Optimal representation to High Order Random Boolean kSatisability via Election Algorithm as Heuristic Search Approach in Hopeld Neural Networks

Hamza Abubakar[a,b,*], Abdu Sagir Masanawa[c], Surajo Yusuf[b], G. I. Boaku[c]

[a]*School of Mathematical Sciences, Universiti Sains Malaysia*
[b]*Department of Mathematics, Isa Kaita College of Education, P.M.B 5007, Dutsin-Ma, Katsina, Nigeria*
[c]*Department of Mathematical Sciences, Federal Uniersity Dutsin-Ma, Katsina, Nigeria*

Abstract

This study proposed a hybridization of higher-order Random Boolean kSatisfiability (RANkSAT) with the Hopfield neural network (HNN) as a neuro-dynamical model designed to reflect knowledge efficiently. The learning process of the Hopfield neural network (HNN) has undergone significant changes and improvements according to various types of optimization problems. However, the HNN model is associated with some limitations which include storage capacity and being easily trapped to the local minimum solution. The Election algorithm (EA) is proposed to improve the learning phase of HNN for optimal Random Boolean kSatisfiability (RANkSAT) representation in higher order. The main source of inspiration for the Election Algorithm (EA) is its ability to extend the power and rule of political parties beyond their borders when seeking endorsement. The main purpose is to utilize the optimization capacity of EA to accelerate the learning phase of HNN for optimal random k Satisfiability representation. The global minima ratio (mR) and statistical error accumulations (SEA) during the training process were used to evaluate the proposed model performance. The result of this study revealed that our proposed EA-HNN-RANkSAT outperformed ABC-HNN-RANkSAT and ES-HNN-RANkSAT models in terms of mR and SEA. This study will further be extended to accommodate a novel field of Reverse analysis (RA) which involves data mining techniques to analyse real-life problems.

1. Introduction

The central motivation of many types of research carried out in the field of, machine learning, decision science and artificial neural networks (ANNs) is the large structure of their training stages. Many ANNs' complex training stage provide a powerful mechanism for solving optimization problems such as decision-making or classification tasks. Neural networks are used extensively in many fields of study that originated in mathematical neurobiology. This network attempts to simulate human brain capabilities. It has been utilized since the last decade as a theoretically sound alternative to traditional statistical models. The classification based on the neural networks model becomes efficient when applied in a hybrid framework with many forms of predictive models [16]. The inception of neural networks

---

*Corresponding author tel. no:
Email address:* zeeham4u2c@yahoo.com (Hamza Abubakar)

(ANNs) was initiated as variety of capable networks that act as a useful tool for specific tasks such as recognition problem [17], data mining [18], forecasting problems [21], prediction problems[14]. There have been many recent developments in an attempt to assemble different structures in refining the existing ANN models by integrating them with proficient searching techniques to intensify the quality of the standalone ANN framework [1].

Hopfield neural network (HNN) is a type of recurrent artificial network that mimics the operating capacity of human memory. The ability of HNN to manage nonlinear complex patterns by its training and testing capability is particularly useful for interpreting complex real-life computer science communities [8]. In this work, a novel and powerful heuristics search technique is known as Election algorithms has been explored for effective training of Hopfield neural network model. Election algorithms has been utilized in solving various mathematics and engineering benchmark problems which yield successful performance in both convergence rate and better identification of global optima. We can infer that the application of employing Election algorithms is endless; from industrial planning, scheduling, decision making and machine learning. Election algorithms have been utilized in [11] successfully for predicting groundwater level [12]. Recently, an election algorithm has been proposed in [20] to accelerate the learning phase of Hopfield neural network for random 2satisfiability. The performance of Election algorithms in solving the travelling salesmen problem has been investigated in [5]. Although Election Algorithm has been applied in various optimization areas, it recorded tremendous achievement in searching for the optimal solution to various combinatorial optimizations problems. In our work, Election algorithms has been applied on the trainig phase of hopfield network model in accelerating the training process for optimal representation (global minimum) solutions to high order logic (RANkSAT for k ≤ 3).

However, we did not come across any work that combined the global and local searching capacities of Election algorithms in enhancing the training phase of Hopfield neural network in searching for optimal representation of high order logic programming (RANkSAT). Therefore, a new hybrid computational technique has been proposed by integrating hybrid Election algorithm in the learning process of Hopfield neural network for optimal RANkSAT logical representation in achieving a efficiency, robustness and better accuracy. The contributions of this work include; (1) upgrading RAN2SAT to RAN3SAT; (2) implementing the new logical rule in the Hopfield neural network (RANkSAT-HNN);(3) incorporating Election algorithm (EA) in the training phase of the Hopfield neural network (HNN) for optimal random kSatisfiability and (4) performing comparison of the Election algorithm with other state-of-the-art searching methods in Hopfield neural network for Random kSatisfiability representation. The present work investigate the effectiveness of the Election algorithms in Hopfield neural network for RANkSAT will be explored by comparing its performances to the state-of-the-art model. By developing an effective intelligent working model based on Artificial neural networks, the proposed hybrid computational model will be useful to various

computational science and optimization communities by proving an alternative approach in doing computation. The structure of this work is as follows. In section 2, we present the proposed methods involving the mapping of RANkSAT as a new logical rule in Hopfield neural network and a heuristic search method in the Hopfield neural network learning phase via Election algorithms. Section 3 reports the models' performance evaluation metrics. The experimental results of their discussions are reported in section 4. The last Section concludes the paper.

