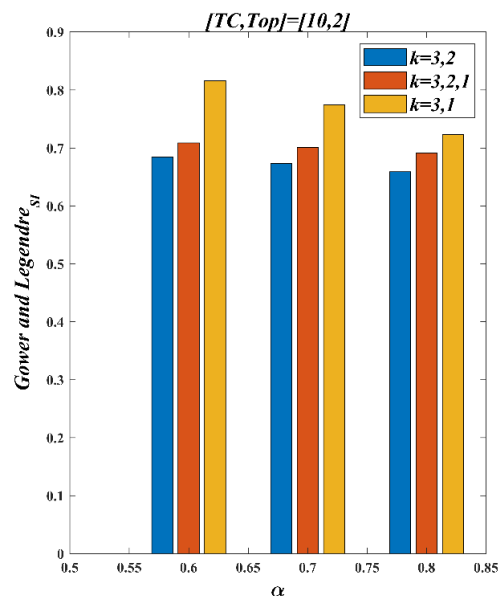


(j)

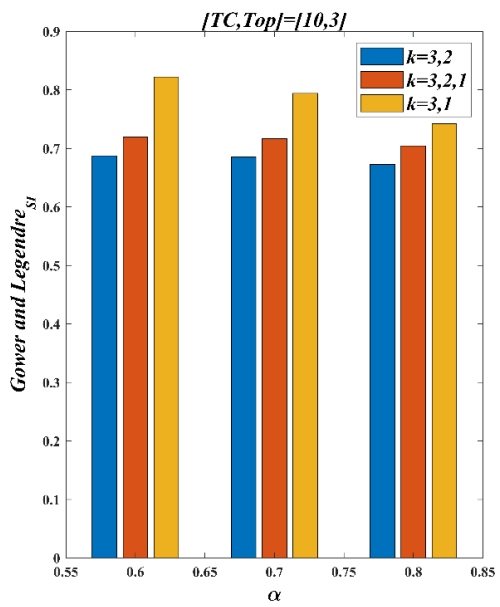
Figure 9. Jaccard Similarity Index (JSI) for (a) $[TC, Top] = [10,1]$, (b) $[TC, Top] = [10,2]$, (c) $[TC, Top] = [10,3]$, (d) $[TC, Top] = [10,4]$, (e) $[TC, Top] = [10,5]$, (f) $[TC, Top] = [20,1]$, (g) $[TC, Top] = [20,2]$, (h) $[TC, Top] = [20,3]$, (i) $[TC, Top] = [20,4]$, (j) $[TC, Top] = [20,5]$.



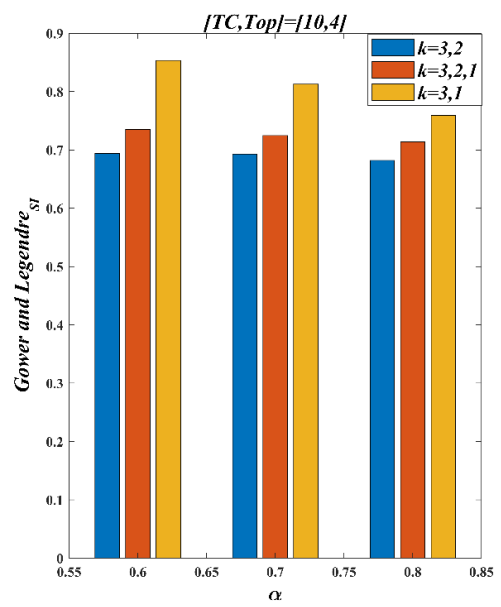
(a)



