Journal IJCSM

Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
ISSN: 2788-7421

Ant Colony Optimization Algorithm in the Hopfield Neural Network for Agricultural Soil Fertility Reverse Analysis

Hamza Abubakar[1,2,*], Abdullahi Muhammad[3], Samaila Bello[4]

[1]School of Mathematical Sciences, Universiti Sains Malaysia, 11800, Penang, Malaysia
[2]Department of Mathematics, Isa Kaita Col. of Education, P.M.B. 5007, Katsina, Nigeria
[3]Department of Agricultural Science Education, Isa Kaita Col. of Education, P.M.B. 5007, Katsina, Nigeria
[4]Department of Computer Sciences, Isa Kaita Col. of Education, P.M.B. 5007, Katsina, Nigeria

*Corresponding Author: Hamza Abubakar

DOI: https://doi.org/10.52866/ijcsm.2022.01.01.004
Received October 2021; Accepted November 2021; Available online January 2022

ABSTRACT: The Boolean satisfiability problem is one of the most important decision problems in mathematical logic and computational science for determining whether or not a solution to a Boolean formula exists. The Hopfield neural network (HNN) is a major type of artificial neural network (ANN), and it is widely used to solve various optimization and decision problems due to its energy minimization mechanism. Existing models that incorporate a standalone network-projected non-versatile framework as a fundamental HNN employ random search in their training stages and are sometimes trapped at the local optimal solution. In this study, the ant colony optimization (ACO) algorithm as a novel variant of the probabilistic metaheuristic algorithm inspired by the behavior of real ants is incorporated into the training phase of HNN to accelerate the training process for random Boolean k satisfiability reverse analysis based on logic mining. The performance of the proposed hybrid model is evaluated in terms of the robustness and accuracy of the induced logic obtained by using the Agricultural Soil Fertility Data Set. Experimental simulation results reveal that ACO can effectively work with HNN for Random 3 satisfiability reverse analysis with 87.5% classification accuracy.

Keywords: Agricultural Soil Fertility Data Set; Artificial Neural Network; Random 3-Satisfiability; Ant Colony Optimisation; Logic Mining

1. INTRODUCTION

Artificial neural networks (ANNs) are one of the most popular tools utilised recently in many disciplines. They are widely used in various classification and prediction tasks and have been employed for some time in the broadly defined field of agriculture. They can be used for precision farming and decision-making systems. These networks attempt to simulate human brain capabilities. Neural networks have been utilized in the last decade as a theoretically sound alternative to traditional statistical models. They are effective in classifying data into identifiable groups or characteristics. When used in a hybrid framework with many forms of predictive neural networks, neural networks become capable of highly efficient classification [1]. Satisfiability is one of most important combinatorial optimisation problems, and it plays a major role in discrete mathematics, theoretical computational science, and operational research on classification and decision problems [2],[3],[4],[5]. ANNs have been used in modern agriculture, which requires high production efficiency and product quality. This requirement is particularly true for crop and livestock production. To meet this requirement, advanced data analysis methods, including those derived from artificial intelligence methods, are increasingly being used. One of the most popular tools of this type is ANNs, which are commonly used in classification and prediction tasks. They have also been utilised in agriculture, particularly in precision farming and decision-making systems. ANNs can replace traditional modelling methods and are one of the primary alternatives to traditional mathematical models. The range of applications for ANNs is broad. For a long time, scholars from all over the world have used these tools to help in agricultural production, thus enabling the achievement of the highest-quality products possible [6].

This study proposes a model that uses the ant colony optimisation (ACO) algorithm to facilitate the training process of ANNs and overcome the complexities of checking random clause satisfaction for generating variation and a vast searching space for real-life soil fertility data sets. To the best of the authors knowledge, no recent work has used ACO to accelerate the learning phase of Hopfield neural networks (HNNs) in extracting knowledge from agricultural soil fertility data sets. The contributions of the present study include the following: (1) formulation of a new logical rule, namely, random Boolean satisfiability (i.e. RAN*k*SAT), in HNN; integration of the ACO algorithm as a robust training algorithm into HNN for random Boolean satisfiability (RANSAT) representation; (3) implementation of a reverse analysis with the proposed model, namely, HNN-RANkSAT-ACO, to facilitate real-life application; and (4) comparison of ACO with other searching techniques on the basis of agricultural soil fertility data sets. The proposed model, HNN-RANkSAT-ACO, exhibits better performance in the training process and successfully interprets real-life data sets to determine which factors are more prominent than others and contribute to the optimisation problems. The results reveal that HNN-RANkSAT-ACO has the smallest errors and more reasonable computational time compared with other existing models for agricultural soil fertility.

The rest of this paper is organised as follows. Section 2 presents the materials and methods employed in this study, including HNN, RANSAT, random k satisfiability-based reverse analysis (RAN*k*SATRA) method, and ACO. Section 3 introduces the implementation of the HNN-RAN*k*SATRA model, the performance evaluation metrics and the experimental setup. The results and discussion are presented in Section 4. Section 5 provides the conclusions.