2. Materials and Methods

2.1. Random kSatisfiability ($k \leq 3$)

Propositional satisfiability Logic can be perceived as a logical rule that consists of clauses that contain literals or variables. RANkSAT belongs to the family of non-systematic Boolean logical clauses in which each logical clause involves a random number of literals [13]. The general description for the formulation $F_{RANkSAT}$ is presented in Equation (1) as follows.

$$F_{RANkSAT} = \bigwedge_{i=0}^{t} C_i^{(3)} \bigwedge_{i=0}^{j} C_i^{(2)} \bigwedge_{i=0}^{d} C_i^{(1)} \, (k \leq 3) \qquad (1)$$

where $t, j, d \in [1, 2, ..., N]$, $t > 0$, $j > 0$ and $d > 0$. The clause $F_{RAN3SAT}$ is defined as a RANkSAT($k \leq 3$)for$C_i^{(k)}$ clauses described in Equation (2) as follows.

$$C_i^{(k)} = \begin{cases} (\alpha_i \vee \beta_i \vee \psi_i) \, , \, if \, k = 3 \\ (\alpha_i \vee \beta_i) \, , \quad if \, k = 2 \\ \tau_i, \quad\quad if \, k = 1 \end{cases} \qquad (2)$$

where the variables $\alpha_i \in [\alpha_i, \neg\alpha_i], \beta_i \in [\beta_i, \neg\beta_i], \tau_i \in [\tau_i, \neg\tau_i], \psi_i \in [\psi_i, \neg\psi_i]$ are defines literals and their negation in logical clauses respectively. In this work, we refer to F_r it as a Conjunctive Normal Form (CNF) formula in which the logic clauses in RANkSAT logic are chosen uniformly, independently among all $2^x \begin{pmatrix} d + j + t \\ y \end{pmatrix}$ without replacement a logic clauses of length$x$. We refer to the optimal mapping $y(F_x) \rightarrow [1, -1]$ as a logical interpretation expressed as 1 for (TRUE) and -1 otherwise. Logically, there are many formulations to RANkSAT logical clauses, one of the formations taking into account $k \leq 3$ is presented as follows.

$$F_{RANkSAT} = (\tau_1 \vee \neg\tau_2 \vee \tau_3) \wedge (\beta_1 \vee \beta_2 \vee \neg\beta_2) \wedge \neg\alpha_1 \, (3)$$

According to Equation (1),$F_{RANkSAT}$ comprises of $C_i^{(3)} = (\tau_1 \vee \neg\tau_2 \vee \tau_3)$, $C_2^{(2)} = (\neg\beta_1 \vee \beta_2)$and $C_1^{(1)} = \neg\alpha_1$. Therefore, the outcome of in Equation (3) is satisfied $F_{RANkSAT} = -1$ if $(\tau_1, \tau_2, \tau_3, \beta_1, \beta_2, \beta_2, \alpha_1) = (1, 1, 1, 1, 1, 1)$ 3 clauses are satisfied $\left(C_i^{(3)}, C_1^{(1)}, C_2^{(1)}\right)$. This research $F_{RANkSAT}$ will be embedded in Hopfield neural network via Election algorithm as a search technique in the proposed model.$F_{RANkSAT}$logic will modify the networks to represent the true structure or behaviour of the data involved since HNN is a non-symbolic platform, $F_{RANkSAT}$ serve as a symbolic mode representation that can be combined with these networks.

2.2. *Random kSatisfiability in Hopfield Neural Network*

Hopfield neural network (HNN) belongs to the family of a recurrent neural network that mimics the human biological brain [15]. Its structure consists of interconnected neurons and a powerful feature of content addressable memory that is crucial in solving various optimization and combinatorial tasks[13]. The network is composed of organised neurons of which is portrayed by a variable called the *Ising* variable. In bipolar recognition, the neurons in discrete HNN are used whereby $S_i \in [1, -1]$. The fundamental overview of neuron state activation in Hopfield neural networks is shown in Equation (4) below,

$$S_j = \begin{cases} 1 & , \quad if \ \sum_j W_{ij} S_j > \varsigma \\ -1 & , \quad Otherwise \end{cases} \tag{4}$$

where W_{ij} is the synaptic weight vector from starting from j neuron to i neuron. We defined S_i as the state of the neuron i in HNN and ς is the predefined value. The value of $\varsigma = 0.001$ has been specified in [6],[3],[20] to certify that the network's energy decreases to zero. The synaptic weight connection in the discrete HNN contains no connection with itself, the synaptic connection from one neuron to other neurons is zero. The model holds symmetrical features in terms of architecture. HNN model has similar intricate details to the *Ising* model of magnetism as described in [15].

