Agent Based Computational Modelling for mapping of Exact *k*Satisfiability representation in Hopfield Neural Network Model

Hamza Abubakar, Sagir Abdu Masanawa, Surajo Yusuf, Yusuf Abdurrahman

Abstract— Recent studies in the field of machine learning and artificial intelligence (AI) are focusing on developing hybrid models to simplify the complexity involved in the training of the neural network. This form of simplicity is valuable for seeking an established convergence artificial neural network. In this paper, agent-based modelling (ABM) using NETLOGO as a platform has been proposed to facilitate the training process of Hopfield neural modelling in carrying Exact *k*Satisfiability programming. The developed ABM hybrid model explored the optimal task representing Exact *k*Satisfiability logic due to the simplicity, flexibility and user-friendly mannerism manifest by ABM model. ABM was used to simulate the process of taking decisions of individual movements, fortification of behaviour, group dynamics, population communications and social interactions within populations. The performance has been displayed based on Global Minimum ratio, local Minimum Ratio, Hamming Distance Mean Square Error and Computation time in evaluating the model performance. The performance of the HN model in carrying Exact *k*Satisfiability (Exact kSAT) logic was demonstrated good agreement when compared with ordinary kSatisfiability (kSAT).

Keywords— Agent base modelling, Artificial Neural network, Hopfield neural network, Satisfiability, Exact Satisfiability, logic program.

---

1 INTRODUCTION

The field of optimization is one of the important fields to be considered when searching for optimality of decision-making related problem, irrespective of its field (Oulasvirta, et al. 2020)[1]. The topic of decision-making can be illustrated by selecting elite choices in a broad range of choices. It is a rational decision that optimization and decision-makers take to select the most comfortable choices to achieve optimal performance. The area of optimization has recently gained a high degree of publicity due to the tremendous advancement of software, which yields to the achievement of excellent success in the optimization, decision, and search problems. Combinatorial optimization and searching are generally regarded as a subclass of mathematical optimization that has been employed in various field of computational research, whether or not they are simple linear or non-linear or complex in their form (Schulz et al. 2015)[2]. Such mathematical problem can be found everywhere in everyday life, such as the electronic system of vehicles, Scheduling problem, supply chain problem, the fingerprint process used by businesses, the barcode utilized in shopping, the pharmaceutical sector and other systems. The aim of these optimization and search problems is to find the best solution out of varieties of candidate solutions that meet the conditions and specifications of the problem (Aliano et al . 2018)[3]. Artificial neural networks (ANNs) are part of a family of computational architectural-based models, viewed as equivalent to brain programming by imitating their design and attempting to mimic nervous system activity through which brain information is handled (Yahaya, et al. 2019)[4]. It consists of many basic processing components (called artificial neurons) that are loosely based on biological neurons. This learns about the relations between the processing elements and the system parameters through a cycle of adjustment. It contains several interconnected neurons that have a specific synaptic weight that essentially influences how much the neuron output can influence the input to the next neurons. Usually, neurons also have their weight called "a bias" term that defines the neurons' effects on the network itself. The data is stored in the network synaptic weight and is leaned and revealed through the network synaptic weight. This step is implemented using several known heuristics (Shang, et al. 1996)[5]. Hopfield networks model (HN) are regarded as a specific type of neural networks that can preserve the certain experience, information or trends in a way similar to the human nervous system, recoverable if just partial, distorted or noisy information is presented to the network (Hopfield, 1982)[6]. This brain capacity is often referred to as memory that can be associative or content-based i.e content addressable memory (CAM). Hopfield explained that stable patterns/memories are energy minima of an HN model. The model originally proposed by Hopfield as an associative memory applied in optimization (Hopfield, 1984, Hopfield and Tank, 1985)[7-8], or its original version of "first-order Hopfield networks" (Joya, et al., 2002)[9] or in its modified / extended version of high order (Kobuch, 1991 and Takefuji, et al., 1991and Wen et al. 2009)[10-12] presented a review of HN modelling processing for in mathematical programming problem. The study described the HN model as one of the popular neural network model (NN) model applies for mathematical optimization problems. Park et al. (1993)[13] utilized the HN model in attacking the problem of economic power dispatch with a piecewise quadratic objective functions. Nguyen et al. (2006)[14] proposed a method that relies on fused imagery as an extra measure for high - resolution method – on the HNN modelling. Fung et al. (2019)[15] proposed a hybrid algorithm named Hopfield network Spatio-temporal data fusion model (HN-SPOT). Duong et al. (2019)[16] presented another modified version of the HN model with Lagrange applied to the optimization of load dispatch problem in respect of the competitive electric market. Kzar et al. (2019)[17] presented a modified HN model algorithm based on remote sensing imagery purposely for classification in the total suspended solids concentrations problem arising in the liquids. Salcedo-Sanz (2004)[18] presented a hybrid HN model with simulated annealing

---

- *Hamza Abubakar is currently pursuing PhD degree program in Mathematical Sciences at Universiti Sains Malaysia, zeeham4u2c@yahoo.com*
- *Sagir Abdu M. is a Lecturer with the dept. of Masthematical Sciences,Federal University Dutsin-Ma. Nigeria.*
- *Surajo Yusuf is a Lecturer with Isa Kaita College of Education, Dutsin-Ma, Katsina,Nigeria*
- *Yusuf Abdurrahman is a Lecturer with Isa Kaita College of Education, Dutsin-Ma, Katsina, Nigeria*

algorithm in finding an optimal solution frequency assignment problem in the field of a communications problem. In this work, the main focus is to introduce the optimal capacity of learning in HN model

