

SOIL FERTILITY CLASSIFICATION VIA HYBRID HOPFIELD NEURAL NETWORK WITH ELECTION ALGORITHM FOR RANDOM BOOLEAN SATISFIABILITY REVERSE ANALYSIS

Hamza Abubakar^{1,2,*} and Mohammad Shafiq¹

Received: May 01, 2021; Revised: June 24, 2021; Accepted: September 19, 2021

Abstract

Artificial neural network (ANN) possesses a comprehensive structure of training and testing stages that made it one of the most efficient tools in patterns and knowledge extraction in solving real-life applications such as classification, forecasting, risk analysis, detection and quantitative analysis. This paper presents a novel heuristic search and optimization-based algorithm inspired by political parties' competition in the pursuit of power known as the Election Algorithm (EA). The purpose is to hybridize the Hopfield neural network (HNN) with a novel Election algorithm (EA) as a single neuro-dynamical model for high order random ksatisfiability reverse analyses in data mining application. The objectives are to reflect knowledge efficiently for the Soil fertility classification problem. The learning process of the Hopfield neural network (HNN) has undergone various modifications and improvements according to various optimization problems. However, the HNN model is associated with the drawback of being easily trapped in the local minimum solution. The Election algorithm (EA) is proposed to address these challenges to accelerate the learning phase of HNN for optimal classification problem. To prove the efficacy of the proposed hybridization, Soil fertility datasets (SFDS) has been employed in this study. The paper considers investigating the performance of algorithms on Hopfield neural network- based Random Satisfiability Reverse Analysis (HNN-RANSATRA). The performance analysis of the proposed technique has been compared with the other two techniques in the Hopfield neural network- based Random Satisfiability Reverse Analysis with Artificially bee colony (HNN-RANSATRA-ABC) and Hopfield neural network- based Random Satisfiability Reverse Analysis with Exhaustive search techniques (HNN-RANSATRA-ES). The performance of all methods under study have been evaluated in terms of error accumulations and accuracy in the classification of Soil fertility data set (SFDS). The result of the experiment revealed that the proposed technique outperformed its two counterparts in terms of robustness, accuracy, and efficiency during the training process. Hence, based on the simulation results it has been proven that the Election algorithm complied effectively with the Hopfield neural network for Random Satisfiability Reverse Analysis for soil fertility data set (SFDS).

Keywords: Hopfield neural network, Election algorithm, Reverse analysis, Random kSatisfiability, Soil fertility data set

¹ School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Palau Penang, Malaysia. E-mail: zeeham4u2c@yahoo.com

² Department of Mathematics, Isa Kaita College of Education, P.M.B 5007, Dutsin-Ma, Katsina, Nigeria.

* Corresponding Author

Introduction

Artificial Neural Networks (ANN) is an abstract mathematical model based on neural biological networks. It is an attempt to model the nervous system in a serial system and has become well-established mathematical frameworks utilizing simple elementary units associated with suitable weighted connections (Sasaki *et al.*, 1992). Artificial Neural Networks (ANN) are massively parallel computational structures that consist of an extremely large number of simple processors with multiple interconnections. One of the main characteristics of ANN is their ability to learn complex and dynamic nonlinear input-output associations, using sequential learning methods, and respond to the environment. The sources of mathematical inspiration in the Neural network modelling has been the ongoing research in an attempt to formalize human thinking capacity and the nervous system (Rojas, 1996).

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 (Emami and Derakhshan, 2015). In this work, a novel and powerful heuristics search technique is known as EA has been explored for effective training of complex model (Yang *et al.*, 2014). EA has been utilized in solving various mathematics and engineering benchmark problems which yield successful performance in both the convergence rate and better identification of global optima. We can infer that the application of employing EA is endless; from industrial planning, scheduling, decision making and machine learning. Election algorithms have been utilized in (Emami *et al.*, 2019) successfully for predicting groundwater level (Zubaidi *et al.*, 2020). Recently, Election algorithm has been proposed (Sathasivam *et al.*, 2020; Abubakar *et al.*, 2020) to accelerate the learning phase of HNN for random 2 satisfiability. The performance of EA in solving the Travelling Salesmen Problem has been investigated in (Abubakar *et al.*, 2020). Although EA has been applied in various optimization areas, it recorded tremendous achievement in searching for the optimal solution to combinatorial types of optimizations. In our work, EA will be applied on the HNN network for searching of optimal representation (global minimum) solutions to Random three Satisfiability Reverse Analysis for SFDS.