2. MATERIALS AND METHODS

2.1 HNN

The model structure of HNN consists of interconnected neurons and a powerful feature of content addressable memory that is crucial in solving various optimisation and combinatorial tasks [7–10]. The HNN model framework comprises organised neurons, each of which is represented by a variable known as the *Ising* variable [11]. In bipolar recognition, the neurons in discrete HNN are used; 1 is adopted to represent the true condition, and its falsification is represented by 1. The fundamental overview of neuron state activation in HNN is shown in Equation (1).

$$S_i = \begin{cases} 1, & \text{if} \quad \sum_j W_{ij} S_i > \psi, \\ -1, & \text{Otherwise} \end{cases} \tag{1}$$

where W_{ij} is the synaptic weight vector of HNN-RAN*k*SAT starting from neuron j to neuron i . S_i is defined as the state of neuron i in HNN, and ψ is the predefined value. The value $\psi = 0$ has been specified in [12–14] to certify that the networks energy decreases to zero. The synaptic weight connection in discrete HNN has no connection with itself, and the synaptic connection from one neuron to other neurons is zero (i.e. $W_{iii} = W_{jjj} = W_{kkk}$ and $W_{ii} = W_{jj} = W_{kk}$). As a result, HNN has symmetrical features in terms of architecture. The HNN model has similar intricate details as the Ising model of magnetism, as described in [15]. In bipolar expression, as the neuron state is called, the spin points implement the magnetic field trajectory. Each neuron is compelled to flip until it reaches a stable equilibrium condition in accordance with Equation (2).

$$S_i \to \operatorname{sgn}[h_i(t)] \tag{2}$$

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

$$h_i = \sum_k^N \sum_j^N W_{ijk} S_j S_k + \sum_j^N W_{ij} S_j + W_i. \tag{3}$$

The task of the local field is to evaluate the final state of neurons and generate all the possible 3-SAT-induced logic 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 generalised fitness function, E_{FRANKSAT} , controls the combinations of neurons in HNN.

E_{FRANKSAT} is presented as follows:

$$E_{F_{RANkAT}} = \sum_{i=1}^{NN} \prod_{j=1}^{V} T_{ijk}. \tag{4}$$

where V and NN are the number of variables (NV) and number of neurons (NN) generated in $F_{RANkSAT}$, respectively. The inconsistency of $F_{RANkSAT}$ representation is defined as follows:

$$T_{ij} = \begin{cases} \frac{1}{2}\left(1 - S_\rho\right), & \text{if } \neg\rho \\ \frac{1}{2}\left(1 + S_\rho\right), & \text{otherwise} \end{cases} \tag{5}$$

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

$$S_i(t+1) = \begin{cases} 1, & h_i = \sum_k^N \sum_j^N W_{ijk} S_j S_k + \sum_j^N W_{ij} S_j + W_i \geq 0 \\ -1, & h_i = \sum_k^N \sum_j^N W_{ijk} S_j S_k + \sum_j^N W_{ij} S_j + W_i < 0 \end{cases} \tag{6}$$

Equation(7) represents the Lyapunov energy function in HNN.

$$H_{F_{\mathrm{RIN}\,NAT}} = -\frac{1}{3} \sum_{i=1,i\neq j,j\neq k}^{N} \sum_{j=1,i\neq j,j\neq i}^{N} \sum_{k=1,i\neq j,k\neq i}^{N} W_{ijk} S_i S_j S_k - \frac{1}{2} \sum_{i=1,i\neq j}^{N} \sum_{j=1,i\neq j}^{N} W_{ij} S_i S_j - \sum_{i=1}^{N} W_i S_i \tag{7}$$

Equation (8) is used to classify whether a solution achieves global or local minimum energy. HNN generates the optimal assignment when the induced neuron state achieves global minimum energy. Limited studies have combined HNN and ACO as a single computational network. Thus, the robustness of ACO improves the training process in HNN. Consequently, the quality of the final neuronal state can be maintained according to Equation (8), as indicated in [12–14, 16].

$$\left| H_{F_{RANKSAT}} - H_{F_{RANKSAT}}^{\min} \right| \leq \xi \tag{8}$$

where ξ is the pre-determined tolerance value. The value $\xi = 0.001$ is from [12–14]. If the $F_{RANkSAT}$ logical representation embedded in HNN does not satisfy the criteria in Equation (8), then the neurons will be trapped in the wrong pattern in the final state.

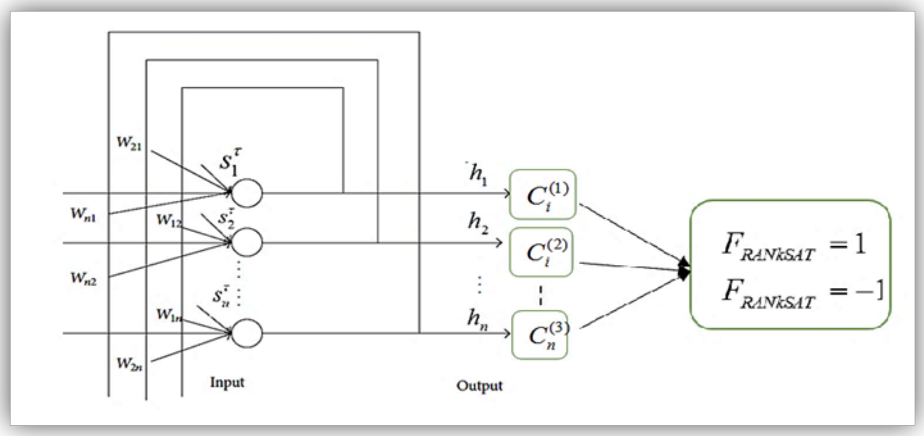

FIGURE 1. Architecture of the Hopfield type of neural network

2.2 RANDOM BOOLEAN k SATISFIABILITY

Propositional satisfiability logic (SAT) or Boolean satisfiability (BSAT) can be perceived as a logical rule that consists of clauses that contain literals or variables. RAN*k*SAT, which is also called RANBSAT, is a class of non-systematic Boolean problems consisting of a random number of variables or literals (can be the negated literals) per clause. No recent study has utilised the non-systematic behaviour of RANkSAT in discrete HNN for application in real data set classification problems. The formulation of RANkSAT logical representation consists of the following properties [7, 14].

(i) A set of variables $\{x_1, x_2, x_3, \ldots, x_n\}$ in clause C_i, where $i\forall = 1, 2, 3, \ldots, n$, that consists of x or $\neg x$.

(ii) Each $x_i \in C_i$ is connected by a disjunction (v) .

(iii) Each clause C i is connected by a conjunction $(\wedge)$.