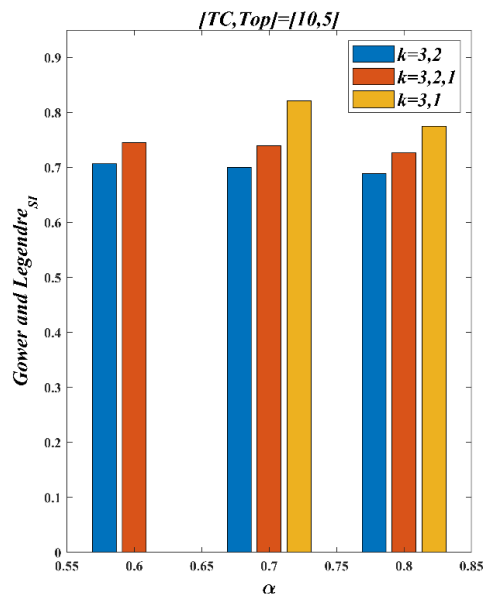
(b)



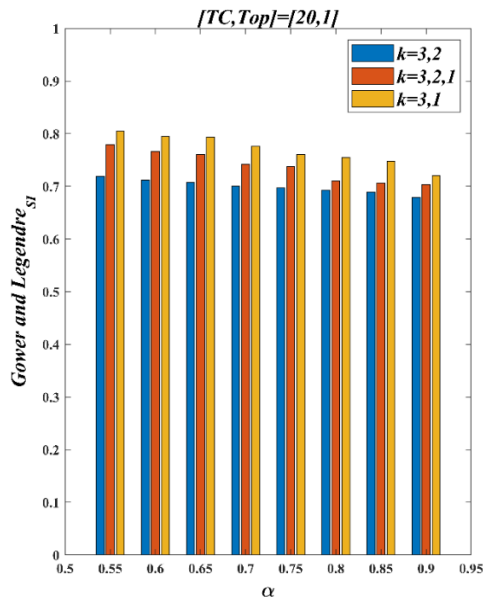
(c)



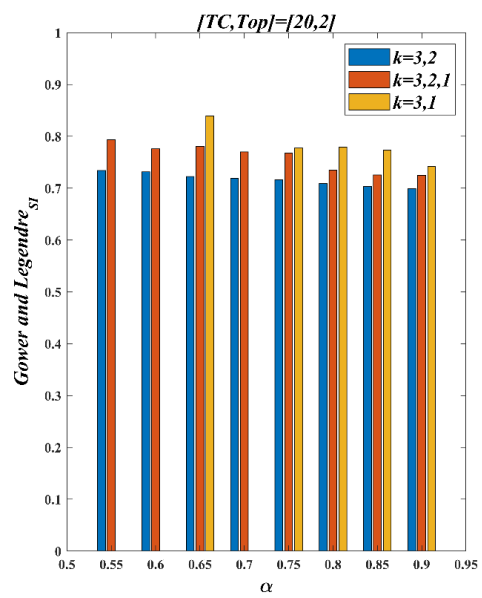
(d)



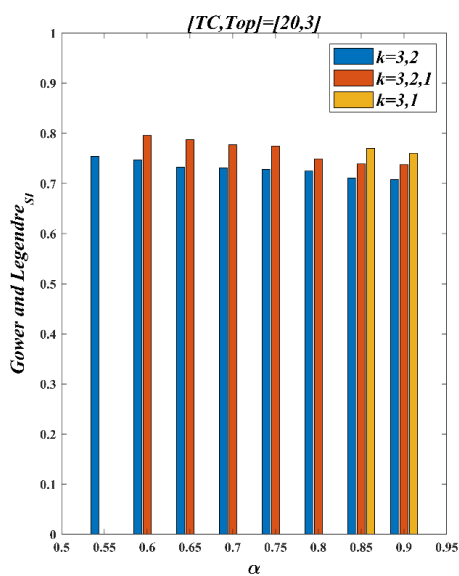
(e)