2 Agent-based Modelling

The fact that even complicated patterns can be described by computational or analytical models depend on a few fundamental rules of decision is remarkable and such modelling remains a vibrant and crucial research area in an artificial neural network. The core concept behind agent-based modelling is to characterize a structure as a starting point, applying its constituent parts. The "agents" demonstrate the physical properties at their fingertips, and the entire network is then based on the behaviour and interactions of the agents in the system. Thus, the dynamics of this bottom-up method come from a microscale, as opposed to macroscopic simulation models, where complexities are often implemented by equations representing parameters of the model. Agent-based modelling (ABM) applies to the various industrial disciplines in which network can be observed as comprising a complex number of interactive entities ABM has been successfully utilized in modelling dynamics in many areas of research. This include, evacuation models proposed by (Chen et al. 2008)[19], where the network has been describing intuitive as an interactive agent. Every agent's goal is straightforward but interesting consequences may emerge due to their interactions. Agent-based models applied to simulations of traffic operations control (Chen et al. 2010, Bouarfa, et al. 2016, Balmer et al. 2004 and Hofer, et al. 2018)[20, 21, 22, 23]. The objectives of the agents are clear when it comes to controlling traffic operations, but the result of interaction with other agents in the environment, they might work differently from what they would do, they may use all roads on their own, resulting to congestion or even to halt-and-go traffic situation. Grignard et al. (2018)[24] modelled ABM in the city planning field. Where mobile agent models with different needs can provide crucial input on how efficient a specific city plan would be. A new generic algorithm that generates an artificial micro-world which enable the modeller to quickly and easily create and update new mobility scenarios. Various applications of ABM can also be found in social sciences (Davidsson, 2007)[25], Organizational Research (Grand, et al. 2019)[26], tourism (Nicholls, et al. 2017)[27] and the simulation of human systems (Bonabeau, 2002)[28]. ABM can be found in the modelling of ecology and Land Degradation (Alijani, et al. 2017)[29] and complexity research (Mandes et al. 2017)[30], Electricity markets for sustainable carbon eco-systems. (Babic et al . 2016)[31]. ABM (Kaplar, 2017)[32] has been used to develop an agent-based distribution strategy for an optimal solution to the Traveling Salesman Problem. In more related studies agent-based modelling was developed based on logic programming (Alzaeemi, et al. 2018)[33], Abubakar, et al. 2020)[34], Azim, et al. 2019 and Li, et. 2019)[35, 36]. This can be best understood through models that see this activity not as an external function of the system, but as something that occurs at the agent level and is disseminated across the network. The each ABM can develop based on the following requirements: (i) definition of agents in the environment; (ii) describing the environment; (iii) describing what agents recognize, and what they experience; (iv) describing the goal of the agents in the environment; and (v) Finding rules regarding their behaviour. Although the first four steps are relatively simple for most models, the last step poses some specific challenges. The rules should use the knowledge that agents have access to as input and include what action they take as output in this case, despite the minimal rationality (Simon, 1972)[37]. The input that agents are using to make their decisions is easy enough to find and validate. In the evacuation model, for example, dealt with quickly their immediate environment and select their course accordingly, but they have little understanding of the incident behind a physical barrier such as a wall. In the traffic model, agents know which roads are typically congested and which are not but may have access to information on the current construction sites or accidents and base their decisions on this intelligence data. Nonetheless, determining the rules that lead to a conclusion is much more complex and frequently relies on psychological or economic assumptions, which are often difficult to justify, either empirically or theoretically.

A variety of interesting NP-complete issues like Exact *k*Satisfiablity can be closely connected into a set of optimization problems, such as Exact Hitting Set (Carastan-Santos et al. 2017)[38], minimum hitting set(Shi et al. 2010)[39] and Exact cover (Bisoyi et al. 2017)[40], monotone ExactSAT is the same as the exact hitting (Fomin, et al. 2019)[41], set hypergraphs problem onset (Korula, et at. 2009)[42], Exact graph colouring problem (Guo et al. 2018)[43], exact independent set (Xiao, et al. 2017)[44] and it is closely related to the set partitioning problem (Björklund, et al. 2009)[45]. Both problems are NP-complete problems that can be transformed and represented into Exact*k*SAT and have many applications in combinatorial optimization that falls into the category of optimization, sorting, decision or counting scheme for the solution to the problem of satisfiability is sub-optimal and partially heuristic in nature. However, we are not aware of any previous research work that model the mapping of the Exact *k*Satisfiability of the Boolean formula in the HN concerning a group of individuals in an agent-based model using the Netlogo framework. We are involved in developing a new approach to modelling NN to be able to catch essential neuron behaviours. The contributions of the present study include the following: (1) To integrate the new logical representation namely Exact *k*SAT in Hopfield network model (HN); (2) To develop a hybrid agent-based computational model in Hopfield network model (HN); (3) To implement newly proposed hybrid Hopfield network for Exact *k*SAT based on agent-based computational modelling Netlogo; (4) To explore the feasibility of HN model for Exact *k*SAT mapping term of a global minimum solution, Local Minimum solution, Hamming Distance, Mean square Error and Computation time; (5) Compare the HN model for Exact *k*SAT with the existing kSAT model.not recreated, but rather converted into the final published version.

3 Exact *k*Satisfiability of a boolean formula

The Exact Satisfiability of a Boolean formula is considered as a decision problem which decided a Boolean formula written in Conjunctive Normal form (CNF) has a truth interpretation satisfying exactly one literal in each clause or determine that no such label assignment exists; The Exact *k*Satisfiability (Exact *k*SAT) logic is an important variant of Satisfiability (SAT), where the input instance is the same but the question is that in Exact *k*SAT a clause is satisfied if exactly one of its literals is true (instead of at least one literal, as in ordinary *k*SAT). Exact3SAT is also called One-In-Three Satisfiability. Exact kSAT is NP-complete even if subjected to clauses that contain at most three literals and all variables that occur only unnegated (Dahllöf, et al. 2004)[46].

Consider a Boolean expression $Q_{ExactkSAT}$ that is built from Boolean variables in CNF which has the following properties.

i. A set of variables $(x_1, x_2, x_3, ..., x_n)$ where $x_i \in \{1, -1\}$;
ii. A collection of literals. literal represents a boolean variable or its negation (x_i or $\neg x_i$);
iii. A collection of m distinct logical clauses $C_i \in (c_1, c_2, c_3, ..., c_m)$;
iv. Each satisfying mapping labels exactly one literal in a clause;
v. Each boolean variables in the clause are linked by connectives OR $(\vee)$;
vi. Each clause consists of literal linked by a Boolean connective AND $(\wedge)$;
vii. Each clause C_i is a disjunction of exactly three literals.
viii. Each clause C_i contains at most three literals.