The Boolean values in the formula for each x_i are bipolar $x_i \in \{-1, 1\}$ and exemplify the notion of FALSE and TRUE. The general formulation of $F_{RANkSAT}$ is presented in Equation (1).

$$F_{RANkSAT} = \bigwedge_{i=0}^{t} C_i^{(3)} \wedge_{i=0}^{n} C_i^{(2)} \wedge_{i=0}^{m} C_i^{(1)}, \tag{9}$$

where $t, n, m \in \square, t > 0, n > 0$ and $m > 0$. The clause $F_{RAND3SAT}$ is defined as RAND 3-SAT that consists of clause $C_i^{(k)}$ described in Equation (2) as follows:

$$C_i^{(k)} = \begin{cases} (A_i \vee B_i \vee D_i), & k = 3 \\ (A_i \vee B_i), & k = 2 \\ A_i, & k = 1 \end{cases} \tag{10}$$

In this study, F_r is designated as the conjunctive normal form (CNF) formula in which the clauses can be selected uniformly and independently without replacement amongst all $2^r \begin{pmatrix} \psi \\ t \end{pmatrix}$ non-trivial clauses of length r and $\psi = m + n$.

Notably, A_i exists in $C_i^{(k)}$ when $C_i^{(k)}$ contains either ω_i or $\neg\omega_i$. The mapping of $V(F_r) \rightarrow \{-1, 1\}$ is called logical interpretation. According to [12], the Boolean value of a Boolean formula for the representation is expressed as 1 (TRUE) and 1 (FALSE).

2.3 RANkSAT-BASED REVERSE ANALYSIS

If the most advantageous HNN-RANkSAT model is used, then logic mining will proceed quickly. The neurons (attributes) are denoted by the bipolar form 1,1. By measuring the synaptic weight (SW) between two neurons, RAN3SAT-based reverse analysis (RANkSATRA) can determine the degree to which they are connected. As a result, WAM [17] is used in the learning phase of RANkSATRA to determine the precise SW between the two neurons. By considering three attributes, namely, A, B, D where $A_i \in \{A_i, \neg A_i\}, B_i \in \{B_i, \neg B_i\}$ and $D_i \in \{D_i, \neg D_i\}$, the possible RAN3SATRA clause and its corresponding SW are obtained and stored as CAM. The cost function (CF) for bipolar expression is computed as follows:

$$E_{F_{RAND3SAT}} = \frac{1}{2}(1 - S_{A_1})\frac{1}{2}(1 - S_{A_2})\frac{1}{2}(1 + S_{A_3}) + \frac{1}{2}(1 - S_{B_1})\frac{1}{2}(1 + S_{B_2})\frac{1}{2}(1 - S_{B_3}) + \frac{1}{2}(1 + S_{D_1})\frac{1}{2}(1 - S_{D_2}). \tag{11}$$

Consistent mapping leads to $E_{F_{RANSAT}} = 0$ or the minimum energy value corresponding to all satisfied clauses in the Boolean formula. The resulting value $E_{F_{\mathrm{RAN3SAT}}}$ is proportional to the number of unsatisfied clauses in the Boolean formula. By equating the cost function in Equation (11) to the energy function in Equation (7), the corresponding SW values of HNN-RAN*k*SAT are derived. Similarly, HNN-RAN*k*SAT can be easily computed with the Wan Abdullah method (WAM) as follows:

$$H_{F_{RAN\text{-}3SAT}} = -\frac{1}{3}\begin{pmatrix} (6Y^{(3)}_{[PRQ]}S_PS_RS_Q + 6Y^{(3)}_{[PRE]}S_PS_RS_E + 6Y^{(3)}_{[PRF]}S_PS_RS_F + \\ 6Y^{(3)}_{[PRG]}S_PS_RS_G + 6Y^{(3)}_{[PRH]}S_PS_RS_H + 6Y^{(3)}_{[PRK]}S_PS_RS_K + \\ 6Y^{(3)}_{[RQE]}S_RS_QS_E + 6Y^{(3)}_{[RQF]}S_RS_QS_F + 6Y^{(3)}_{[RQG]}S_RS_QS_G + \\ 6Y^{(3)}_{[RQH]}S_RS_QS_H + 6Y^{(3)}_{[RQK]}S_RS_QS_K + 6Y^{(3)}_{[QEF]}S_QS_ES_F + \\ 6Y^{(3)}_{[QEG]}S_QS_ES_G + 6Y^{(3)}_{[QEH]}S_QS_ES_H + 6Y^{(3)}_{[QEK]}S_QS_ES_K + \\ 6Y^{(3)}_{[EFG]}S_ES_FS_G + 6Y^{(3)}_{[EFH]}S_ES_FS_H + 6Y^{(3)}_{[EFK]}S_ES_FS_K + \\ 6Y^{(3)}_{[FGH]}S_FS_GS_H + 6Y^{(3)}_{[FGK]}S_FS_GS_K + 6Y^{(3)}_{[GHK]}S_GS_HS_K \end{pmatrix} - \frac{1}{2}\begin{pmatrix} (2Y^{(2)}_{[PR]}S_PS_R + 2Y^{(2)}_{[PQ]}S_PS_Q + 2Y^{(2)}_{[PE]}S_PS_E + 2Y^{(2)}_{[PF]}S_PS_F + \\ 2Y^{(2)}_{[PG]}S_PS_G + 2Y^{(2)}_{[PH]}S_PS_H + 2Y^{(2)}_{[PK]}S_PS_K + 2Y^{(2)}_{[RQ]}S_RS_Q + \\ 2Y^{(2)}_{[RE]}S_RS_E + 2Y^{(2)}_{[RF]}S_RS_F + 2Y^{(2)}_{[RG]}S_RS_G + 2Y^{(2)}_{[RH]}S_RS_H + \\ 2Y^{(2)}_{[RK]}S_RS_K + 2Y^{(2)}_{[QE]}S_QS_E + 2Y^{(2)}_{[QF]}S_QS_F + 2Y^{(2)}_{[QG]}S_QS_G + \\ 2Y^{(2)}_{[QH]}S_QS_H + 2Y^{(2)}_{[QK]}S_QS_K + 2Y^{(2)}_{[EF]}S_ES_F + 2Y^{(2)}_{[EG]}S_ES_G + \\ 2Y^{(2)}_{[EH]}S_ES_H + 2Y^{(2)}_{[EK]}S_ES_K + 2Y^{(2)}_{[FG]}S_FS_G + 2Y^{(2)}_{[FH]}S_FS_H + \\ 2Y^{(2)}_{[FK]}S_FS_K\, 2Y^{(2)}_{[GH]}S_GS_H + 2Y^{(2)}_{[GK]}S_GS_K + 2Y^{(2)}_{[HK]}S_HS_K) \end{pmatrix}$$
$$-\left(Y^{(1)}_PS_P + Y^{(1)}_RS_R + Y^{(1)}_QS_Q + Y^{(1)}_ES_E + Y^{(1)}_FS_F + Y^{(1)}_GS_G + Y^{(1)}_HS_H\right) \quad . \tag{12}$$