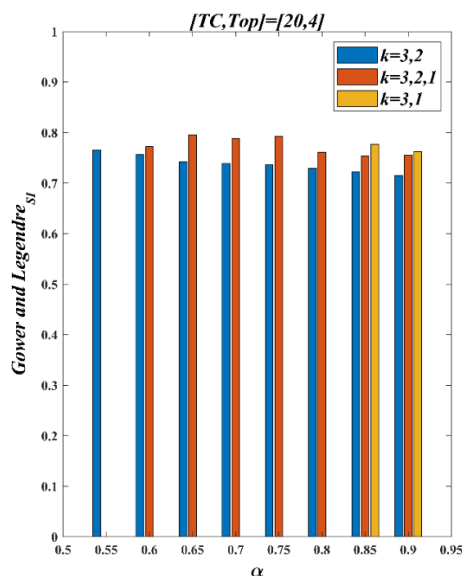
(f)



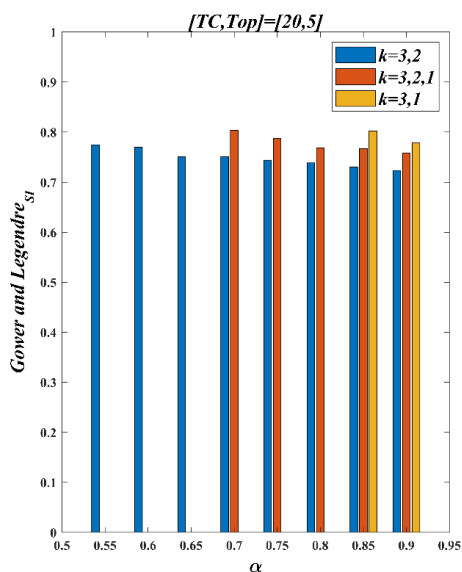
(g)



(h)



(i)



(j)

Figure 10. Gower and Legendre Similarity Index for (a) $[TC, Top] = [10, 1]$, (b) $[TC, Top] = [10, 2]$, (c) $[TC, Top] = [10, 3]$, (d) $[TC, Top] = [10, 4]$, (e) $[TC, Top] = [10, 5]$, (f) $[TC, Top] = [20, 1]$, (g) $[TC, Top] = [20, 2]$, (h) $[TC, Top] = [20, 3]$, (i) $[TC, Top] = [20, 4]$, (j) $[TC, Top] = [20, 5]$.

This part analyzed the behavior of the solutions and their similarity of the solution. Figure 8

describes the variation of the solution generated by DHNN-MAJ3SATEA based on 5 CAM. An optimal solution is when the model is able to retrieve non-repeating final neuron states. JSI in Figure 9 analyzed the condition of the $\{1,1\}$ states to the other states excluding the $\{-1,-1\}$ state while GLSI in Figure 10 includes the $\{-1,-1\}$ states. Figure 9 shows that $k = 3,2$ able to achieve an ideal performance of final neuron states as it holds the lowest JSI value compared to the other two k . The non-systematic structure that $\varphi_{MAJ3SAT}$ possesses is able to generate more satisfied different interpretations. More satisfied interpretations lead to a wide range of solutions space. The performance of $\varphi_{MAJ3SAT}$ for all k are proportional to Figure (8-10). In $TC = 20$, TV ranges from 0.79 to 0.97 for $k = 3,2$ which shows prominent results. While it is ranging from 0 to 0.97 for $k = 3,1$. $k = 3,1$ did not manage to produce more variety of neurons due to first-order logic that hinders the satisfaction of the logical string. The neurons are trapped in a suboptimal state preventing more variety of neurons to be generated. Figure 10 also shows that $k = 3,1$ suffers low neuron variations. EA is able to improve the learning process, but EA tends to generate over-fitting solutions for $k = 3,1$. Causing a high number of GLSI. The same thing occurs in Figure 9 which $\{-1,-1\}$ are not considered. As α increases, the neuron variation and JSI value also increase. This portrays how the majority patterns that $\varphi_{MAJ3SAT}$ possess the ability to produce solutions with different dimensions. It gives higher chances for the logical string to be satisfied as α increases, conjuring more neurons able to achieve global minimum energy. $k = 3,2$ also seems able to reduce the number of repetitive solutions followed by $k = 3,2,1$. Some top for lower α in $TC = 20$, some k is not able to generate even top 5 solutions. This is due to the poor diversity performance that it possesses during the learning phase which is due to bad synaptic weight management that k has (especially $k = 3,1$). This can be seen clearly in the fifth top in both Figures 9 and 10 where $k = 3,2$ able to generate value for each α while $k = 3,2,1$ only able to generate value starting from $\alpha = 0.7$ and $k = 3,1$ able generate value starting from $\alpha = 0.85$. Due to its suboptimal condition $k = 3,2,1$ and $k = 3,1$ was not able to generate even the top five different satisfied interpretations. This concludes how bad the impact of the first-order logic has provided causing over-fitting solutions and low diverse solutions.