These properties simplify the formulation of Exact *k*SAT via Hopfield network model (HN) and still preserves NP-completeness (Schaefer, 1978)[47]. The Boolean values for each x_i are bipolar $x_i \in \{-1, 1\}$ that exemplifies the notion of FALSE and TRUE respectively. The general formulation of Exact *k*SAT is presented as follows:

$$Q_{ExactkSAT} = \bigwedge_{i=1}^{k} C_i \tag{1}$$

when $k = (1,2,3)$ Equation (1) describes the Boolean formula for Exact *k*Satisfiability containing logical clause C_i given in Equation (2) as follows:

$$C_i = \bigvee_{j=1}^{3} \left(E_{ij}, D_{ij}, F_{ij}\right) \tag{2}$$

where $E_{ij}, D_{ij}, F_{ij} \in [1, -1]$. If the context is clear we denote the number of clauses of some formula Exact*k*SAT (Porschen et al. 2014)[48]. The Boolean formula for the mapping Exact *k*Sat in HN model is represented in bipolar form.

Examples for Exact *k*SAT formulation $k = 3$ is presented as follows according to (Schaefer, 1978)[47].

$$Q_{ExactkSAT} = (E_1 \vee E_2 \vee E_3) \wedge (D_1 \vee \neg D_2 \vee D_3) \wedge (\neg F_1 \vee F_2 \vee F_3) \tag{3}$$

Equation (3) is satisfiable since it gives true value resulting $Q_{ExactkSAT} = 1$. If the neuron states are considered as $E_i (i = 1,2,3)$, $D_i (i = 1,2,3)$ and $F_i (i = 1,2,3)$. The Boolean expression will be unsatisfiable if $Q_{ExactkSAT} = -1$. In this study, $Q_{ExactkSAT}$ will be embedded in HNN as a proposed model, Exact *k*SAT-HN model in the next section. The properties of Exact *k*SAT can be used to govern the behaviour of discrete HN model. In optimization terms, satisfiability problem is classified as a variant of combinatorial optimization in the discrete domain and it first problem to belongs to NP-hard class in most of its variants (Cook, 1971) [49]. A calculus-based search and optimization techniques cannot easily address this type of problem. A strong method is required that can behave in a dynamic way to find an optimal or near-optimal solution to this kind of problem. To the author's knowledge, the HN model has not yet been used to address Exact Satisfaction Problem in Agent-Based Modelling framework.

4 Mapping Of Exact *k*Satisfiability in Hopfield Network

The HN model is made up of various components which include the inputs, outputs as well as the synaptic weights. The other two major components of the HN model are its learning mechanism and its energy function. The interconnection of all these functionalities into the HN model makes it an attractive tool for optimization. The basic architecture and structure of the HN model consist of discreet interconnected bipolar neurons with no hidden neurons (Hopfield and Tank 1985)[8]. The synaptic weights are entirely symmetrical, without self-mapping between the interconnected neurons. Hence, the Content Addressable Memory (CAM) of HN model is studied as a dynamical storage network for the synaptic weights (Joya et al. 2002 and Kzar et al. 2016)[9, 17].

Given an initial vector that is mapped to the neuron state S_i^{η} where $i \in N$ and with no noise interference, the HN model will converge to a state of equilibrium corresponding to the constant value of $H_{Q_{ExactkSAT}}^{\min}$ (Barra, 2018 and Sathasivam et al. 2020) [50, 51]. Henceforth, the final configuration of the HN model corresponds to the optimal representation to the combinational problem. Neurons in HN model are represented in a bipolar form $S_i^{\eta} \in (-1, 1)$ in conformity to the dynamics of the model structure $S_i^{\eta} \to \operatorname{sgn}(h_i)$ described by *Ising* variables in spin glass and the Dean's problem of mechanical physics (Sherrington, 2010) where the local field, h_i is described as follows;

$$h_i(t) = \sum_{j=1, i \neq j}^{m} J_{ij}^{(2)} S_j(t) + J_i^{(1)} \tag{4}$$

The advantage of using bipolar values over binary values is the symmetry of the states of the network. If some pattern S_i^{η} in bipolar form is stable, its inverse is stable too whereas. The HN model can be general updated asynchronous obeying the following rule;

$$S_i^{\eta}(t+1) = \begin{cases} 1, \ if \ \sum_{j}^{N} J_{ij} S_j(t) + \varphi_j \\ -1, \ otherwise \end{cases} \tag{5}$$

where J_{jk} is the synaptic connection matrix that established the connections between *j* to *k* neurons, S_i^{η} defines the unit condition *k* and φ_j described the threshold function of neurons *j*. Several studies (Joya et al. 2002, Duong, et al. 2019, Kzar, et al. 2016, Salcedo-Sanz, et al. 2004, Barra et al. 2018, Bag, et al. 2019, Peng, Barra, 2018 and Sathasivam et al. 2020)[9, 15, 16, 17, 18, 50,51] defined $\tau_j = 0$ to verify that the HN model always leads to a decrease in energy monotonically. Each time neuron was connected with J_{jk} , the value of the synaptic connection will be preserved as a stored pattern in an interconnected vector where $J^{(1)} = [J_{jk}^{(1)}]_{n \times n}$, $J^{(2)} = [J_{jk}^{(2)}]_{n \times n}$ and $J^{(3)} = [J_{ijk}^{(3)}]_{n \times n}$ or *N*-dimensional variable vectors $\tau = (\tau_1, \tau_2, ..., \tau_N)^T$ that the constraint of synaptic weight matrix $J^{(1)}$. self-loop neuron connection is not allowed in HNN model $J_{jjj}^{(3)} = J_{kkk}^{(3)}, ..., = J_{iii}^{(3)} = 0$. It follows the symmetrical neuron synaptic weight matrix $J_{kji}^{(3)} = J_{ijk}^{(3)} = J_{jik}^{(3)}$. The HN model energy dynamics function and CAM offers a versatile system with high capacity, error tolerance, rapid memory recovery and partial inputs (Hopfield, 1984, Wen et al. 2009, Fung, et al. 2019, Alzaeemi etal. 2018, Barra, 2018 and Sathasivam et al. 2020) [7,8, 12, 15, 33, 50, 51]. To make HN ideal for incorporation with combinatorial optimization such as SAT. HN model can be used as a logical rule to instruct the behaviour of the system configuration based on the synaptic strength matrix. Exact *k*SAT can be integrated into HN model by allocating the established cost function to each variable with neurons. Besides the generalized cost function $E_{Q_{ExactkSAT}}$ that governs the combinations of HN and $Q_{ExactkSAT}$ is given in eq. (6).