However, we did not come across any work that combined the global and local searching capacities of EA in enhancing the training phase of HNN in searching for optimal representation to random Satisfiability Reverse Analysis (RANKSATRA) for SFDS. Therefore, a new computational method been proposed by hybridization of Hopfield neural network with Election algorithm as a single model. The aim is to improve the learning process of HNN in searching for optimal RANKSATRA in achieving a global optimum with robustness and better accuracy for SFDS classification. The contributions of our work include; (1) To upgrade the "RANKSAT" logical rule to accommodate high order logic; (2) To the implementation of the upgraded logical rule RANKSAT in Hopfield neural network as a single model; (3) To the implementation of Election algorithm (EA) in discrete Hopfield artificial neural network (HNN); (4) to implement the proposed logic rule for Reverse Analysis (HNN-RANSATRA-EA); (5) To perform Soil fertility classification based on the Reverse Analysis in (4) and (6) To compare of the Election algorithm with other state-of-the-art searching methods in HNN for Random kSatisfiability representation.

The effectiveness of the EA in HNN for RANKSAT will be explored by comparing its performances to the state-of-the-art model. By developing an effective ANN working model, the new hybrid integrated model will be beneficial to various combinatorial optimization and computational mathematical communities by proving an alternative or optional approach to doing computation. The structure of this work is as follows. In Section 2, the proposed methods involve the mapping of RANKSAT as a new logical rule in HNN and a heuristic search method in the HNN learning phase via EA. We presented the implementation procedure and the experimental setup for RANKSAT in HNN have been presented in section 3. Section 4 reported the models' performance evaluation metrics. The experimental results of their discussions were reported in section 5. The last section concludes the paper.

Materials and Methods

Proposed 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 (Feldman *et al.*, 2018). The general description for the formulation $F_{RANKSAT}$ is presented in (1) as follows.

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

Where $t, j, d \in \mathbb{N}$, $t > 0$, $j > 0$ and $d > 0$. The clause $F_{RANKSAT}$ is defined as a RANKSAT ($k \leq 3$) for $C_i^{(k)}$ clauses described in (2) as follows.

$$C_i^{(k)} = \begin{cases} (\alpha_i \vee \lambda_i \vee \psi_i), & \forall k = 3 \\ (\alpha_i \vee \lambda_i), & \forall k = 2 \\ \omega_i, & \forall k = 1 \end{cases} \quad (2)$$

where the variables $\alpha_i \in [\alpha_i, \neg\alpha_i]$, $\lambda_i \in [\lambda_i, \neg\lambda_i]$, $\omega_i \in [\omega_i, \neg\omega_i]$, $\psi_i \in [\psi_i, \neg\psi_i]$ are represent the 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 $2^{\binom{d+j+t}{y}}$ without replacement a logic clause 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.

$$F_{RANKSAT} = (\omega_1 \vee \neg\omega_2 \vee \omega_3) \wedge (\lambda_1 \vee \lambda_2 \vee \neg\lambda_2) \wedge \neg\alpha_1 \quad (3)$$

According to Equation (3), $F_{RANKSAT}$ comprises of $C_i^{(3)} = (\omega_1 \vee \neg\omega_2 \vee \omega_3)$, $C_2^{(2)} = (\neg\lambda_1 \vee \lambda_2)$ and $C_1^{(1)} = \neg\alpha_1$. Therefore, the outcome of in (3) is satisfied $F_{RANKSAT} = -1$ if

$$(\omega_1, \omega_2, \omega_3, \lambda_1, \lambda_2, \lambda_3, \alpha_1) = (1, 1, 1, -1, 1, 1) \quad (4)$$

3 clauses are satisfied $(C_i^{(3)}, C_1^{(2)}, C_2^{(1)})$.

This research $F_{RANKSAT}$ will be embedded in HNN via EA 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.

Representing 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 (Hopfield, 1982). Its structure consists of interconnected neurons and a powerful feature of content addressable memory that is crucial in solving various optimization and combinatorial tasks (Abubakar *et al.*, 2020; Abubakar *et al.*, 2020; Sathasivam *et al.*, 2020; Abubakar and Danrimi, 2021; Abubakar *et al.*, 2021). The network is composed of organised neurons of which is portrayed by a variable called *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 HNN is shown in Equation (5) as follows.