7.3. The limitation of the DHNN-MAJ3SATEA

Upon experimenting with this model, there are several limitations of this study that need to be mentioned. Major 3 Satisfiability is a non-systematic logic; the model will need to deal with multi-clauses logical structures with different order combinations which include the first-order logic which has a very low probability to be satisfied. MAJ3SAT logical structure are dealing with random negated literals per clause, so there might exist clauses with no negated literals which will decrease the diversity of the solution. Next, since MAJ3SAT is focusing on the majority of the 3SAT clauses, the neuron number will also increase rapidly causing a high computational burden. On the other hand, DHNN utilizes bipolar representation, which implies that this model cannot use for the continuous problem. This study also deals with single-layer neural networks. Hence, this model is not suitable for deep learning optimization. In addition, this study integrated EA in DHNN. EA is known as a strict metaheuristic which means that if EA already has a neuron with max fitness, the neuron still needs to undergo all the stages in EA. It has multiple optimization operators which consist of local and global operators. These stages and operators in EA will increase the computational time for the learning process as it gets complex. This study also did not apply the chaotic positive advertisement and migration operator parameter that [15] has proposed and the caretaker party that [6] has proposed.

8. Conclusions

The presented findings have proven that $\varphi_{MAJ3SAT}$ are compatible with DHNN that have been assisted by Election Algorithm in the learning phase. The majority pattern that $\varphi_{MAJ3SAT}$ has provided succeeded in increasing satisfied interpretations. In other words, $\varphi_{MAJ3SAT}$ are able to generate more correct synaptic weight during the learning phase due to the majority factor that it possessed. This model has been optimal in terms of synaptic weight management, cost function minimization, diversity, energy profile and neuron variation.

During the learning phase, MAJ3SAT $k = 3,2$ has the best learning performance. This logical combination can maximize the satisfied interpretation and minimize the cost function. This happens also for higher NN ($TC = 20$). $k = 3,2$ outperformed other logical combinations in terms of diversity. The absence of first-order logic has provided an advantage to $\varphi_{MAJ3SAT}$. It provides more room for correct synaptic weight to be generated causing more diverse neurons to be generated. Next, during the testing phase, $k = 3,2$ also seems to have the best testing phase. DHNN-MAJ3SATEA succeed to attain more stable final neuron states. This shows that neurons that suffer high oscillations are more for $k = 3,2,1$ and $k = 3,1$ compared to $k = 3,2$ leading to local minimum energy. Furthermore, $k = 3,2$ also outperformed other logical combinations in terms of neuron variation. Generally, for all k , the performance of DHNN-MAJ3SATEA increases as α increases. DHNN-MAJ3SATEA manages to provide a higher satisfied interpretation during logical inconsistency. To sum up, the objective that has been stated in Section 1, this study succeeds in formulating the logical structure of MAJ3SAT. This study is also able to imply the multi-objective Election Algorithm in this model and succeeds in minimizing the cost functions. The multi-objective functions are also able to optimize the solutions' fitness and diversity while expanding the storage capacity. The performance and the compatibility of the model have also shown prominent results.