$$E_{Q_{ExactkSAT}} = \sum_{i=1}^{NC} \prod_{j=1}^{m+n} T_{ijk} \tag{6}$$

where NC and $m+n$ represent the number of clauses and the number vector variables $Q_{ExactkSAT}$ respectively. Note that, the inconsistency of $Q_{ExactkSAT}$ is given as;

$$T_{ij} = \begin{cases} \frac{1}{2}\left(1 - S_\rho\right), if \neg\rho \\ \frac{1}{2}\left(1 + S_\rho\right), otherwise \end{cases} \tag{7}$$

The real value obtained $E_{Q_{ExactkSAT}}$ is corresponding to the number of “inconsistencies” of the clauses $(E_{ij} = -1,\ D_{ij} = -1,\ F_{ij} = -1)$. Minimum $E_{Q_{ExactkSAT}}$ mapping to the “most consistent” selection of S_i^η. Hence, the updating rule for $E_{Q_{ExactkSAT}}$ in HN model obeys Equation (4) and (5) can be upgraded to embraced third-order connection is defined in equation (8) and (9) respectively as follows.

$$h_i(t) = \sum_{i=1, i\neq j, j\neq k}^{m} \sum_{i=1, i\neq j, j\neq k}^{m} J_{ijk}^{(3)} S_j(t) + \sum_{i=1, i\neq j, j\neq k}^{m} J_{ij}^{(2)} S_j(t) + J_i^{(1)} \tag{8}$$

In this case, bipolar values for the output of each neuron are presented as;

$$S_i(t+1) = \begin{cases} 1, & \sum_{i=1, i\neq j, j\neq k}^{m} \sum_{i=1, i\neq j, j\neq k}^{m} J_{ijk}^{(3)} S_j(t) + \sum_{i=1, i\neq j, j\neq k}^{m} J_{ij}^{(2)} S_j(t) + J_i^{(1)} \geq 0 \\ -1, & \sum_{i=1, i\neq j, j\neq k}^{m} \sum_{i=1, i\neq j, j\neq k}^{m} J_{ijk}^{(3)} S_j(t) + \sum_{i=1, i\neq j, j\neq k}^{m} J_{ij}^{(2)} S_j(t) + J_i^{(1)} < 0 \end{cases} \tag{9}$$

where $J_{ijk}^{(3)}$ $J_{ij}^{(2)}$ and $J_i^{(1)}$ are third, second and first order synaptic weights vector of the embedded $Q_{ExactkSAT}$. Equation (8) and Equation (9) is necessary to ensure that neurons S_i^η always converge to $E_{ExactkSAT} \to 0$. The Lyapunov energy function $H_{Q_{ExactkSAT}}$ has been utilized to ensures the energy dynamics for the network decrease monotonically. The quality of the retrieved S_i^η can be measure by obeying the Lyapunov energy dynamics, $H_{Q_{EXACTkSAT}}, k=2$ defined in Equation (9):

$$H_{Q_{ExactkSAT}} = -\frac{1}{2} \sum_{i=1, i\neq j}^{m} \sum_{j=1, i\neq j}^{m} J_{ij}^{(2)} S_i S_j - \sum_{i=1, i\neq j}^{m} J_i^{(1)} S_j \tag{10}$$

The change in energy is always negative until the system reaches its global minimum energy. Equation (10) is a monotonic decrease with the dynamics and can be generalized to include third-order connections as follows;

$$H_{Q_{ExactkSAT}} = -\frac{1}{3} \sum_{i=1}^{m} \sum_{j\neq k}^{m} \sum_{i\neq k}^{m} J_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2} \sum_{i=1,}^{m} \sum_{j=1}^{m} J_{ij}^{(2)} S_i S_j - \sum_{i=1}^{m} J_i^{(1)} S_j \tag{11}$$

The energy value obtained based on Equation (11) will be classified as the global or local energy level. When the induced neuron configuration has reached global minimum energy, the network will produce the correct solution. Since the energy of the network finitely decreases as the network states change, hence as the number of iterations increases to its maximum level, the energy differential between states is approaching towards zero. Equation (8) demonstrates that the energy depicted from the always diminishes monotonically (Hopfied, 1984)[8]. The value of $H_{Q_{ExactkSAT}}$ obtained refers to the energy value concerning the absolute final energy achieved from $Q_{ExactkSAT}$ (Wan Abdullah, 1992)52]. As network approaches final energy, the changes in network energy approach zero. Consequently, the quality of the final neuronal state can be fully evaluated obeying the following condition.

$$\left| H_{Q_{ExactkSAT}} - H_{Q_{ExactkSAT}}^{\min} \right| \leq \xi \tag{12}$$

where ξ is The pre-defined tolerance value of HN model. If the embedded $Q_{ExactkSAT}$ does not satisfy Equation (11), the final state achieved will be stranded in a local minimum solution. It is worth noting that, $J_{ijk}^{(3)}$ $J_{ij}^{(2)}$ and $J_i^{(1)}$ It can be effectively achieved by using Wan Abdullah's learning method.

The algorithm of the HN model
Step1: Set $S_i = (S_1, S_2, S_3, \ldots, S_n)$ as the inputs pattern and store as CAM
Step2: Initialize the weights synaptic matrices, $J_{ijk}^{(3)}$ $J_{ijk}^{(3)}, J_{ij}^{(2)}, J_i^{(1)}, J_j^{(1)}$, and $J_k^{(1)}$
Step3: Compute HN model output vectors, $h_i = (h_1, h_2, h_3, \ldots h_n)$
Step4: Compute changes of HN model energy state $H_{Q_{EXACTkSAT}}$
Step5: If the condition stated in equation (12), is not satisfied, go to step 2, and update HN model output vectors.
Step6: If the condition in equation (12) satisfied, proceed to step 7
Step7: Initialize Exact *k*SAT logic program and assign neurons to variables.
Step8: If Exact *k*SAT not satisfied, go to step 2, otherwise proceed to step 9.
Step9: Compute the fitness values of the Exact *k*SAT using equation (6). If fitness criterion satisfied then end program and print solutions, otherwise, go to step 2.

If the fitness condition is not met, then the program continues iteratively, then it suspends and prints the solutions. The fitness requirement of the HN model is met if the network output condition is satisfied (halts) to some constant value (which implies that no further optimization emerges in the objective function), no constraints are violated and all the decision variables are non-negative.