$$S_j = \begin{cases} 1 & , \text{ if } \sum_j W_{ij} S_j > \varsigma \\ -1 & , \text{ Otherwise} \end{cases} \quad (5)$$

where w_{ij} is the synaptic weight vector from starting from j neuron to i neuron. We defined S_i it as the state of the neuron i in HNN and ς is the predefined value. The value of $\varsigma = 0$ has been specified in (Saratha, 2010; Abubakar *et al.*, 2020; Abubakar *et al.*, 2020; Sathasivam *et al.*, 2020; Abubakar and Danrimi, 2021; Abubakar *et al.*, 2022) 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 i as follows.

$$W_{iii}^{(3)} = W_{jjj}^{(3)} = W_{kkk}^{(3)} = 0 \quad (\forall k \leq 3) \quad (6)$$

$$W_{ii}^{(2)} = W_{jj}^{(2)} = W_{kk}^{(2)} = 0 \quad (\forall k \leq 2) \quad (7)$$

In resulting, HNN holds symmetrical features in terms of architecture. HNN model has similar intricate details to the *Ising* model of magnetism as described in (Neelakanta and De Groff, 2018).

$$S_i \rightarrow \text{sgn}[h_i(t)] \quad (8)$$

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,

$$h_i = \sum_k^N \sum_j^N w_{ijk} S_j S_k + \sum_j^N w_{ij} S_j + w_i \quad (9)$$

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.

$$E_{F_{\text{RANKSAT}}} = \sum_{i=1}^{NN} \prod_{j=1}^V T_{ijk} \quad (10)$$

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.

$$T_{ij} = \begin{cases} \frac{1}{2}(1 - S_\rho), & \text{if } \neg \rho \\ \frac{1}{2}(1 + S_\rho), & \text{otherwise} \end{cases} \quad (11)$$

The value of $E_{F_{\text{RANKSAT}}}$ is proportional to the value of “inconsistencies” of the logical clauses. The rule for updating the neural state is,

$$S_i(t+1) = \begin{cases} 1, & \forall h_i \geq 0 \\ -1, & \forall h_i < 0 \end{cases} \quad (12)$$

Equation (13) represents the 3rd order Lyapunov energy function (LEF) of HNN model and subsequently to classify whether a solution is a global or local minimum energy

$$H_{F_{\text{RANKSAT}}} = -\frac{1}{3} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N w_{ijk} S_i S_j S_k - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} S_i S_j - \sum_{i=1}^N w_i S_i \quad (13)$$

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 EA is observed in improving the training process in HNN model.

Consequently, the quality of the final neuronal state can be maintained according to Equation (14) as follows.

$$|H_{F_{\text{RANKSAT}}} - H_{F_{\text{RANKSAT}}}^{\min}| \leq \xi \quad (14)$$

where ξ is the pre-determined tolerance value. The value was taken in (Saratha, 2010; Abubakar *et al.*, 2020; Abubakar *et al.*, 2020; Sathasivam *et al.*, 2020; Abubakar and Danrimi, 2021; Abubakar *et al.*, 2022). If the logical representation embedded in HNN does not satisfy the criteria state in Equation (14), then the final state of the neurons have been trapped in the local solution (wrong pattern).

Random kSatisfiability Based Reverse Analysis Method (RANkSATRA)

Data mining based on Logic rule will be executed efficiently in achieving an optimal representation to HNN-RANkSATRA which is used in the knowledge extraction. The neurons (attributes) is usually being represented in a bipolar form representing 1 for true and -1 for falsification. The optimal representation of data used can be revealed by obtaining the correct synaptic weight vector between 2 neurons in HNN-RANkSATRA. Therefore, Wan Abdullah's method (1992) which is equivalent to Hebbian learning methods has been applied in the computing the synaptic weight vector of the HNN-RANkSATRA between the two neurons. By considering three attributes and where, and are the possible RANkSATRA logical clauses with its corresponding synaptic weights obtained and store as HNN CAM. Equation (12) is the cost function obeying the bipolar follows:

$$E_{F_{\text{RAND3SAT}}} = \frac{1}{2}(1 - S_p) \frac{1}{2}(1 - S_r) \frac{1}{2}(1 + S_q) + \frac{1}{2}(1 - S_e) \frac{1}{2}(1 + S_f) \frac{1}{2}(1 - S_g) + \frac{1}{2}(1 + S_h) \frac{1}{2}(1 - S_k) \quad (15)$$

The consistent interpretation leading to $E_{F_{\text{RAND3SAT}}} = 0$ is the minimum value corresponding to random clauses. The value of $E_{F_{\text{RAND3SAT}}}$ that is proportional to the number of unsatisfied clauses. Equating the cost function in Equations (15) to LEF in Equation (13), the corresponding synaptic weight of HNN-RANkSAT can be obtained as follows