The value of SW computed by WAM is stored in CAM. The optimal SWs are obtained and later used in the retrieval/recovery phase. The training process yields the optimal CF to estimate the systems best SWs. In this research, $F_{RANkSAT}$ is embedded in the proposed model, HNN-RANkSAT, on the basis of the reverse analysis technique for data classification. $F_{RANkSAT}$ helps the modified model unveil the optimal pattern or behaviour of the real data sets involved [15].

2.4 ACO ALGORITHM

The energy dynamic characteristic of HNN is applied as the second eligibility measure to select the best weight matrix for representing the RANkSAT logic program. ACO searches for the optimal parameter that best represents the set of random clauses before connecting HNN. HNN-RANkSAT-ACO is composed of seven distinct stages. In general, global improvement can be demonstrated in the minimisation problem without sacrificing generality. According to [18], the inferred components of HNN-RANkSAT-ACO are as follows.

Stage 1: *Initialisation*

Initialise bit strings $S_i(t)$ as a bipolar interpretation, where $S_i(t) \in [-1,1]$ is denoted by true and false. Every bit portrays the possible interpretation $F_{RANDkSAT}$

Stage 2: *Fitness (objective) evaluation*

Measure the fitness $S_i(t)$ of each variable according to

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

where NC is defined as the number of a clause in RANkSAT.

$$C_{ij}^{(k)} = \begin{cases} 1 \;\;, & \text{Rand } [0,1] \geq 0.5 \\ -1, & \text{Otherwise} \end{cases} \tag{14}$$

Step 3: *Pheromone density initialisation*

In this study, the pheromone for each bit of candidate group $1(-1)$ denoted by True(False) is presented in Equations (17) and (18), respectively, as follows:

$$C_{ij}(1) = [C_{i1}, C_{i2}, C_{i3}, \ldots, C_{iN}], \tag{15}$$

$$C_{ij}(-1) = [C_{i1}, C_{i2}, C_{i3}, \ldots, C_{iN}], \tag{16}$$

where each $C_{ij} \in [0,1], i = [1,2,3,\ldots,l]; j = [1,2,3,\ldots,k]$.

Step 4: *Visibility density initialisation*

The visibility density for each bit of candidate group $1(-1)$ denoted by True(False) can be represented by a real vector in Equations (19) and (20), respectively, as follows:

$$\psi_{ij}(1) = (\psi_{i1}, \psi_{i2}, \psi_{i3}, \ldots, \psi_{ik}), \tag{17}$$

$$\psi_{ij}(-1) = (\psi_{i1}, \psi_{i2}, \psi_{i3}, \ldots, \psi_{ik}) \tag{18}$$

where each i $\psi_{ij} \in [0,1], i = [1,2,3,\ldots,l]; j = [1,2,3,\ldots,k]$.

Step 5: *Ant optimisation*

The search/optimisation in ACO is carried out by the movement probability "a"$(a = 1,2,\ldots\ldots,M)$ in accordance with Equation (19) as follows:

$$p_{ij}^{a}(-1) = \frac{[C_{ij}(-1)]^{\varphi} \cdot [\psi_{ij}(-1)]^{\mu}}{[C_{ij}(-1)]^{\varphi} \cdot [\psi_{ij}(-1)]^{\mu} + [C_{ij}(1)]^{\varphi} \cdot [\psi_{ij}(1)]^{\mu}}, \tag{19}$$

where $\varphi(\forall\varphi > 0)$ and $\mu(\forall\mu > 0)$ represent the relative importance of the ACO pheromone and the visibility of the ants in the colony, respectively. Hence, the complementary ant movement in Equation (19) is written as follows:

$$p_{ij}^{a}(1) = 1 - p_{ij}^{a}(-1), \tag{20}$$

where p_{ij}^{a} represents the probability of movement from bit i to state j at time t.

Step 6: *Evaporation*

Evaporation is the decrement in pheromone and is defined as follows:

$$C_{ij}(-1)(t+1) = (1-\tau)C_{ij}(-1)(t) + \Delta C_{ij}^{\text{optimal}}, \tag{21}$$

$$C_{ij}(1)(t+1) = (1-\tau)C_{ij}(1)(t) + \Delta C_{ij}^{\text{optimal}}, \tag{22}$$

$$\Delta C_{ij}^{optimal} = \frac{1}{f\left(S_{i}^{optimal}\right)} \tag{23}$$

where τ denotes the coefficient of evaporation based on the evaporation rate and $\tau \in [0,1], f\left(S_i^{\text{optimal}}\right)$ is the number of clauses for RANkSAT.