From the obtained experimental results, this model can be extended to real-life data sets for future work and can decrease its limitation by adding certain features to it. For future possible work of $\varphi_{MAJ3SAT}$, the logical structure can be extended to control its negated literals. Therefore, a weighted major logic can be proposed named Weighted Major 3 Satisfiability logic ($rMAJ3SAT$) as a weighted factor that can increase the diversity of the solution [27]. In addition, the fuzzy logic that [28] used in this study can be merged with MAJ3SAT in order to explore MAJ3SAT's continuous search space in the learning phase. Since we already observe how the majority of clauses of the logical structure have performed, maybe a new logical structure that highlights the minority characteristics can be explored. Besides, HEA that [6] has proposed also can be used as a substitution rather than classic EA. It is also interesting if Arithmetic Optimization Algorithm (AOA) that [29] proposed can be used as the learning algorithm. This algorithm operates based on the behavior of arithmetic operators, such as multiplication, division, subtraction and addition. A powerful learning algorithm called Firefly Algorithm (FA) can also be implemented [30]. FA is widely used in solving complex scheduling problems [30]. Another algorithm that can be integrated is the Bat Algorithm (BA) [31]. BA is a renowned metaheuristic that utilizes the echolocation feature of a bat and has been successfully applied to a variety of optimization issues. The classic DHNN also can be replaced with Rotor Hopfield Neural Network (RHNN) for further observation [32]. Since this model is using simulated data only, the data that are analyzed are synthetic. For its applications, this model can be further embedded in the logic mining field where it can filter the best logical rules that are suitable for the data [27,33]. Since it has great forecasting and classification ability, this model is good for CSI indoor fingerprint localization

[34], weather forecasting [35] and unmanned aerial vehicle (UAV) [36].

Use of AI tools declaration

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

Acknowledgments

Warmest acknowledgment to the Ministry of Higher Education Malaysia for the financial support of the Fundamental Research Scheme Grant (FRGS) with Project Code: FRGS/1/2022/STG06/USM/02/6 and Universiti Sains Malaysia.

Conflict of Interest

All authors declare no conflicts of interest in this paper. All authors approved the version of the manuscript to be published.

References

1. M. Abdallah, M. A. Talib, S. Feroz, Q. Nasir, H. Abdalla, B. Mahfood, Artificial intelligence applications in solid waste management: A systematic research review, *Waste Manage.*, **109** (2020), 231–246. <https://doi.org/10.1016/j.wasman.2020.04.057>
2. S. Agatonovic-Kustrin, R. Beresford, Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research, *J Pharm. Biomed. Anal.*, **22** (2000), 717–727. [https://doi.org/10.1016/s0731-7085\(99\)00272-1](https://doi.org/10.1016/s0731-7085(99)00272-1)
3. J. J. Hopfield, D. W. Tank, “Neural” computation of decisions in optimization problems, *Biol. Cybern.*, **52** (1985), 141–152. <https://doi.org/10.1007/bf00339943>
4. W. A. T. W. Abdullah, Logic programming on a neural network, *Int. J. Intell. Syst.*, **7** (1992), 513–519. <https://doi.org/10.1002/int.4550070604>
5. A. Alway, N. E. Zamri, S. A. Karim, M. A. Mansor, M. S. M. Kasihmuddin, M. M. Bazuhair, Major 2 satisfiability logic in discrete Hopfield neural network, *Int. J. Comput. Math.* **99** (2022), 924–948. <https://doi.org/10.1080/00207160.2021.1939870>
6. S. A. Karim, M. S. M. Kasihmuddin, S. Sathasivam, M. A. Mansor, S. Z. M. Jamaludin, M. R. Amin, A Novel Multi-Objective Hybrid Election Algorithm for Higher-Order Random Satisfiability in Discrete Hopfield Neural Network, *Mathematics*, **10** (2022), 1963. <https://doi.org/10.3390/math10121963>
7. M. M. Bazuhair, S. Z. M. Jamaludin, N. E. Zamri, M. S. M. Kasihmuddin, M. A. Mansor, A. Alway, S. A. Karim, Novel Hopfield Neural Network Model with Election Algorithm for Random 3 Satisfiability, *Processes*, **9** (2021), 1292. <https://doi.org/10.3390/pr9081292>
8. M. S. M. Kasihmuddin, M. A. Mansor, S. Sathasivam, Hybrid Genetic Algorithm in the Hopfield Network for Logic Satisfiability Problem, *Pertanika J. Sci. Technol.*, **25** (2017), 139–152. <https://doi.org/10.1063/1.4995911>