Election Algorithm as Heuristic Search technique in Hopfield Learning Phase The Election Algorithm (EA) is an iterative population-based algorithm that works for a variety of solutions. The local search mechanism of EA have been partitioned into a search room. The optimization procedure is inspired by the voting process of human society (Emami and Derakhshan, 2015). The primary goal of EA is to encourage candidates to agree on a global minimum solution (best solution), which is thought to have shown a robust method in solving optimization tasks. EA is designed to be distinct and beneficial in comparison to other metaheuristics because it facilitates conducting neighbourhood motions in both continuous and discrete search spaces (Emami, 2019). Optimization process based on standard HNN model has a high chance of being trapped at a suboptimal solution more especially when the number of neurons(NN) fired into the network is high. Metaheuristics algorithms, such as EA, 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

their τ_v form some P_c . The τ_v are divided among 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 Emami (2019) the τ_v of a δ_c , are identified by normalizing e_{δ_c} computed as follows.

$$\alpha_i = |e_{\delta_{c_i}}| = \left| \frac{e_{\delta_{c_i}} - \max(I)}{\sum_{k \in \delta_c} e_{\delta_k} - \max(I)} \right| \quad (20)$$

where $I = \{\delta_{c_j} \mid j \in \delta_c\}$. $e_{\delta_{c_i}}$, α_i defined the eligibility of the candidate and the normalized eligibility 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.

$$\delta_{c_i} = \alpha_r N_{pop} \quad (21)$$

The original number of $\delta_{\tau_{v,j}}$ is modelled as follows.

$$\delta_{\tau_{v,j}} = N_{pop} - \delta_{c_i} \quad (22)$$

The $\delta_{\tau_{v,j}}$ of the τ_v have been randomly select and add to δ_{c_i} which form a P_c in the search space.

Stage 4: Campaign advertisement: this stage is modelled in step 1 to step 3 as follows

Step 1: Positive advertisement ($\mathcal{G}$)

In the modelling of EA $\mathcal{G}$. 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 $\lambda_r \in [0,1]$. The number of variables transferred to the τ_v by the δ_c is defined according to the following.

$$\psi_r = \lambda_r S_c \quad (23)$$

The random choice variables in the problem space that must be replaced are signified by ψ_r . The selection rate is defined by λ_r . 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,

$$e_d = \frac{1}{\|e_{\delta_{c_i}} - e_{\delta_{\tau_{v,j}}}\| + 1} \quad (24)$$

where $e_{\delta_{c_i}}$ is utilized to described the eligibility of δ_{c_i} and $e_{\delta_{\tau_{v,j}}}$ has been used to represent the eligibility of $\delta_{\tau_{v,j}}$ 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.

$$\omega = \tau_v^T = \begin{cases} \tau_{v,j}, R_j \leq \varphi \\ \tau_{v,j}, R_j > \varphi, \end{cases} \quad (25)$$

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

Step 3: Coalition strategy (C_L)

To model the optimum solution search in the search space, EA employs a coalition operation. This is done by building trial vectors using elements of existing party candidates in the solution space and improving the solution search space, the coalition strategy can effectively gather information about effective party mergers.

$$\hat{\tau}_{v,i} = \tau_{v,i} + \lambda_r (\tau_{v,i} - \tau_{v,i}) \quad (26)$$

where $\hat{\tau}_{v,i}$ defined as a coalition parameter of political parties and $\lambda_r \in [0,1]$ served as a scaling factor and i is an index of current solution during the coalition process (C_L).

Stage 5: Stopping condition (Election Day)

At the start of EA, a population of optimization algorithm is generated at random. In the EA algorithm, each new solution will participate in the search space with a unified group (Emami and Derakhshan, 2015).

Description of Real datasets

The proposed hybrid model HNN-RANKSATRA-EA is evaluated using the Soil fertility dataset (SFDS). The dataset ASFDS were retrieved from the UCI Machine Learning Repository platform. It contains information about each data set that contains different purposes. ASFDS dataset has 9 attributes and a total of 1988 instances of soil samples. In this experiment, we aim to analyse a comprehensive comparison of accuracy between HNN-RANSATRA-EA with HNN-RANSATRA-ABC and HNN-RANSATRA-ES using the same data set. The effectiveness of implementing EA in the training stage of HNN-RANKSATRA model is investigated in this paper. The investigation of a model's performance is separated into two parts. The first significant part is to examine the quality of the solution generated by different searching techniques by employing suitable training errors. Secondly is to analyse the robustness and efficiency of the proposed model based on implementation time and accuracy during the training and retrieval process. Therefore, this research main contribution is to explore the competency of HNN-RANKSATRA-EA it compares with the existing models for the classification of Soil fertility data sets (SFDS). The details of ASFD are shown in Table 1.