5 Learning Phase in Hopfield Network Exact *k*Satisfiability

The mathematical formulation of the Exact *k*SAT logic representation is identical to the Lyapunov energy function of the HN model; hence, the proposed HN model can be used to explicitly address the Exact *k*SAT problem. Training the HN model minimizes the energy of configuration that the HN model network should "recognize". This allows the network to end up serving as a CAM system the network will converge to a "recollected" state if it is given only part of the state. For instance, if we train a five-unit HN model so that the state (1, -1, 1, -1, 1) is minimum energy, and we assign the network the state (1, -1, -1, -1, 1) it will converge to (1, -1, 1, -1, 1). Consequently, the network is adequately trained when the energy of states the network will consider is local minima.
The learning phase of HN model is implemented by incorporating the "behaviour" of Exact *k*SAT in HN model. Interesting to note that, the action of Exact kSAT can be generalized by determining the appropriate synaptic weight of Exact *k*SAT logic. Exact kSAT logical rule and HN hybridisation can be abbreviated as HN- Exact *k*SAT model. In HNN-Exact *k*SAT, deciding the correct synaptic weight for Exact *k*SAT clause is a primary objective of the learning process. Wan Abdullah (1992)[52] has pioneered the cutting-edge logic programming model in HN. The clauses given in a logic program need to be transformed into Boolean algebra form, based on the Wan Abdullah method. Then, each of the respective ground neurons is allocated one neuron. In this paper, the implementation of $Q_{ExactkSAT}$ in HN is denoted as HN- Exact *k*SAT model. However, there is no effort in the mapping of Exact *k*SAT in HN.

Consider the following Exact *k*SAT program:

$$\begin{array}{llll} Q_{ExactkSAT} = E_1, & E_2 & E_3 \leftarrow & \\ & F_2, & F_3 & \leftarrow F_1 \\ & D_1, & D_3 & \leftarrow D_2 \end{array} \quad (13)$$

Given the goal of the program

$$\leftarrow Q$$

where is the conjunction of clauses which defined the goal of the logic program in the HN model. The task of the logic program is to show that $Q_{ExactkSAT} \leftarrow Q$ is inconsistent to prove the goal Q. Sathasivam (2011) based on WA method, any logic program with a set of goal requires to find a consistent interpretation which builds up the model for any given logic program that yields Q true to Boolean representation. If the program label the vale *1* to be *true* and -1 to be the false labelling, then $\neg Q = -1$ implies a consistent interpretation and $\neg Q = 1$ indicates at least one of the clause is not satisfied. This can be represented as a combinatorial optimization problem where the “inconsistency” of Equation (13) is to be optimized (Minimization problem). The inconsistency is presented as a negation of Equation (13) after translating all clauses in Equation (13) into Boolean algebraic form; as follows,

$$\neg Q_{ExactkSAT} = \left(\neg E_1 \vee \neg E_2 \vee \neg E_3\right) \wedge \left(\neg D_1 \vee D_2 \vee \neg D_3\right) \wedge \left(F_1 \vee \neg F_2 \vee F_3\right) \quad (14)$$

The costs function of Equation (14) Boolean algebra formula for bipolar representation is that represents $Q_{ExactkSAT}$ logical inconsistency based on Wan Abdullah learning method is presented as follows:

$$E_{Q_{ExactkSAT}} = \frac{1}{2}\left(1-S_{E_1}\right)\frac{1}{2}\left(1-S_{E_2}\right)\frac{1}{2}\left(1-S_{E_3}\right)+\frac{1}{2}\left(1-S_{D_1}\right)\frac{1}{2}\left(1+S_{D_2}\right)\frac{1}{2}\left(1-S_{D_3}\right) + \frac{1}{2}\left(1+S_{F_1}\right)\frac{1}{2}\left(1-S_{F_2}\right)\frac{1}{2}\left(1-S_{F_3}\right) \quad (15)$$

where $S_{E_i}, S_{D_i}, \& \ S_{F_i}$ $(i = 1,2,3)$ represent the truth values of neurons E_i D_i & F_i respectively. In this work $S_{E_i}, S_{D_i}, \& \ S_{F_i}$ can take any of the two possible value of 1 (True) and -1(False) for deriving the cost function to be minimized. The optimum value for $E_{Q_{EXACTkSAT}} \to 0$, corresponding to the fact that all logical clauses are satisfied. The value $E_{Q_{EXACTkSAT}}$ is proportional to the number of unsatisfied logical clauses. Since the process involves a massive search space as the number of clause increases, the clause satisfaction is checked systematically by incorporating searching technique such as local searching, exhaustive searching or metaheuristics techniques such as artificial dragonflies. Since the number of correct interpretations will increase proportionally with the number of clauses, a robust training method is required to reduce the complexity (Sathasivam et al. 2010, Mansor et at. 2017, Alzaeemi et al. 2018, Sathasivam, 2020)[53, 54, 33, 50].

The $Q_{ExactkSAT}$ logic can be regarded as one of the representation problems in the HNN model. This can be implemented in the network by storing atom truth values and generating a optimize cost function when optimal clauses are represented. The value of synaptic strengths can be determined by comparing the cost function and the energy function (Wan Abdullahi, 1992)[52]. The cost function in equation (15), when programmed onto a third-order HNN energy equation in (11) yields the correct synaptic strengths of HNN-ExactkSAT as summarized in Table 2, to be stored as Content Addressable Memories (CAM) of HNN. The synaptic weights will, therefore, be used during the recovery phase. The training process provides the optimal cost function to determine the optimal synaptic weights
Equation (11) can further be expanded and simplified as follows by considering all connections associated with $Q_{ExactkSAT}(k = 3)$,