9. M. A. Mansor, M. S. M. Kasihmuddin, S. Sathasivam, Artificial Immune System Paradigm in the Hopfield Network for 3-Satisfiability Problem, *Pertanika J. Sci. Technol.*, **25** (2017), 1173–1188.
10. N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, A. Alway, S. Z. M. Jamaludin, S. A. Alzaeemi, Amazon employees resources access data extraction via clonal selection algorithm and logic mining approach, *Entropy*, **22** (2020), 596. <https://doi.org/10.3390/e22060596>
11. S. A. Karim, N. E. Zamri, A. Alway, M. S. M. Kasihmuddin, A. I. M. Ismail, M. A. Mansor, et al., Random satisfiability: A higher-order logical approach in discrete Hopfield Neural Network, *IEEE Access*, **9** (2021), 50831–50845. <https://doi.org/10.1109/access.2021.3068998>
12. E. Pacuit, S. Salame, Majority Logic, *KR*, **4** (2004), 598–605.
13. Z. Zheng, S. Yang, Y. Guo, X. Jin, R. Wang, Meta-heuristic Techniques in Microgrid Management: A Survey, *Swarm Evol. Comput.*, **78** (2023), 101256. <https://doi.org/10.1016/j.swevo.2023.101256>
14. H. Emami, F. Derakhshan, Election algorithm: A new socio-politically inspired strategy, *AI Commun.*, **28** (2015), 591–603. <https://doi.org/10.3233/aic-140652>
15. H. Emami, Chaotic election algorithm, *Comput. Inform.* **38** (2019), 1444–1478. https://doi.org/10.31577/cai_2019_6_1444
16. S. Sathasivam, M. A. Mansor, M. S. M. Kasihmuddin, H. Abubakar, Election algorithm for random k satisfiability in the Hopfield neural network, *Processes*, **8** (2020), 568. <https://doi.org/10.3390/pr8050568>
17. B. F. B. A. Boya, B. Ramakrishnan, J. Y. Effa, J. Kengne, K. Rajagopal, Effects of bias current and control of multistability in 3D hopfield neural network, *Heliyon*, **9** (2023), 13034. <https://doi.org/10.1016/j.heliyon.2023.e13034>.
18. S. Z. M. Jamaludin, N. A. Romli, M. S. M. Kasihmuddin, A. Baharum, M. A. Mansor, M.F. Marsani, Novel logic mining incorporating log linear approach, *J. King Saud Univ. Comput. Inform. Sci.*, **34** (2022), 9011–9027.
19. M. S. M. Kasihmuddin, S. Z. M. Jamaludin, M. A. Mansor, H. A. Wahab, S. M. S. Ghadzi, Supervised learning perspective in logic mining, *Mathematics*, **10** (2022), 915.
20. Y. Guo, M. S. M. Kasihmuddin, Y. Gao, M. A. Mansor, H. A. Wahab, N. E. Zamri, et al., YRAN2SAT: A novel flexible random satisfiability logical rule in discrete hopfield neural network, *Adv. Eng. Software*, **171** (2022), 103169. <https://doi.org/10.1016/j.advengsoft.2022.103169>
21. N. Bacanin, C. Stoean, M. Zivkovic, M. Rakic, R. Strulak-Wójcikiewicz, R. Stoean, On the Benefits of Using Metaheuristics in the Hyperparameter Tuning of Deep Learning Models for Energy Load Forecasting, *Energies*, **16** (2023), 1434. <https://doi.org/10.3390/en16031434>
22. S. Z. M. Jamaludin, M. S. M. Kasihmuddin, A. I. M. Ismail, M. A. Mansor, M.F.M. Basir, Energy based logic mining analysis with hopfield neural network for recruitment evaluation. *Entropy*, **23** (2020), 40. <https://doi.org/10.3390/e23010040>
23. J. Chen, M. S. M. Kasihmuddin, Y. Gao, Y. Guo, M. A. Mansor, N.A. Romli, et al., PRO2SAT: Systematic Probabilistic Satisfiability logic in Discrete Hopfield Neural Network, *Adv. Eng. Software*, **175** (2023), 103355. <https://doi.org/10.1016/j.advengsoft.2022.103355>
24. L. C. Kho, M. S. M. Kasihmuddin, M. A. Mansor, S. Sathasivam, Logic Mining in League of Legends, *Pertanika J. Sci. Technol.*, **28** (2020), 211–225.