Table 1. List of Attributes for data sets

Data Set	Details attribute	Output $F_{RANKSAT}$
SFDS Bhuyar (2014)	ω_1 pH value of soil	Classify between fertile and non fertile soil
	ω_2 Organic Carbon,%	
	ω_3 Phosphorous, ppm	
	λ_1 Potassium, ppm	
	λ_2 Iron, ppm	
	λ_3 Zinc, ppm	
	α_1 Manganese, ppm	
	α_2 Copper, ppm	

RANKSATRA Performance Evaluation

The efficiency of performance of the EA-HNN-RANKSAT model has been quantified based statistical error measurement based on the root mean square error (RMSE), mean absolute error (MAE), Schwarz Bayesian Criterion (SBC) as well as model accuracy(Q) presented in Equation (28) until Equation (32) respectively.

$$RMSE = \sum_{i=1}^n \sqrt{\frac{1}{n} (f(x_i) - h_i)^2} \quad (27)$$

$$MAE = \sum_{i=1}^d \frac{1}{n} |f(x_i) - h_i| \quad (28)$$

$$SBC = n \ln(MSE) + pa \ln(n) \quad (29)$$

MSE in Equation (30) is defined as follows.

$$MSE = \frac{\sum_{i=1}^n (o_i - h_i)^2}{n} \quad (30)$$

$$Accuracy(Q) = \frac{P_{induced}^{Correct}}{N_{p_{test}}} \times 100\% \quad (31)$$

where $f(x)$ and h_i characterized the HNN the output and the target output values respectively, d categorized the number of the permutations in HNN.

Results and Discussions

Figure 1 displayed the implementation procedure for RANKSAT in HNN. From Figure 2 until Figure 5, the performance evaluation of all HNN model for RANKSTARA for high order of logic $\forall k \leq 3$ in terms of errors accumulations, accuracy and CPU time during the program execution has been displayed. The performance has been evaluated using a different number of neurons complexities ($10 \leq NN \leq 120$). The general trend of the model performance displayed a massive increase in errors accumulation and CPU time during according to the complexity of the neuron fired to HNN in searching for optimal representation to RANKSATRA for SFDS classification.

Conclusions

The optimal behaviour for HNN models based on their global minimum ratio has been explored. The efficiency of the Election algorithm (EA) has been observed in comparison with the other metaheuristics search approaches in the literature in enhancing the performance of HNN for RANKSAT representation leading to $E_{F_{RANKSAT}} \rightarrow 0$. The HNN-RANKSAT-EA and HNN-RANKSAT-ES can retrieve more accurate neural assignment that leads

to the best global minimum solution during the training process, while the HNN-RANKSAT-ABC model trapped at some sub-optimal solutions (local solution) but managed to achieve closed to 73% success. The core objective of incorporating metaheuristics in HNN like EA is to increase the flow of HNN's learning process by reducing the sensitivity of the neuron's complexity, allowing the neurons to effectively progress to the relaxation and regeneration phase.