$$
\begin{aligned}
H_{Q_{ExactkSAT}} = & -\frac{1}{3}(6J^{(3)}_{[E_1E_2E_3]}S_{E_1}S_{E_2}S_{E_3} + 6J^{(3)}_{[D_1D_2D_3]}S_{D_1}S_{D_2}S_{D_3} + 6J^{(3)}_{[F_1F_2F_3]}S_{F_1}S_{F_2}S_{F_3}) \\
& -\frac{1}{2}(2J^{(2)}_{[E_1E_2]}S_{E_1}S_{E_2} + 2J^{(2)}_{E_1E_3}S_{E_1}S_{E_3} + 2J^{(2)}_{[E_2E_3]}S_{E_2}S_{E_3}\ 2J^{(2)}_{[D_1D_2]}S_{D_1}S_{D_2} \\
& + 2J^{(2)}_{[D_2D_3]}S_{D_2}S_{D_3} + 2J^{(2)}_{[D_1D_3]}S_{D_1}S_{D_3} + 2J^{(2)}_{[F_1F_2]}S_{F_1}S_{F_2} + 2J^{(2)}_{[F_1F_3]}S_{F_1}S_{F_3} \\
& + 2J^{(2)}_{[F_2F_3]}S_{F_2}S_{F_3}) - (J^{(1)}_{E_1}S_{E_1} + J^{(1)}_{E_2}S_{E_2} + J^{(1)}_{E_3}S_{E_3} + J^{(1)}_{D_1}S_{D_1} + J^{(1)}_{D_2}S_{D_2} \\
& + J^{(1)}_{D_3}S_{D_3} + J^{(1)}_{F_1}S_{F_1} + J^{(1)}_{F_2}S_{F_2} + J^{(1)}_{F_3}S_{F_3})
\end{aligned} \tag{16}
$$

In this study, the training process that yields the optimal cost function will determine the best synaptic weights for the system. The computation of optimized global minimum energy involves the correct interpretations and optimized synaptic weights. The optimized global minimum energy can be delineated as the expected global minimum energy to achieve at the end of the retrieval process. The Exact *k*SAT logic program can be treated as a combinatorial optimization problem. According to Equation (8), $Q_{ExactkSAT}$ consist of three (3) logical clauses with nine (9) variables that are randomly chosen from the set of pre-determined n variables seven (7) positive laterals and two (2) negative literals.

$$
\begin{aligned}
& S_{E_1} = -1, S_{E_2} = -1, S_{E_3} = 1, S_{D_1} = -1, \\
& S_{D_2} = 1, S_{D_3} = -1, S_{F_1} = 1, S_{F_2} = 1, S_{F_3} = -1
\end{aligned} \tag{17}
$$

Equation (19) is one of the consistent interpretations that make the entire Boolean formula $Q_{ExactlkSAT}$ true. Substituting equation (19) into equation (20), the expected global minimum energy is obtained as follows;

$$
.H^{\min}_{\square_{ExactkSAT}} = -\frac{3}{8} \tag{18}
$$

$H^{\min}_{Q_{ExactkSAT}}$ will be used to separate the correctness of the neuron state produced by the network during the retrieval phase.

6 Methodology/Implementation procedure

Implementation of Neuro-Heuristic searching method of Exact *k*SAT in HN. The program's main task is to find the best "model" that find the optimal occurrences of Exact *k*SAT Both variables and clauses were initially randomized. Simulations performed with a different number of neurons complexity ranging from $10 \le NN \le 80$. The executions of these models were carried out on Exact *k*SAT logical representation is presented according to the following steps.

i. Given a logic program:

$$
\begin{aligned}
Q_{ExactkSAT} = E_1,\ E_2\ \ E_3 \leftarrow \\
F_2,\ F_3\ \ \leftarrow F_1 \\
D_1,\ D_3\ \ \leftarrow D_2
\end{aligned}
$$

ii. Translate all the Exact *k*SAT logical clauses into Boolean algebra form as in Equation (8):

$$Q_{ExactkSAT} = (E_1 \vee E_2 \vee E_3) \wedge (D_1 \vee \neg D_2 \vee D_3) \wedge (\neg F_1 \vee F_2 \vee F_3)$$

iii. Designate neuron to each variable in an Exact *k*SAT representation.

iv. Randomize the state of the neurons and initialize all connection strengths zero.

$$J^{(3)}_{ijk} = J^{(3)}_{kji} = J^{(3)}_{jik}, J^{(2)}_{ji} = J^{(2)}_{ij} = J^{(1)}_{i} = J^{(1)}_{j} = 0$$

v. Derive the cost function, $E_{\square_{ExactkSAT}}$ for Exact*k*SAT using Equation (2) -(3).

vi. Equate the cost function Equation (2) and (3) to energy dynamics Equation (6) and obtain the values of the synaptic weight vector $J^{(3)}_{ijk}, J^{(2)}_{ij}, J^{(1)}_{i}, J^{(1)}_{j}, J^{(1)}_{k} = 0$.

vii. Check clause satisfaction that corresponds to $E_{\square_{EXACTkSAT}} = 0$. The satisfied assignment will be store as CAM in HNN.

viii. Apply Sathasivam's relaxation method via equation.

ix. Randomise the state of the neurons. Compute the respective local field $h_i(t)$ of the state space using Equation (4) and (5). If it remains unchanged after five loops, it will be considered as a stable configuration.

x. Find the corresponding final state of the network $H_{\square_{ExactkSAT}}$ by using the Lyapunov energy dynamics Equation (6). Check whether the final energy derived is a global or a local minimum based on condition in Equation (7).

Find the terms global minimum ration (Zm), Local minimum ration (Ym) and Hamming Distance (HD) and computation time of the to explore the efficiency, accuracy, robustness of implementing the mapping of Exact kSAT in HN model.