$$S_i \quad \rightarrow \quad sgn\,[h_i(t)] \tag{5}$$

where h_i is the local field that connects all neurons in HNN. The sum of the field is induced by each neuron state as follows,

$$h_i = \sum_k^N \sum_j^N W_{ijk} S_j S_k + \sum_j^N W_{ij} S_j + W_i \tag{6}$$

The task of the local field is to evaluate the final state of neurons and generate all the possible 3-SAT induced logic that was obtained from the final state of neurons. One of the most prominent features of the HNN network is the fact that it always converges. The generalized fitness function $F_{RANkSAT}$ regulates the synaptic combinations in HNN and $F_{RANkSAT}$ is presented as follows.

$$E_{F_{RANkSAT}} = \sum_{i=1}^{NN} \prod_{j=1}^{V} T_{ijk} \tag{7}$$

where V and NN are the number variables and the number of neurons generated in $F_{RANkSAT}$ respectively. We defined the inconsistency of $F_{RANkSAT}$ representation as follows.

$$T_{ij} = \begin{cases} \frac{1}{2}\left(1 - S_\rho\right) , & if \ \neg\rho \\ \frac{1}{2}\left(1 + S_\rho\right) , & otherwise \end{cases} \tag{8}$$

The value of $E_{F_{RANkSAT}}$ is proportional to the value of "inconsistencies" of the logical clauses. The rule for updating the neural state is,

$$S_i(t+1) = \begin{cases} 1 , & h_i \geq 0 \\ -1 , & h_i < 0 \end{cases} \tag{9}$$

Equation (9) represents the Lyapunov energy function in HNN for 3rd order logic.

$$H_{F_{RANkSAT}} = -\frac{1}{3} \sum_{i=1}^{N} \sum_{j\neq k}^{N} \sum_{k=1,i\neq k}^{N} W_{ijk}^{(3)} S_i S_j S_k \\ -\frac{1}{2} \sum_{i=1,i\neq j}^{m} \sum_{j=1}^{m} W_{ij}^{(2)} S_i S_j - \sum_{i=1,i\neq j}^{m} W_i^{(1)} S_j \tag{10}$$

Equation (10) has been applied to classify whether a solution is a global or local minimum energy. The HNN will generate the optimal assignment when the induced neurons state achieved global minimum energy. There are limited works to combine HNN and EA as a single computational network. Thus, the robustness of Election algorithm manage in improving the training process in Hopfield model. Consequently, the quality of the final neuronal state can be maintained according to Equation (11) as utilized in [6],[4],[3],[20] as follows.

$$\left| H_{F_{RANkSAT}} - H_{F_{RANkSAT}}^{\min} \right| \leq \xi \tag{11}$$

where ξ is the pre-determined tolerance value. The value $\xi = 0.001$ was taken in [6],[4],[3],[20],[19],[2][10-11,18-21]. If the $F_{RANkSAT}$ logical representation embedded in HNN does not satisfy the criteria state in Equation (11) then the final state of the neurons have been trapped in the wrong pattern.

2.3. *Election Algorithm as Heuristic Search in Hopfield Learning Phase*

The Election Algorithm (EA) is an iterative population-based algorithm that provides a varieties of solutions in the search space. The local search function have been partitioned into search spaces. The optimization procedure is inspired based on the voting process in human society [11],[10]]. Generally, the population of an individual is divided into two parties which later carry out a series of operations such as initialization, eligibility stages, advertisement and alliance. These stratgiies are resulted in a stronger searching technique. The primary goal of EA is to encourage candidates to coverge on a global minimum solution (best solution) to optimzation problem [10]. Optimization process based on standard Hopfield model has a high probability of becoming caught at a suboptimal solution with the number of neurons firing into the network [3],[9],[7].

Various metaheuristics algorithms, such as Electin algorithm, have been purposefully used in HNN to improve its searching capabilities and to solve the issue of premature convergence before achieving the global optimal, which will maximize the number of fulfilled clauses during the network's training process. Because of its ability to merge local search into a partitioned search area, EA used mechanisms such as a constructive marketing approach, a negative campaign, and an alliance to maximize the entire searching space. The main steps of the procedure in the EA-HNN-RANkSAT model considering $k \leq 3$ is presented from stage 1 to 5 as follows:

Stage 1:Initialization

The main component of any search and optimization is to find the right solution in terms of the problem's variable's parameter. It is created an array of vector parameters to be optimized. A population of this size N_{POP} as a starting point of an individual $\tau_v = [\tau_i, \tau_2, \tau_3, ..., \tau_{N_{POP}}]^T$ for each solution were

generated. Each solution is distributed within the vector boundary range depending on the variable as follows,

$$\tau_v = \lambda_v^{\min} + \tau_1 * \left(\lambda_v^{\max} - \lambda_v^{\min}\right) \tag{12}$$

Where $v \in N$ and τ_v described the location of the *vth* supporter the N_{var} _ _ dimensional search space and N_{POP} described to the number of search representative. A random number with a uniform distribution is defined as $\tau_1 \in [0, 1]$. This problem searches for the optimal RANkSAT clauses.