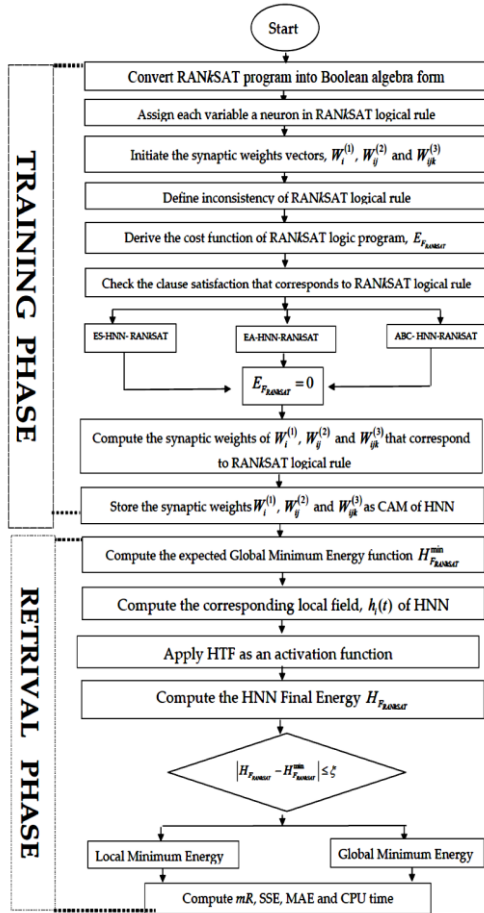


Figure 1. Flowchart for RANKSAT in HNN

Figure 2 and Figure 3 displayed the MAE and RMSE accumulation during the training and retrieval process of all model understudy for classification of SFDS data. It can deduce that all models exhibit minimal MAE and RMSE errors compared to the assigned threshold time. The general trend of MAE and RMSE for HNN-RANKSAT classification behaviour was reported to increase rapidly with neurons complexity. However, the proposed HNN-RANKSATRA-EA was able to achieve $E_{F_RANKSAT} = 0$, lower MAE and RMSE

accumulation than HNN-RANKSATRA-ABC and HNN-RANKSATRA-ES. This is due to the three optimization layers in the EA optimization process which has a better screening stages than the existing model in the solution space. That guide the entire search process to converge to the desired solution in minimum iterations which explored the robust capability of EA in lowering the sensitivity of the HNN towards the error iterations to the minimal level.

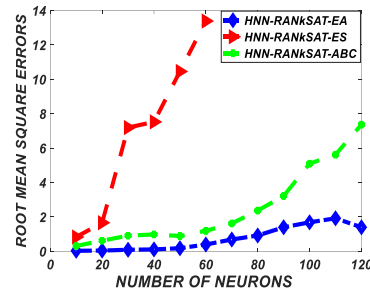


Figure 2. RMSE evaluation of SFDS for all HNN-RANKSATRA models

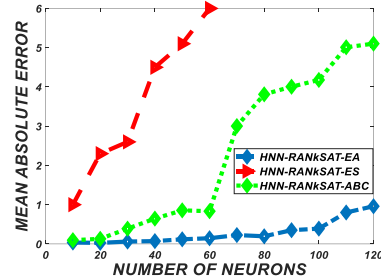


Figure 3. MAE evaluation of SFDS for all HNN-RANKSATRA models

The MAE and RMSE result displayed in Figure 2 and Figure 3 have been supported by SBC in Figure 4. The BIC evaluation is used in selecting best model out of many. The one with lower BIC is usually considered as the best model. Here BIC is used in representing RANKSATRA for the SFDS classification problem. It recorded the models' performance throughout the HNN learning phase $10 \leq NN \leq 120$. It is explicitly shown that HNN-RANKSATRA-EA outshines the HNN-RANKSATRA-ES and HNN-RANKSAT-ABC models based on the BIC assessments. In term of BIC, the performances both HNN-RANKSATRA-EA and HNN-RANKSATRA-ABC displayed a closed margin. This is because the optimization mechanism, such as a coalition in the EA searching process, is much simpler without requiring

additional iterations to reach a satisfying assignment. The non-improving solution will be enhanced by a coalition strategy during the HNN learning phase. The HNN-RANKSATRA-ES model observed to have reported an accumulation of MSE during the learning stage, as a result, more iteration needed to achieve global convergence. The accumulation of MSE tends to penalize the values of BIC. The BIC for HNN-RANKSATRA-ES is, therefore, the highest compared to the other two models. In terms of BIC assessment, HNN-RANKSATRA-EA is an acceptable approach for SFDS classification.

Figure 5, displayed the RANKSATRA models in term of implementation time during the simulation cycle. Computational time is predefined as the expanse of time needed for the network to complete the overall computational process. This explored the effectiveness of EA during the training stage with different level of complexity. It can be observed that the computational time needed to execute HNN-RANKSATRA-ES is high than HNN-RANKSATRA-EA and HNN-RANKSATRA-ABC. However, for ES, as the number of neurons is gradually increasing, more time is required due to the nature of the brute force that needed more iterations to check the clause satisfaction which resulting in overfitting the solution. HNN-RANKSATRA-EA set apart from other models because it achieved a satisfactory percentage of accuracy for SFDS data sets. Table 3 outlined the accuracy of all models based on the SFDS classifications problem. It can be observed that for SFDS classification, HNN-RANSATRA-EA and HNN-RANKSATRA-ABC enumerate the closer of accuracy with 87.1% and 72.6% respectively while HNN-RANKSATRA-ES achieved an accuracy of 58.7%. This revealed the optimal capability of both models in attaining optimized induced logic for SFDS classification. The accuracy for both methods was promising, particularly HNN-RANKSATRA-EA, due to optimized induced logic generated at the end of the executions. By executing the simulation of HNN-RANKSATRA-EA for data set classification, we can see the induced logic attained according to Equation (33),

$$P_{induced}^{Correct} = (\neg\omega_1 \vee \omega_2 \vee \neg\omega_3) \wedge (\lambda_1 \vee \lambda_2 \vee \neg\lambda_3) \wedge (\alpha_1 \vee \neg\alpha_2) \quad (31)$$