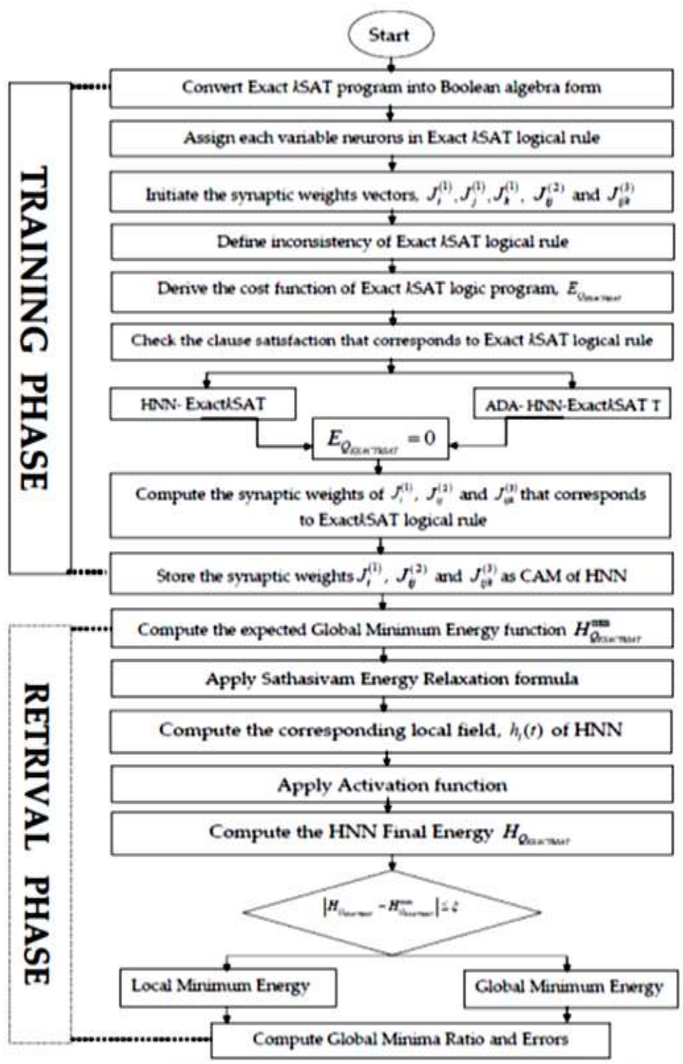

Figure 1: Flowchart for model-Exact*k*SAT implementation Procedure

7 Experimental Setup For HN-Exact *k*SAT Models

The objective functions for hybridization are for an optimal searching for to Exact *k*SAT logic representation. The proposed model will be compared with the existing model, HN model *k*SAT model in the literature. All HN models utilized simulated datasets by generated based on Exact *k*SATand *k*SAT logical clauses. In order to obtain a fair comparison among all HN models, all source code will be implemented based on simulation program developed in ABM with Netlogo Version 6.11 run on a computer equipped with Intel® Celeron® CPU B800@2GHz processor and 4GB RAM using Windows 8. Table 1 until Table 4 display the necessary parameters involved in each HN models.

Table 1. List of Parameters and their corresponding values

Parameter	HNNExact*k*SATES	HNN*k*SATES
Number of Neurons (NN)	80	80
Neuron Combination (c)	100	100
Tolerance Value (∂)	0.001	0.001
Selection_Rate (η)	0.2	0.2
No_Neuron String (μ)	100	100

8. Performance Evaluation

The evaluation on performance is a key aspect in the design process of an HN model for exact *k*SAT and HN model for *k*SAT. It is deemed that a sufficiently reliable estimate of accuracy and precision of predictions of a model is given by measurements made on the differences between the expected values and those observed, said "difference measures". Once training process finished, neural network calculated the values for the Global Minimum ratio (*Zm*), local Minimum Ratio (***Ym***), Hamming Distance (*HD*), Mean square Error (*MSE*) and Computation time (*CPU*).

$$Zm = \frac{1}{n}\sum_{i=1}^{n} H_{Q_{ExactkSAT}}$$

$$Ym = \frac{1}{n}\sum_{i=1}^{n} (H_{Q_{ExactkSAT}} - H_{Q_{ExactkSAT}}^{\min})$$

$$HD = \frac{1}{tc}\sum_{i=1}^{k} \left| \Delta H_{Q_{ExactkSAT}} - H_{Q_{ExactkSAT}} \right|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left(H_{Q_{ExactkSAT}} + H_{Q_{ExactkSAT}}^{\min} \right)^2$$

CT_Time = Learning_Time + Retrieval_Time

9 Experimental results and Discussions

In this paper, ABM based on Netlogo was used in simulating the results. To validate our proposed network, we generated the simulations result using different number of neurons from $10 \leq NN \leq 80$. The result of the simulation has been presented in Figure 2-7. The performance of simulated result generated based on HN-ExactkSAT models and existing HN-*k*SAT model have been presented as follows:

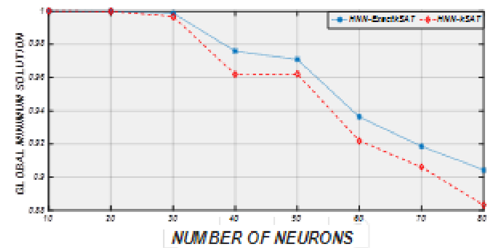

Figure 2. Global Solution of ABM-HN -Exact *k*SAT and ABM-HN-*k*SAT

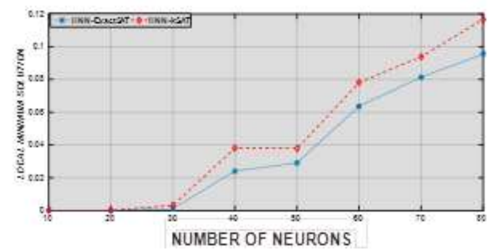

Figure 3. Local solution of ABM-HN-Exact *k*SAT and ABM-HN-*k*SAT

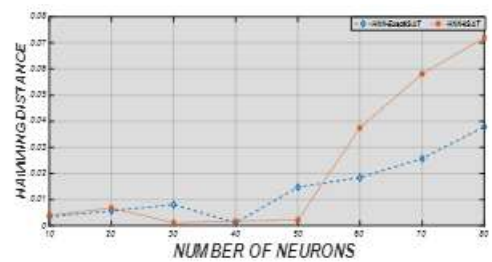

Figure 4. Hamming Distance of ABM-HNN-Exact *k*SAT and ABM-HNN-*k*SAT

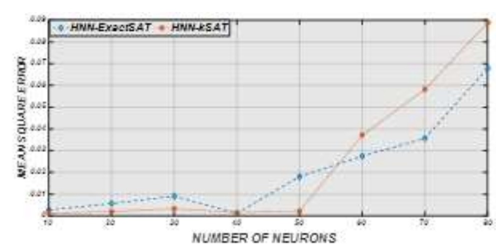

Figure 5. Mean Square Error *of ABM*-HNN-Exact *k*SAT and ABM-HNN-kSAT

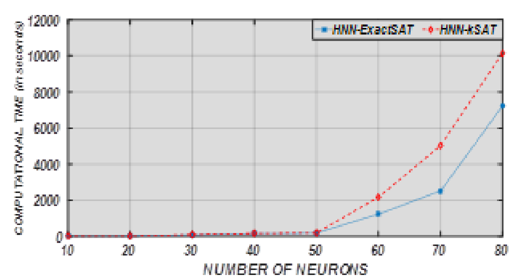

Figure 6. CPU TIME of ABM-HN-Exact *k*SAT and ABM-HN-*k*SAT

Figures 2–6 displayed the search performances HN model in carrying Exact *k*SAT and *k*SAT logical. It exhibits the behaviours of the Global solution and Local solution, HD, MSE and CPU time observed in the searching processes from $10 \le NN \le 80$. In general, both HN model displayed similar searching trend in the behaviours in both cases. When the problem becomes larger, the disparity of HN model searching trend is more noticeable (as the *NN* increases, both managed to maintain the same *NN.* The number of satisfied clause decreases sharply as the *NN* complexity raises). The HN model searching the optimal solution for Exact kSAT and kSAT logical representation is shown in this study.