Stage 2:Eligibility measurement

Everyone's eligibility(fitness) function is measured based on the $F_{RANkSAT}$ clauses in using as follows.

$$f_{\tau_v} = \sum_{i=1}^{t} C_i^{(3)} + \sum_{i=1}^{j} C_1^{(2)} + \sum_{i=1}^{d} C_i^{(1)} \tag{13}$$

where f_{τ_v} denoted as eligibility of each person in search spaces τ_v, $C_i^{(k)}$ is the clause in $F_{RANkSAT}$ and $t, j, d \in [1, N]$ are the total number of logical clauses in $F_{RANkSAT}$ logic.

Stage 3: Creating an initial population

A population of persons (individuals) P_C of N_{pop} search agent was employed in EA. Each solution represents the eligibility (fitness) of a candidate(e_{δ_C}) and the general political parties (P_C) represent search space. Individuals' populations are splitting population of individual N_{pop} into political parties (P_C) is part of the EA policy. Each P_C includes the contender (δ_c) and their followers(τ_v) which served as search agents in the solution space. The δ_c together with their τ_v form some P_C. They τ_v are divided δ_c based on e_{δ_c} in which the initial number τ_v of a δ_c is proportionate to e_{δ_c}. According to [10] the τ_v of a δ_c, are identified by normalizing e_{δ_c} computed as follows.

$$\alpha_i = \left|e_{\delta_{C_i}}\right| = \left|\frac{e_{\delta_C} - \max(I)}{\sum k \in \delta_C e_{\delta_k} - \max(I)}\right| \tag{14}$$

where $I = \left\{\delta_{c_j} | j \in \delta_c\right\}$, $e_{\delta_{C_i}}$, α_i defined the eligibility of candidate and normalized eligibilities of the candidate δ_{C_i} respectively and δ_C designated the initial number of candidates in the solution space. The number of an individual to serve as initial candidates δ_{C_i} was modelled as follows.

$$\delta_{C_i} = \alpha_r N_{pop} \tag{15}$$

The original number of $\delta_{\tau_{vi}}$ is modelled as follows.

$$\delta_{\tau_{vi}} = N_{pop} - \delta_{C_i} \tag{16}$$

Then, we randomly select $\delta_{\tau_{vi}}$ of the τ_v and add to δ_{C_i} which form an P_C in the search space.

Stage 4: *Campaign strategy:* ' This stage is modelled from step 1 to step 3 as follows

Step 1: *Positive advertisement* (ϑ)

In the modelling of EA ϑ. We select τ_v a position that is in form of a variable of δ_C in the search space. The intention of sampling the random numbers is to choose a τ_{v_1} location to be replaced by a new voter τ_{v_2}. The method for determining the

selection rate $\lambda_\tau \in [0, 1]$. The number of variables transferred to the τ_v by the δ_C is defined according to the following.

$$\psi_\tau = \lambda_\tau S_C \tag{17}$$

The random choice variables in the problem space that must be replaced are signified by ψ_τ. The selection rate is defined by λ_τ. S_C signifies the total number of δ_C in the problem space. To transfer τ_v to another δ_C, the eligibility distance coefficient (e_d) was used model as follows,

$$e_d = \frac{1}{\left\|e_{\delta_{C_i}} - e_{\delta_{\tau_{vi}}}\right\| + 1} \tag{18}$$

where $e_{\delta_{C_i}}$ is utilized to described the eligibility of δ_{C_i} and $e_{\delta_{\tau_{vi}}}$ has been used to represent the eligibility of $\delta_{\tau_{vi}}$ voter in the search space.

Step 2: *Negative advertisement* (ω)

EA algorithm uses negative campaign operation (ω)as a search mechanism in their opposition movement, to fascinate members of other parties This leads to the marginalized parties' revival and deterioration of progress in the following ways.

$$\omega = \tau_v^T = \begin{cases} \tau_{vi}, & R_j \leq \varphi \\ \tau_{vj}, & R_j > \varphi, \end{cases} \tag{19}$$

where $R_j \in [0, 1]$ and φ is the negative advertisement constant in EA.

Step 3: *Coalition strategy* (C_L)

In Election algorithm, two or more parties come together for a new party, sharing the same ideas and aims in space to find solutions; confederates if they have the same ideas; EA, two or more parties may sometimes come together for a new party, possessing the same ideas and aims in space to find solutions. As a result, some applicants are exiting the advertisement with a new nominee dubbed "leader," while the candidate who withdrew from the election arena is dubbed "follower." To rule the optimum solution search in the search space, EA employs a coalition operation. By building trial vectors using elements of existing party candidates in the solution space and improving the search space, the coalition strategy can effectively gather information about effective party mergers.

$$\widehat{\tau}_{vi} = \tau_{v1} + \lambda_\tau(\tau_{v3} - \tau_{v2}) \tag{20}$$

where $\widehat{\tau}_{vi}$ defined as a coalition parameter of political parties and $\lambda_\tau \in [0, 1]$ served as a scaling factor and i is an index of current solution During the coalition process(C_L). In EA, a population of solution vectors is randomly created at the start. In Election algorithm, each fresh solution achieved will compete with a united party in the search space.