The attributes ω_1 , λ_3 , and α_2 in Equation (28) are trivial enough not to be analysed. However, attributes like λ_1 and α_1 will result in out-of-tune convergence. Considerably from Table 2, the generated induced logic by HNN-RANKSATRA-

EA in the testing stage accomplished an accuracy of 83.7%. This finding set forth HNN-RANKSATRA-EA ability to acquire an optimized induced logic that could best represent the SFCDs data set. Contrary to that, HNN-RANKSATRA-ABC, HNN-RANKSATRA-ES method deployed achieved lower accuracy compared to HNN-RANKSATRA-EA. This is due to the fact of the EA mechanism to improve the clause satisfaction process which could lead to a better training stage in resulting in a construction of an optimized induced logic. From Equation (33), we can distinguish whether the Persons is fertile or not and whether the soil is fertile and not fertile, such attributes like ω_1 and λ_1 could exhibit fair fertility status. Other than that, the induced logic could reveal insignificant and trivial attributes, like λ_3 and α_2 to sort out between fertile or not. The training stage is crucial in improving the standalone HNN system. The capacity of EA is depicted in Figure 2 until 5 by enhancing the training stage of HNN to achieve

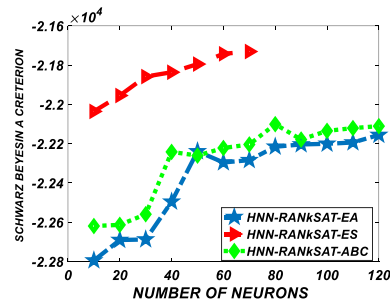


Figure 4. SBC evaluation of SFDS for all HNN-RANKSATRA models

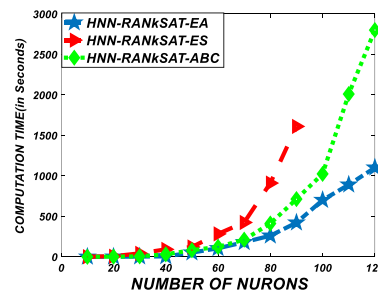


Figure 5. CPU time evaluation of SFDS for all HNN-RANKSATRA models.

Table 2 Model Accuracy

MODEL	Accuracy(%)
RANKSATRA-EA	87.1
RANKSATRA-ABC	72.6
RANKSATRA-ES	58.7

a successful solution from HNN-RANkSATRA-EA data. Because of the optimization operator used in EA, the error deviation is usually lower than the other equivalents. Advertisement and coalition in EA played a significant role in generating fewer iterations in achieving satisfactory interpretations. Aside from that, fewer iterations indicate that the model generates less RMSE, MAE and BIC. The results obtained for HNN-RAN3SATRA-ABC and HNN-RAN3SATRA-ES, on the other hand, were purposefully larger than the proposed model, especially in SFCDS. In comparison to EA, particularly ES, HNN-RAN3SATRA-ABC and HNN-RAN3SATRA-ES used an undeniably ineffective searching tool. ES initiates brute force or random search, but as the number of neurons increases, so does the complexity, causing the process to produce greater errors. However, in this research, we found out that HNN-RANkSATRA-EA works exceptionally well for SFCDS. On contrary, the HNN-RANKSATRA-EA and HNN-RANDSAT-ABC work analogously well for classification. HNN-RANkSATRA-EA outperformed HNN-RANkSATRA-ABC and HNN-RANkSATRA-ES in term of MAE, RMSE, BIC and accuracy in classifications.

According to the results, finding, analysis, and discussion of this paper, the following conclusions can be drawn: EA shows a quick convergence rate and higher result accuracy. Also, EA resulted in a lower value of RMSE, MAE, BIC and faster computation time for training HNN-RAN3SATRA. EA has proved to be an effective approach to train HNN-RAN3SATRA effectively to classify SFDS datasets with a diverse number of features and training samples. The simulation results have proven that the EA algorithm complied effectively with the Hopfield neural network Random 3 Satisfiability Reverse Analysis in terms of mean absolute error (MAE), Root mean square error (RMSE), Schwarz Bayesian Criterion (SBC) with able to classify 87.1 % of the SFDS test samples better than ABC and ES with 72.6% and 58.7% respectively. The hybrid approach has effectively searched for the optimal solution in all of the described spaces thanks to the partitioning mechanism of the search space. Finally, the HNN-RANkSATRA-EA 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.