25. N. Khentout, G. Magrotti, Fault supervision of nuclear research reactor systems using artificial neural networks: A review with results, *Ann. Nucl. Energy*, **185** (2023), 109684. <https://doi.org/10.1016/j.anucene.2023.109684>
26. N. Kanwisher, M. Khosla, K. Dobs, Using artificial neural networks to ask ‘why’ questions of minds and brains, *Trends Neurosci.*, **46** (2023), 240–254. <https://doi.org/10.1016/j.tins.2022.12.008>
27. S. S. M. Sidik, N. E. Zamri, M. S. M. Kasihmuddin, H. A. Wahab, Y. Guo, M. A. Mansor, Non-Systematic Weighted Satisfiability in Discrete Hopfield Neural Network Using Binary Artificial Bee Colony Optimization, *Mathematics*, **10** (2022), 1129. <https://doi.org/10.3390/math10071129>
28. S. Subiyanto, A. Mohamed, M. A. Hannan, Intelligent maximum power point tracking for PV system using Hopfield neural network optimized fuzzy logic controller, *Energ. Buildings*, **51** (2012), 29–38.
29. L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Method. Appl. M.*, **376** (2021), 113609. <https://doi.org/10.1016/j.cma.2020.113609>
30. Z. Wang, L. Shen, X. Li, L. Gao, An improved multi-objective firefly algorithm for energy-efficient hybrid flowshop rescheduling problem, *J. Clean. Produc.*, **385** (2023), 135738. <https://doi.org/10.1016/j.jclepro.2022.135738>
31. W. H. Bangyal, A. Hameed, J. Ahmad, K. Nisar, M. R. Haque, A. A. A. Ibrahim, et al., New modified controlled bat algorithm for numerical optimization problem, *Comput., Mater. Con.*, **70** (2022), 2241–2259.
32. M. Kobayashi, Quaternion projection rule for rotor hopfield neural networks, *IEEE T. Neur. Net. Lear.*, **32** (2020), 900–908.
33. M. Safavi, A. K. Siuki, S. R. Hashemi, New optimization methods for designing rain stations network using new neural network, election, and whale optimization algorithms by combining the Kriging method, *Environ. Monit. Assess.*, **193** (2021), 4. <https://doi.org/10.1007/s10661-020-08726-z>
34. X. Dang, X. Tang, Z. Hao, J. Ren, Discrete Hopfield neural network based indoor Wi-Fi localization using CSI, *EURASIP J. Wirel. Commun.*, **2020** (2020), 76.
35. G. J. Sawale, S. R. Gupta, Use of artificial neural network in data mining for weather forecasting, *Int. J. Comput. Sci. Appl.*, **6** (2013), 383–387.
36. Z. Zhang, L. Zheng, Y. Zhou, Q. Guo, A Novel Finite-Time-Gain-Adjustment Controller Design Method for UAVs Tracking Time-Varying Targets, *IEEE T. Intell. Transport. Syst.*, **23** (2021), 12531–12543.



AIMS Press

© 2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)