Stage 5:Stopping condition (Election Day)

At the start of EA, a population of the optimization algorithm is generated at random. In the Election algorithm, each new solution will participate in the search space with a unified group [11].
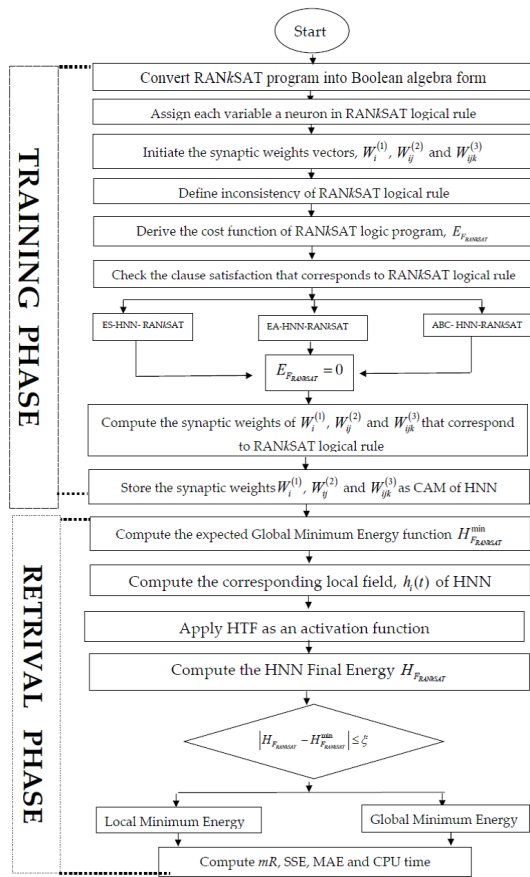
Figure 1. Implementation of RANkSAT

$$MAE = \sum_{i=1}^{d} \frac{1}{n} |f_{NN} - h_d| \tag{23}$$

where f_{NN} and h_d characterized the HNN the output and the target output values respectively, d categorized the number of permutations in HNN.

4. Results and Discussion

Table 1. Global minimum ratio for RANkSAT logic

NN	EA	ES	ABC
10	1	1	1
20	1	1	1
30	1	1	1
40	1	1	0.9999
50	1	1	1
60	1	1	0.9997
70	1	NIL	1
80	1	NIL	1
90	1	NIL	0.9996
100	1	NIL	1
110	1	NIL	0.9899

Table 1 and Figure 2 until Figure 4 displayed the performance of RANkSAT in HNN considering the third order of $k \leq 3$ in terms of mR, errors accumulations and time-consuming during the program execution. In our experiments, we explore the performance of the proposed training approach using a different number of neurons $10 \leq NN \leq 110$. The general trend of the model performance indicates a massive increase in errors accumulation and CPU time considering the complexity of the neuron fired to HNN in searching for optimal RANkSAT representation. The increasing trend in error behaviour shows the complexity of the neuron states of RANkSAT which proved to be an NP problem [7],[9]. According to SSE and MAE in Figure4 and Fig 5 respectively in measuring the HNN performance during the learning phase in searching for correct optimal assignment to RANkSAT logical clauses, the proposed method, EA-HNN-RANkSAT, was able to achieve $E_{F_{RANkSAT}} = 0$, with lower statistical errors accumulation than ABC-HNN-RANkSAT. This may be due to the multiple optimization layers involve in EA which has a better screening stage in searching for optimal assignment leading to $E_{F_{RANkSAT}} \to 0$ in fewer iterations than ABC-HNN-RANkSAT.This explores the optimal capacity of EA in lowering the complexity of the HNN searching of RANkSAT logical towards error accumulation by reducing the number of iterations in the optimization process.

The optimal behaviour of the Hopfield neural network models based on mR has been recorded in Table 1. The efficiency of the Election algorithm (EA) are observed in comparison with other metaheuristics search approaches in hopfield neural network for RANkSAT representation leading to $E_{F_{RANkSAT}} \to 0$. Table 1, EA-HNN-RANkSAT and ES-HANN-RANkSAT can retrieve more accurate neural assignment that leads to the best global solution during the training process,
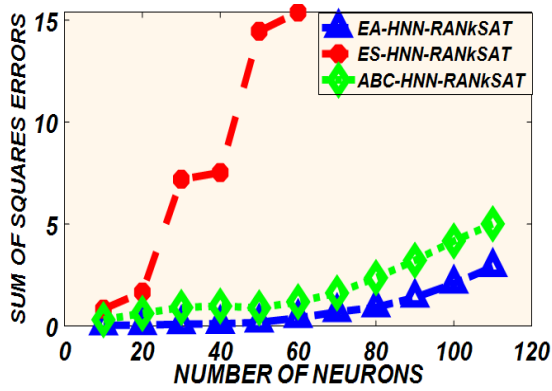
2.4. Simulation Procedure

Implementation of the Neuro-Heuristic searching method of RANkSAT in HNN. The program's main task is to find the best "model" that find the optimal occurrences of random-kSAT. Both logical variables and clauses were initially randomized. Simulations were executed by manipulating a different number of neurons complexity ranging from $10 \leq NN \leq 110$. The simulation has been conducted on RANkSAT as a logical clause in HNN according to the flowchart in Figure 1.