Step 7: *Refinement*

The fitness (objective) of the solution $f(S_i)$ is determined using Equation (1). The best solution found thus far is saved as $S_i^*(t+1)$. If $S_i^*(t+1)$ is superior, then $S_i^*(t)$ is substituted for $S_i^*(t+1)$. The equation is then used to update pheromone density (Equation (3)). If $f(S_i) \neq NC$, then Steps 4-7 are repeated until the predetermined trial is completed.

3. IMPLEMENTATION PROCEDURE

3.1 PERFORMANCE EVALUATION

The performance of the proposed learning model in logic mining is measured based on the global minima ratio (Z_m) achieved, error accumulation, speed and accuracy (Q) The performance evaluation metrics used in this work are presented in Equations (26) to (30).

$$\text{Global Minima Ratio } = \frac{1}{b\rho}\sum_{i}^{n} N_{F_{\text{RaNk-SAT}}} \tag{24}$$

$$\text{Mean Absolute Error } = \sum_{i=1}^{n} \frac{1}{n}|f_t - f_c| \tag{25}$$

$$\text{Mean Absolute Percentage Error } = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{f(w_i)-y_i}{f(w_i)}\right| \quad (26)$$

$$\text{Root Mean Square Error } = \sum_{i=1}^{n}\sqrt{\frac{1}{n}(f_t-f_c)^2} \quad (27)$$

Computational (CPU) Time

Computational time is used to determine the effectiveness of the proposed model. The value of CPU is measured in seconds. The equation of CPU is

$$\text{CPUTime } = \text{ TrainingTime } + \text{ TestingTime} . \quad (28)$$

Accuracy (Q)

Accuracy is used to measure the capability of the network to train the Agricultural Soil Fertility Data Set (ASFDS) presented for classification purposes. Equation (36) is used to measure accuracy.

$$Q = \frac{P_{\text{induced}}^{\text{Correct}}}{N_{P_{\text{tert}}}} \times 100\% \quad (29)$$

3.2 EXPERIMENTAL SETUP

To obtain a good comparison between the proposed HNN-RANkSATRA-ACO logical rule and the existing HNN-RANkSATRA-ES, model implementation was carried out using the Microsoft Visual Dev C++ software on a PC with Microsoft Windows 10 64-bit OS, 500 GB hard drive, 4 GB RAM, and 3.40 GHz processor. Using the RANSAT logical rule, the reverse analysis method extracts valuable information from ASFDS for classification purposes. The RAN-3-satisfiability-based reverse analysis (RAN-3SATRA) method has been proposed as a logic mining tool that uses the HNN-RAN3SAT model to extract useful logical rules from ASFDS.

In this study, ASFDS was inputted into RANkSATRA for analysis. The data set for ASFDS was collected from @www.researchgate.net.Soil-Dataset-inCSV-format_fig1_303882815 Soil Dataset in CSV format. The collected data had 9 attributes and a total of 1988 soil samples. This experiment aimed to perform a comprehensive comparison of accuracy between HNN-RANkSATRA-ACO and existing HNN-RANkSATRA-ES by using the same data set. The details of ASFD are presented in Table 1.

Table 1. List ofattributes of ASFDS

Data Set	Details of each attribute			Output$F_{RANDkSAT}$
ASFDS (Swapna et al., 2017)	A_1	:	pH value of soil	To distinguish between high fertility and low fertility .
	A_2	:	Organic carbon, %	
	A_3	:	Phosphorous, ppm	
	B_1	:	Potassium, ppm	
	B_2	:	Iron, ppm	
	B_3	:	Zinc, ppm	
	D_1	:	Manganese, ppm	
	D_2	:	Copper, ppm	

4. RESULTS AND DISCUSSION

The effectiveness of implementing ACO in the training stage of HNN for the RANkSAT model for real data set classification was investigated. The proposed model, HNN-RANkSAT-ACO, was compared with an existing model that has entrenched real-life data sets. The proposed hybrid model's performance was evaluated in two parts. The first part investigated the quality of the solution generated by different searching techniques based on the training errors accumulated. The second part evaluated the robustness and efficiency of the proposed model on the basis of implementation time and accuracy during the training and retrieval processes. This research's main contribution is that it explores the competency of HNN-RANkSATRA-ACO for the classification of soil fertility data sets. The generated results are presented in Figures 2 to 7 and Table 2.