References

- Abubakar, H. (2022). An optimal representation of optimal maximum random ksatisfiability in hopfield neural network for $k \leq 3$. *Kuwait J. of Science.*, 49(2):1-16.
- Abubakar, H. and Danrimi, M.L. (2021). Hopfield type of artificial neural network via election algorithm as heuristic search method for random boolean ksatisfiability. *International Journal of Computing and Digital Systems.*, 10(1):659-673.
- Abubakar, H., Masanawa, S. A., and Yusuf, S. (2020). Neuro-Symbolic Integration of Hopfield Neural Network for Optimal Maximum Random kSatisfiability (Maxrksat) Representation. *Journal of Reliability and Statistical Studies.*, 13(1):199-220.
- Abubakar, H., Muhammad, A., and Bello, S. (2022). Ants colony optimization algorithm in the Hopfield neural network for agricultural soil fertility reverse analysis. *Iraqi Journal For Computer Science and Mathematics.*, 3(1):32-42.
- Abubakar, H., Sabri, S. R. M., Masanawa, S. A., and Yusuf, S. (2020). Modified election algorithm in hopfield neural network for optimal random k satisfiability representation. *International Journal for Simulation and Multidisciplinary Design Optimization.*, 11(1):16.
- Abubakar, H., Sathasivam, S., and Alzaeemi, S.A. (2020). Effect of negative compaign strategies of election algorithm in solvngg optimization problem. *Journal of Quality Measurement and Analysis.*, 19(2):171-181.
- Abubakar, H., Masanawa, A.S., Yusuf, S., and Boaku, G.I. (2021). Optimal representation to High order Random Boolean kSatisfiability via Election Algorithm as heuristics search approach in the Hopfield neural networks. *Journal of the Nigerian Society of Physical Sciences.*, 3(3):201-208
- Achlioptas, D. (2009). Random Satisfiability. *Handbook of Satisfiability.*, 185:245-270.
- Biere, A., Heule, M., and van Maaren, H. (Eds.). (2009). *Handbook of satisfiability (185)*. IOS press.
- Bhuyar, V. (2014). Comparative analysis of classification techniques on soil data to predict fertility rate for Aurangabad District. *Int. J. Emerg. Trends Technol. Comput. Sci.*, 3(2):200-203.
- Crestani, F. (1995). *Neural networks for knowledge Representation and inference: DS Lavine and M. Aparicio IV (Eds.)*. Lawrence Erlbaum Associates Publishers, xv+ 503 99.95., ISBN 0-8058-1159-1.
- Emami, H. (2019). Chaotic election algorithm. *Computing and Informatics.*, 38(6):1,444-1,478.
- Emami, H., and Derakhshan, F. (2015). Election algorithm: A new socio-politically inspired strategy. *AI Communications.*, 28(3):591-603.
- Emami, S., Choopan, Y., and Parsa, J. (2018). Modeling the Groundwater Level of the Miandoab Plain Using Artificial Neural Network Method and Election and Genetic Algorithms. *Iranian journal of Ecohydrology.*, 5(4):1,175-1,189.
- Feldman, V., Perkins, W., and Vempala, S. (2018). On the complexity of random satisfiability problems with planted solutions. *SIAM Journal on Computing.*, 47(4):1,294-1,338.
- Hopfield, J. J. (1982). Rigorous bounds on the storage capacity of the dilute Hopfield model. *Proceedings of the National Academy of Sciences.*, 79:2,554-2,558.

- Neelakanta, P.S. and De Groff, D.F. (2018). Neural network modeling: Statistical mechanics and cybernetic perspectives., CRC Press. 256.
- Rojas, R. (1996). Threshold logic. In Neural Networks. Springer, Berlin, Heidelberg., 29-53.
- Sathasivam, S. (2010). Upgrading logic programming in Hopfield network. Sains Malaysiana., 39(1):115-118.
- Zubaidi, S.L., Abdulkareem, I.H., Hashim, K.S., Al-Bugharbee, H., Ridha, H.M., Gharghan, S.K., and Al-Khaddar, R. (2020). Hybridised artificial neural network model with slime mould algorithm: a novel methodology for prediction of urban stochastic water demand. Water., 12(10):2,692.