3. Performance

The efficiency of performance of the EA-HNN-RANkSAT model has been quantified based on Global minimum ratio (mR), statistical error measurement based on the sum of square error (SSE) and mean absolute error (MAE) as well as model computational time (CT) presented in Equation (20) to Equation (23) respectively.

$$mR = \frac{1}{ab} \sum_{i}^{t} H_{F_{RANkSAT}} \tag{21}$$

$$SSE = \sum_{i=1}^{d} (f_{NN} - h_d)^2 \tag{22}$$
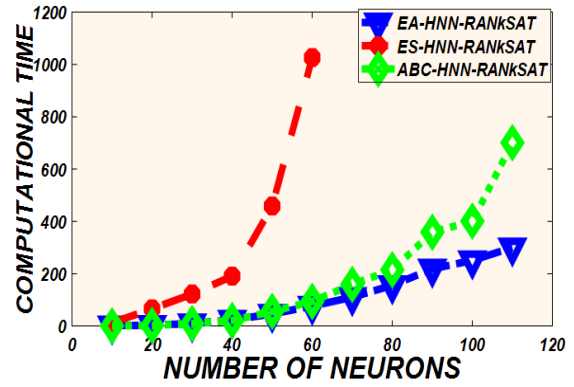
Figure 2. SSE for RANkSAT in HNN.
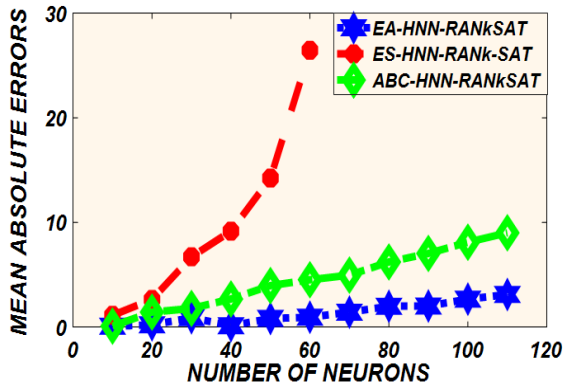

Figure 4. CPU Time for HNN performance.

Figure 3. MAE for RANkSAT in HNN

while in ABC-HNN-RANkSAT model, some neural states were stuck at $NN = 60, NN = 60, NN = 90, NN = 110$ a suboptimal local solution, but still managed to achieved closed to 92% success. Meanwhile, ES-HNN-RANkSAT can only accommodate $NN \leq 60$, as the model exceeds the running time threshold, the neuron complexity increases, which is particularly problematic in the case of inconsistent interpretation $\neg F_{RANkSAT}$. The core objective of incorporating metaheuristics in artificial neural network is to increase the flow of learning process by reducing the sensitivity of the neuros complexity, allowing the neurons to progress into relaxation and recovery stages sucessfully.

The EA-HNN-RANkSAT model was able to recovered a much more appropriate final configuration that leads to a global minimum solution. This confirms the robustness and higher efficiency in neuro-searching embedded by EA to strengthen the HNN learning process for RANkSAT logical representation. If the mR of the model HNN network approaches one after the computing cycle, all solutions generated in the network have achieved global minimum energy[19].

It can be observed from Figure 3 and Figure 4 that the learning errors measured in terms of SSE and MAE increase massively as the neurons passed $NN \leq 20$. Figure 2, presents the trend of performance based on SSE measure. The high accumulation of SSE was demonstrated by ES-HNN-RANkSAT. EA-HNN-RANkSAT accumulated lower error with close to 98%

accuracy. EA-HNN-RANkSAT performs well as the number of neurons reaches a certain threshold. $NN \geq 20$ the rapid increase in error was noticed. A similar pattern is reported in Figure 3, which evaluate the models' performance based on MAE error. It revealed that EA-HNN-RANkSAT accumulated lower MAE than ES-HNN-RANkSAT with higher MAE accumulation. However, the performance of ABC-HNN-RANkSAT and EA-HNN-RANkSAT displayed similar trend. In general, ES-HNN-RANkSAT has the lowest results, with over 45 per cent of searches failing.