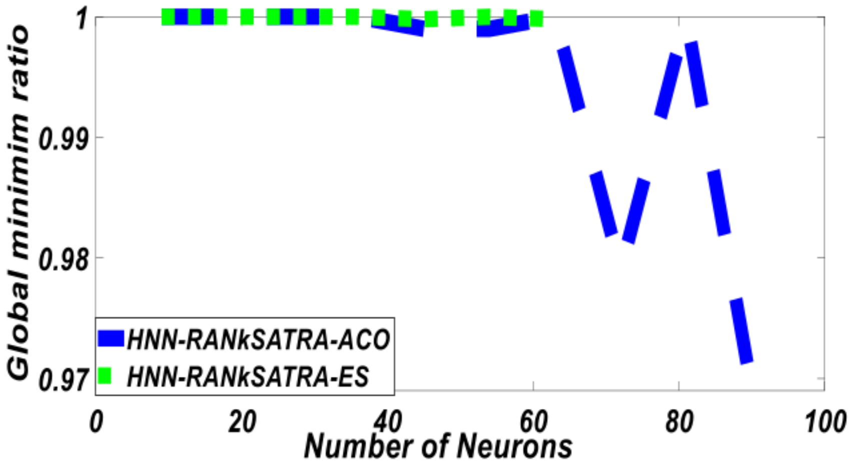

FIGURE 2. Global minimum ratio of HNN models for ASFDS

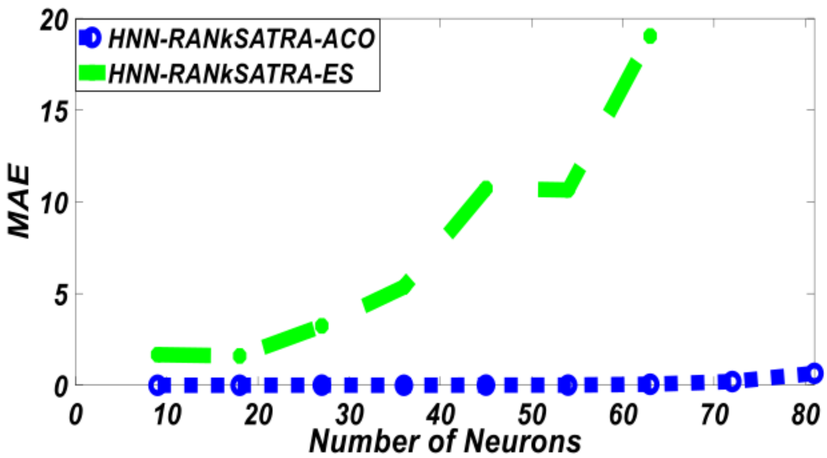

FIGURE 3. Mean absolute error of HNN models for ASFDS

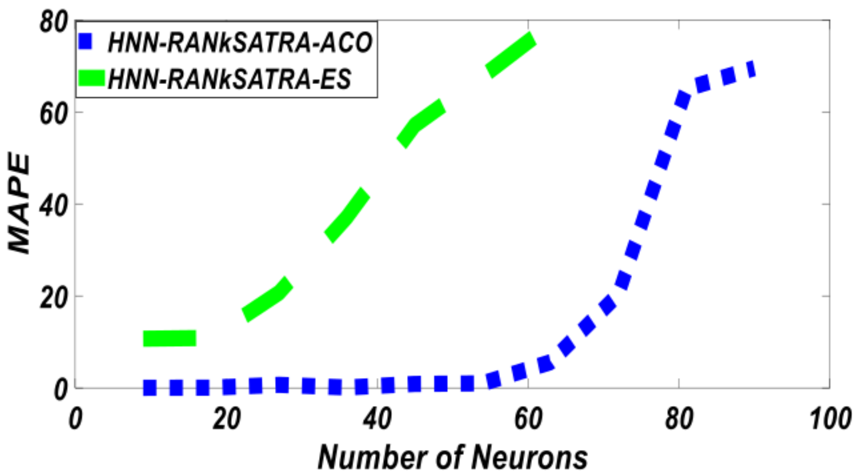

FIGURE 4. Mean absolute percentage error of HNN models for ASFDS

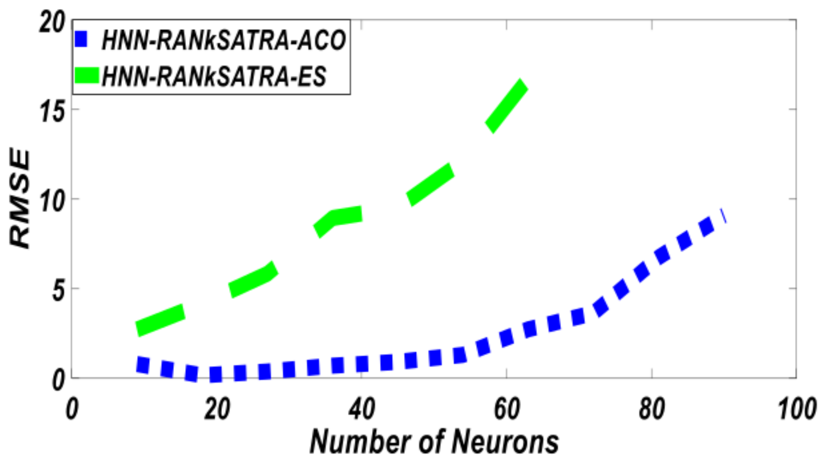

FIGURE 5. Root mean square error of HNN models for ASFDS

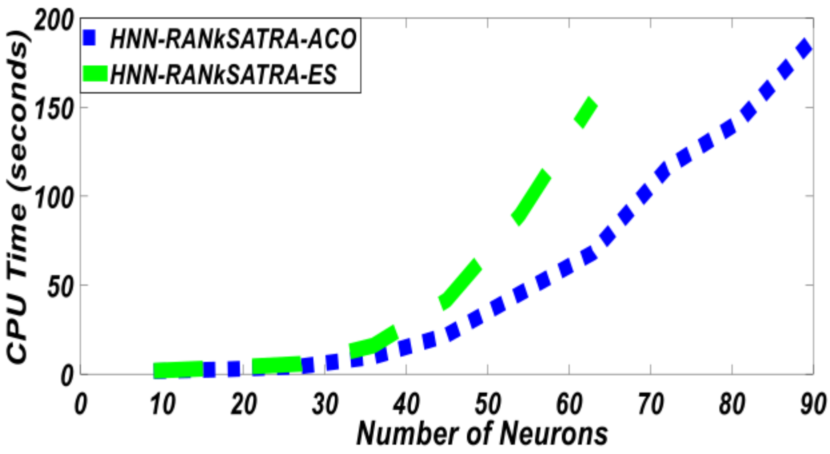

FIGURE 6. Computational time of HNN models for ASFDS

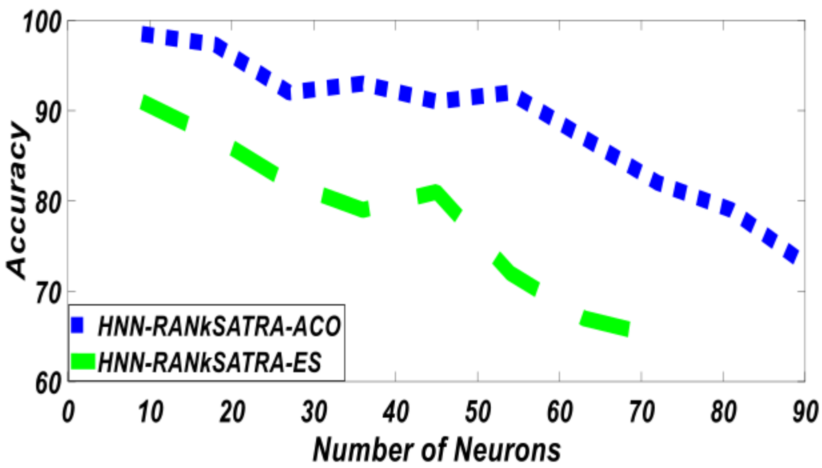

FIGURE 7. Accuracy of HNN- models for ASFDS

Table 2. Accuracy of HNN-RAND3SAT models for data set classification

Data Set	HNN-RANDSATES	HNN-RANDSATACO
ASFDS	67.7%	87.5%

In this study, the RAN3SATRA logical rule was optimally represented when ACO was incorporated into the learning phase of HNN for ASFDS classification. The result was compared with that of exhaustive search (ES) by using the same data set. A consistent value of zero was displayed, indicating that ACO provided a better and more well-trained HNN framework. However, the results of HNN-RANSAT-ACO displayed a similar trend in the searching process at the initial stage. The error accumulation in HNN-RANDSATES increased as the number of neurons (NN) increased because the training stage of HNN has no modification. Figures 2 and 3 show the models' performance in terms of Zm for the classification of ASFDS. A good model has a global minimum ratio that is close to 1. Figure 3 indicates that the global minimum ratio approached 1 even with the increasing complexity of networks and different NNs $(9 \leq NN \leq 90)$ These results reveal the optimal capability of the training methods employed by HNN-RANSATRA to attain the optimal solution that leads to the global minimum ratio. Figures 2 to 5 present the MAE, MAPE and RMSE accumulations, respectively, during the training and retrieval process in the classification of ASFDS. Minimal errors were observed compared with the assigned threshold time. The proposed HNN-RANkSAT-ACO converged to $E_{F_{RANKSAT}} = O$ with lower error accumulation than the existing HNN-RANkSAT-ES. The proposed HNN-RAN*k*SAT-ACO converged and achieved the best global solution with lower MAE, MAPE and RMSE accumulations than HNN-RANkSAT-ES for ASFDS. Figure 6 shows the trend of ACO behaviour in searching for HNN-RANkSAT models in terms of implementation time during the simulation cycle. The proposed HNN-RANkSAT-ACO was faster than HNN-RAN*k*SAT-ES in the optimisation process. Computational time is predefined as the expanse of time needed for a network to complete the overall computational process. This study explored the effectiveness of ACO during the training stage with different levels of complexity. The computational time needed to execute HNN-RANkSAT-ACO was considerably different from that for the existing HNN-RANkSAT-ES model. HNN-RANSAT-ACO attained faster execution because the ACO algorithm has fewer parameters compared with ES. However, for ES, as the number of neurons increased gradually, more time was required due to the nature of the brute force that needed numerous iterations to check the clause satisfaction, resulting in overfitting