The SSE and MAE values for HNN-RANkSAT searching behaviour have been observed to increase rapidly as the network's neurons become more complex. Compared to EA-HNN-RANkSAT and ES-HNN-RANkSAT, the proposed EA-HNN-RANkSAT was able to obtain $E_{F_{RANkSAT}} \to 0$ with a lower error accumulation. This is due to the EA scanning process involving multiple optimization surfaces that have a better filtering point in the state space, enabling the selection procedure to get excellent performance leading to $E_{F_{RANkSAT}} \to 0$ in fewer iterations. In comparison to ES-HNN-RANkSAT and ABC-HNN-RANkSAT, EA-HNN-RANkSAT registered lower SSE and MAE, as per error analysis in Figures 2 and 3. This study looked into the robustness of EA in decreasing the resistance of the HNN to error occurrence by lowering the number of iterations to the bare minimum. Consequently, in ES-HNN-RANkSAT, the study of mR, SSE, and MAE abruptly end at $NN = 60$. This may be attributed to the ineptness of the learning method used in ES-HNN-RANkSAT, which can't handle ambiguity. As a result of several fluctuations by neurons, the solutions were crafted at a sub-optimal solution (wrong pattern). It is clear that EA-HNN-RANkSAT agreed with ABC-HNN-RANkSAT but outperformed ES-HNN-RANkSAT in searching optimal representation RANkSAT logical representation.

Figure 4 displayed the HNN-RANkSAT models according to their running time during the implementation cycle. The proposed EA-HNN-RANkSAT was able to execute $NN = 90$ within 513.23 seconds faster than ABC-HNN-RANkSAT which produce $NN = 90$ in about 763.02 seconds. The conventional ES-HANN-RANkSAT model was able to withstand $NN \leq 60$ in 1026.53 seconds. Examining the CPU's use patterns in Figure 4, it could be observed that, the RANkSAT logical clauses are

becoming complicated and complex, the searching for global solutions to the RANkSAT logical clause required more effort which subsequently required more execution time to achieve. The search capacity in ES-HNN-RANkSAT demands more time in to search $30 \leq NN \leq 60$neurons. The EA-HNN-RANkSAT and ABC-HNN-RANkSAT are similar in their running time to $10 \leq NN \leq 80$neurons. However, EA-HNN-RANkSAT was slightly faster than ABC-HNN-RANkSAT at the initial and final searching process. This is because more neurons are reqyured during the training phase to allow network to migrate through the energy level and best on optimal solutions. In other words, as the number of neurons grew, the number of errors accumulated less in EA-HNN-RANkSAT, which subsequently reduce the CPU time comsumption. In this case, ES-HNN-RANkSAT required further iterations to find the best solution that corresponds to $E_{F_{RANkSAT}} = 0$, which took more CPU time.

In the context of the RANkSAT logical representation $k \leq 3$, the robustness of integrating EA to promote the training phase of HNN can be seen. Even during the learning process, EA's stochastic scanning behaviour expands the structure HNN for appropriate RANkSAT representation. Consequently, the EA-HNN-RANkSAT model's composition would reflect the diversification of the final neural states. As a result, the EA-HNN-RANkSAT, where the chance of obtaining diversified $F_{RANkSAT}$ solutions is much higher, the solutions are dynamically swapped. As a result, EA-HNN-RANkSAT will produce a greater variety of $F_{RANkSAT}$ logical clauses that are achievable leading to $E_{F_{RANkSAT}} = 0$. The essence of HNN-RANkSAT, on the other hand, will present difficulties in the event of conflicting assignment. As the number of neurons grew throughout the trial, the integration of EA in HNN dealt systematically with the higher learning complexity. This paper investigates the robustness and efficacy of EA's local quest and global approach. The robustness of the local and global search capability involved in EA, which serves as the learning mechanism in HNN, is linked to the success of EA-HNN-RANkSAT in exploring the global solution. The robustness of the local and global search capability contained in EA, which serves as the learning mechanism in HNN, is linked to the success of EA-HNN-RANkSAT in hunting for the global solution $F_{RANkSAT}$. At the early stages of EA, where the number of neurons is limited, the local search potential has a major impact. This paper investigates how to improve the control parameters in EA to improve the learning process and achieve optimum $E_{F_{RANkSAT}} = 0$ logical representation. At the outset, a candidate selection optimization operator is needed to speed up the process of choosing the most qualified candidate to act as a leader (solution). Multiple optimization layers employed by EA-HNN-RANkSAT to diversify the approach space and increase the searching capability in a specific area [11]. The positive advertisement is the first optimization layer in EA, which optimizes among candidates in a specific political party in the search space. The negative advertisement is another layer that allows other candidates from another party to take the supporter from their party. The coalition's policy has the potential to have a huge effect in attracting the largest number of people who support the party's manifesto in seeking global solutions. Within a fair period, this process would provide a collabora-tive candidate of equal fitness [12],[10]. Because of these features in the EA, the hybrid proposed model can reduce the number of iterations an HNN is needed during the learning process by ensuring that there is a minimal amount of error accrued after the experimentation. The EA-HNN-RANkSAT model's systemic solution search space can make the local and global search processes easier to achieve global solutions. The hybrid approach will effectively search for the optimal solution in all of the described spaces thanks to the partitioning mechanism of the search space. Finally, the EA-HNN-RANkSAT model has a shorter computing time because of its campaign and alliance mechanisms, which systematically improve the unified party's chances of victory in a decent period.

5. Conclusion

A hybrid approach was proposed in this work, in which the Election algorithm (EA) was combined with a Hopfield neural network (HNN) to perform RANkSAT representation as a new logical law. The proposed hybrid EA-HNN-RANkSAT model can be conclusively shown to be a robust heuristic methodology that is effective in improving or following preferred assignments, even in clauses of high difficulty, based on the findings presented based on experimental simulations performed. This is related to the EA process's stronger optimization layers, which speed up HNN's learning process in looking for an optimal RANkSAT assignment of greater eligibility. It was revealed that the EA-HNN-RANkSAT model was able to complete the searching process slightly faster than ABC-HNN-RANkSAT and ES-HNN-RANkSAT. Notwithstanding, all of the HNN models under consideration produced excellent results when it came to representing RANkSAT logic in HNN and computing the global solution to $E_{F_{RANkSAT}} = 0$ within the confines of a reasonable CPU timeline. As a result, in terms of mR, MAE, SSE, and CPU power, EA-HNN-RANkSAT faced less numerical strain during the training phase than other models. Finally, other metaheuristics approach such as firefly algorithm, dronefly algorithm etc will be hybridized with the Hopfield neural network (HNN) to enhance its computational process. We will also explore other NP problem such as a reliability problems in the Hopfield neural network. The study will be expanded to include reverse analysis (RA) for actual data sets using data mining tools.

References

[1] W. A. T. W. Abdullah, "Logic programming on a neural network", International journal of intelligent systems **7** (1992) 513.
[2] H. Abubakar, S. A. Mmasanwa, S. Yusuf & Y. Abdurrahman, "Agent Based Computational Modelling For Mapping Of Exact Ksatisfiability Representation In Hopfield Neural Network Model", International Journal of Scientific and Technology Research **9** (2020) 76.

[3] H. Abubakar, S. R. M. Sabri, S. A. Masanawa & S. Yusuf, "Modified election algorithm in hopfield neural network for optimal random k satisfiability representation", International Journal for Simulation and Multidisciplinary Design Optimization **11** (2020) 16.

[4] H. Abubakar & S. Sathasivam, "Developing random satisfiability logic programming in Hopfield neural network", AIP Conference Proceedings, AIP Publishing LLC **2266** (2020).

[5] H. Abubakar, S. Sathasivam & S. A. Alzaeemi, "Effect of negative campaign strategy of election algorithm in solving optimization problem", Journal of Quality Measurement and Analysis JQMA **16** (2020) 171.

[6] H. Abubakar, S. Yusuf & S. A. Masanwa, "Exploring the Feasibility of Integrating Random k-Satisfiability in Hopfield Neural Network", International Journal of Modern Mathematical Sciences **18** (2020) 92.

[7] D. Achli optas, *Random satisfiability*, IOS press **185** (2009) 245.

[8] B. U. Ayhan & O. B. Tokdemir, "Accident analysis for construction safety using latent class clustering and artificial neural networks", Journal of Construction Engineering and Management **146** (2020) 1.

[9] A. Biere, M. Heule & H. van Maaren, *Handbook of satisfiability*, IOS press **185** (2009).

[10] H. Emami, "Chaotic election algorithm", Computing and Informatics **38** (2019) 1444.

[11] H. Emami & F. Derakhshan, "Election algorithm: A new socio-politically inspired strategy", AI Communications **28** (2015) 591.

[12] S. Emami, Y. Choopan & J. Parsa, "Modeling the Groundwater Level of the Miandoab Plain Using Artificial Neural Network Method and Election and Genetic Algorithms", Iranian journal of Ecohydrology **5** (2018) 1175.

[13] V. Feldman, W. Perkins & S. Vempala, "On the complexity of random satisfiability problems with planted solutions", SIAM Journal on Computing **47** (2018) 1294.

[14] J. Heo, J. G. Yoon, H. Park, Y. D. Kim, H. S. Nam & J. H. Heo, "Machine learning–based model for prediction of outcomes in acute stroke", Stroke **50** (2019) 1263.

[15] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", Proceedings of the national academy of sciences **79** (1982) 2554.

[16] Z. Meng, X. Guo, Z. Pan, D. Sun & S. Liu, "Data segmentation and augmentation methods based on raw data using deep neural networks approach for rotating machinery fault diagnosis", IEEE Access **7** (2019) 79510.

[17] C. R. Rahman, P. S. Arko, M. E. Ali, M. A. I. Khan, S. H. Apon, F. Nowrin & A. Wasif, "Identification and recognition of rice diseases and pests using convolutional neural networks" Biosystems Engineering **194** (2020) 112.

[18] F. Sarafa, A. Souri & M. Serrizadeh, "Improved intrusion detection method for communication networks using association rule mining and artificial neural networks", IET Communications **14** (2020) 1192.

[19] S. Sathasivam, "Upgrading logic programming in Hopfield network", Sains Malaysiana **39** (2010) 115.

[20] S. Sathasivam, M. Mohd, M. S. M. Kasihmuddin & H. Abubakar, "Election algorithm for random k satisfiability in the Hopfield neural network", Processes **8** (2020) 568.

[21] A. Wanto, A. P. Windarto, D. Hartama & I. Parlina, "Use of binary sigmoid function and linear identity in artificial neural networks for forecasting population density",IJISTECH (International Journal of Information System & Technology) **1** (2017) 43.