the solution. The accuracy achieved by the proposed method for the classification of ASFDS was close to 87.5%, whereas 67.7% was achieved by the existing model. This result reveals the optimal capability of the models in attaining optimised induced fertility logic for ASFDS classification. The accuracy of the methods applied in this study is promising, particularly that of HNN-RANkSAT- ACO, due to the optimal induced logic generated at the end of the executions. On the basis of the simulation executed via HNN-RANkSAT-ACO for data set classification, the induced logic was attained with Equation (30).

$$P_{\text{induced}}^{\text{Correct}} = (\neg\omega_1 \vee \omega_2 \vee \neg\omega_3) \wedge (\lambda_1 \vee \lambda_2 \vee \neg\lambda_3) \wedge (\delta_1 \vee \neg\delta_2) \tag{30}$$

According to Equation (31), the attributes ω_1, λ_3 and δ_2 are trivial enough not to be analysed. However, attributes like λ_1 and δ_1 result in out-of-tune convergence. As shown in Table 2, the generated induced logic via HNN-RAN*k*SAT-ACO at the testing stage was accomplished with close to 87.5% accuracy. This finding shows HNN-RAN*k*SAT-ACOs capability to acquire an optimal fertility induced logic that could best represent the data set. By contrast, the HNN-RANSATES model achieved lower accuracy than HNN-RANkSAT-ACO due to the fact that ACO possesses a mechanism that improves the clause satisfaction process, which could lead to improved training stages that result in the construction of an optimised induced logic. From Equation (31), we can distinguish whether an individual is fertile or not and whether the soil is fertile or infertile, and attributes such as ω_1 and λ_1 could denote a fair fertility status. Moreover, the induced logic could reveal insignificant and trivial attributes, such as λ_3 and δ_2, to distinguish between fertile or infertile.

The training stage is crucial in improving the standalone HNN system. Figures 2 to 6 show the capability of ACO to enhance the training stage of HNN and achieve a successful solution from the HNN-RANkSAT-ACO model. The error deviation is usually lower than the others because of the optimisation operator in ACO. The visibility density initialisation and evaporation and refinement mechanisms in ACO played a significant role in generating few iterations for achieving the best interpretations. Few iterations indicate that the model generates low MAE, MAPE and RMSE. In comparison with ACO, ES uses an undeniably ineffective searching tool. ES initiates brute force or random search, but as the number of neurons increases, so does complexity, which causes the process to produce large errors.

The optimisation operator in ACO introduces a large searching space and variance of the solution, leading to a strong training point. The main drawback of HNN-RANkSAT-ACO is the lack of variability of the generated induced logic, which causes overfitting. To overcome such this drawback, alteration of the data sets is crucial. Rearrangement and permutation of the attributes with a randomised selection of attributes should be implemented. However, in this research, we found out that HNN-RANkSAT-ACO works exceptionally well for ASFDS.

5. CONCLUSION

ANN possesses a comprehensive structure of training and testing stages, making it one of the most efficient tools in pattern and knowledge extraction for solving real-life applications for classification, forecasting, risk analyses, detection and quantitative analyses. In this study, ACO displayed a significant improvement in RANkSAT logical representation as a logic mining tool for controlling HNN performance in extracting the behaviour of a real data set (ASFDS). ACO showed a high convergence rate and accuracy, which resulted in low values of MAE, MAPE and RMSE and fast computation for training HNN for RANkSATRA in classifying ASFDS with diverse features and training samples. The simulation results proved that ACO complied effectively and efficiently with the neural network for RAN3SATRA with 87.5% classification accuracy for the ASFDS test samples. Therefore, we presume that this research contributes to amplifying the efforts to enhance the capability of fundamental HNN via metaheuristic algorithms for Boolean satisfiability. The proposed hybrid model can assist in obtaining the best RANkSATRA logical rule in governing the behaviour of ASFDS as a logic mining tool. In the future, we will use the same model to classify various real-life data sets, such as medical, credit and financial data sets. We will also hybridise other novel metaheuristic algorithms, such as artificial bee colony, firefly and dragonfly, with other types of ANN and other variants of Boolean satisfiability for the optimal classification of ASFDS.

ACKNOWLEDGEMENTS

The first author thanks the reviewers for their useful suggestions that improved the presentation of this paper.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] J. Ker, L. Wang, J. Rao, and T. Lim, Deep Learning Applications in Medical Image Analysis, *IEEE Access*, 2017, doi: 10.1109/ACCESS.2017.2788044.
[2] A. Layeb, A. H. Deneche, and S. Meshoul, A new artificial immune system for solving the maximum satisfiability problem, 2010, doi: 10.1007/978-3-642-13025-0_15.
[3] S. A. Alzaeemi, S. Sathasivam, M. Shareduwan, and M. Kasihmuddin, MAXIMUM 2-SATISFIABILITY IN RADIAL BASIS FUNCTION, 2020; vol. 16, no. November 2019: pp. 107115.
[4] S. Alzaeemi, M. A. Mansor, M. S. Mohd Kasihmuddin, S. Sathasivam, and M. Mamat, Radial basis function neural network for 2 satisfiability programming, Indones. J. Electr. Eng. Comput. Sci., 2019, doi: 10.11591/ijeecs.v18.i1.pp459-469.
[5] S. Abdulhabib, S. Alzaeemi, S. Sathasivam, M. Velavan, and M. Mamat, Artificial Immune System Algorithm for Training Symbolic Radial Basis Function Neural Network Based 2 Satisfiability Logic Programming, 2021; vol. 12, no. 12: pp. 25912600.
[6] M. Pathan, N. Patel, H. Yagnik, and M. Shah, Artificial cognition for applications in smart agriculture: A comprehensive review, Artificial Intelligence in Agriculture. 2020, doi: 10.1016/j.aiia.2020.06.001.
[7] H. Abubakar, S. Yusuf, and S. A. Masanawa, Exploring the Feasibility of Integrating Random k-Satisfiability in Hopfield Neural Network, 2020. [Online]. Available: www.ModernScientificPress.com/Journals/ijmms.aspx.
[8] H. Abubakar and S. Sathasivam, Developing random satisfiability logic programming in Hopfield neural network, 2020, doi: 10.1063/5.0018058.
[9] Abubakar Hamza, Effect nagetive Campaign Strategy of Election Algorithm (EA) in Solving Optimzation Probelm, 2020; vol. 16, no. November 2019: pp. 171181.
[10] H. Abubakar and M. L. Danrimi, Hopfield type of Artificial Neural Network via Election Algorithm as Heuristic Search method for Random Boolean kSatisfiability, Int. J. Comput. Digit. Syst., 2021; vol. 10, no. 1: pp. 659673. doi: 10.12785/ijcds/100163.
[11] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities., Proc. Natl. Acad. Sci. U. S. A., 1982, doi: 10.1073/pnas.79.8.2554.
[12] H. Abubakar, S. Rijal, M. Sabri, S. A. Masanawa, and S. Yusuf, Modified election algorithm in hopfield neural network for optimal random k satisfiability representation, Int. J. Simul. Multidisci. Des.Optim., 2020; vol. 16, no. 11: pp. 113.
[13] S. Sathasivam, M. A. Mansor, M. S. M. Kasihmuddin, and H. Abubakar, Election Algorithm for Random k Satisfiability in the Hopfield Neural Network, Processes, 2020, doi: 10.3390/pr8050568.
[14] H. Abubakar, S. A. Masanawa, S. Yusuf, and Y. Abdurrahman, Agent Based Computational Modelling For Mapping Of Exact Ksatisfiability Representation In Hopfiedl Neural Network Model, Int. J. Sci. tech. Res., 2020; vol. 9, no. 08: pp. 7685.
[15] P. S. Neelakanta and D. F. De Groff, Neural network modeling: Statistical mechanics and cybernetic perspectives. 2018.
[16] H. Abubakar, S. Yusuf, and S. A. Masanawa, Exploring the Feasibility of Integrating Random k-Satisfiability in Hopfield Neural Network, 2020. [Online]. Available: www.ModernScientificPress.com/Journals/ijmms.aspx.
[17] W. A. T. bin Wan Abdullah, Logic programming in neural networks, Malaysian J. Comput. Sci., 1996.
[18] L. C. Kho, M. S. M. Kasihmuddin, M. A. Mansor, and S. Sathasivam, Logic mining in football matches, Indones. J. Electr. Eng. Comput. Sci., 2019, doi: 10.11591/ijeecs.v17.i2.pp1074-1083.