Figure 2 display *Gm* achieved conducted via HN models in searching the optimal representation for Exact *k*SAT and *k*SAT logical. The capability of both hybrid networks was measured in term of *Gm* by considering different *NN* complexity $10 \le NN \le 80$. Sathasivam (2010) and Mansor et al. (2017) [53,45] clarified the connection between the Gm and Lm with the energy nature attained at the end of the simulation time. Hypothetically, if the Gm of the proposed method is approaching one, virtually all solutions in the search space have reached equilibrium globally (100 per cent clauses satisfied). According to Figure 2, HN model based on ABM Netlogo can retrieve a more accurate state for Exact *k*SAT for its ability to hunt closed to 90% success at *NN* =80. HN model manages to get more number of satisfied clauses in Exact *k*SAT than of *k*SAT. Specifically, during $10 \le NN \le 20$, HN model achieved Gm 1(100% success) higher than *k*SAT. The real task of the HN model searching is to significantly decrease the learning complexity so that the neurons can proceed to relaxation and recovery process successfully (Sathasivam et al. 2010, Abubakar et at. 2020, Alzaeemi etal. 2018, Sathasivam, 2020)[53, 54, 33, 50]. Furthermore, HN model was able to achieve closed to 90% success in both Exact *k*SAT and *k*SAT searching. It is noticeable that both logic display closed and similar searching trend in term of Gm.

In Figure 3 Local solution (Lm) is presented, the lower the Local solution, the higher the acceptability of the searching process of HN model. However, as NN complexity increase Lm will always be going away from zero. Based on Figure 3, it can be observed that the trend displays by HN model were increasing significantly from $30 \le NN \le 80$ Exact *k*SAT logic and $20 \le NN \le 80$ for *k*SAT logic. Furthermore, HN model recorded 0.875 (87.5 %) neurons at local solution for *k*SAT which is higher than 0.75 (75%) for Exact *k*SAT logic. Generally speaking, closed to 10% of the neuron's final state obtained both models are in the local solution. Although the complexity of the network is increased by increasing the *NN* affects the network searching stability and capacity in both cases. Based on the results from Figure 3, local solution portrays in both cases demonstrate a similar trend as the searching complexity increases. This is possible because both case the same only their pattern of satisfiability differed. As a result of the increasing NN, the local solution can also observe to be increasing due to the searching becoming more complicated more local minima will be recorded. The quality of Lm reported in the HN model tends to indicate that it has a substantial capacity to obtain the optimal final state of Exact *k*SAT logic.

Figure 5 displays the behaviours of HD for both Exact *k*SAT and *k*SAT logic based on ABM Netlogo. It can be observed that Exact *k*SAT and *k*SAT display lower HD from $NN \le 20$ and the HD is increasing as the NN increase. It was apparent from the HD trend in Figure 5 that HN model registered HD continuously less than 3% for all *NN* in Exact *k*SAT logic and increases significantly up to 7% until NN=80 for *k*SAT logic. Both models observed to agree with each other in term of HD by manifesting similar movement as demonstrated in Figure 5. However, even as the NN increase, the error value for HD is near to zero in both logics. The solutions obtained are closed to one, so the disparity between the stable states and the attractors is approaching zero. It is back up by having closed to zero HD values. This is because the chance for the same neuron engaged in more than one clause also increased as the number of clauses grew. The lower HD value offers profound proof of HN's ability to function well with Exact kSAT program.

In Figures 5, MSE evaluation of HN model for optimal representation of Exact *k*SAT and *k*SAT logic. The magnitude of error in performance evaluation based on MSE in this study is very important.

Based on Figure 5, it can be seen that the value of MSE recorded by HN model in searching for Exact *k*SAT and *k*SAT logic representation were increasing significantly from $10 \le NN \le 80$. The lower value of MSE manifests the high performance of HN in generating the optimal final state of the model performance. Specifically, during $10 \le NN \le 80$, HN model recorded less 7% MSE in Exact kSAT lower than kSAT logic (8%). It is explicitly shown that both models display similar behaviour in term of MSE. They both show a similar steady rise in errors mapping from $10 \le NN \le 80$. However, HN model manifests better performance in searching Exact kSAT logic. Accumulation of MSE were observed in both model during the learning stage, as a result, more iteration needed to achieve global convergence in both logic.

In this paper, the CPU time can be considered as the time taken for any given program to generate, execute and retrieve the correct Exact *k*SAT and *k*SAT logical representation. Based on Figure 5 CPU Time is required to complete one execution of learning and testing the HN model. According to Figure 5, it can be seen that CPU time increases when the number of crucial points increases. It is seen that in the HN model for ABM with Netlogo consume less time in searching for *k*SAT solution in compares with Exact *k*SAT logic may be due to the nature of searching the correct representation. The different in CPU time between the two logic is significantly large. Moreover, as the searching becomes more complex the *k* SAT searching show faster convergence compare to Exact k SAT logic. Hence, the correct searching of *k*SAT can be obtained in less computation burden for each number than Exact *k*SAT. The lower value of CPU time proves the improved performance of the proposed model in achieving model convergence. Specifically, when the $10 \le NN \le 20$, HN model executes *k*SAT logic about 90.5% faster than Exact *k*SAT logic the program. When from $30 \le NN \le 80$, the execution of Exact *k*SAT logic is faster than *k*SAT logic the program at about the difference of 93924 seconds. Both models have proven to contend with higher-order neurons, even though the computational cost is higher.

10. Conclusions

We have successfully developed a new logical representation by integrating the Exact *k*SAT in Hopfield network model (HN). The newly proposed hybrid computational mode Hopfield network for Exact *k*SAT was developed based on agent-based computational modelling Netlogo. We have also shown the feasibility of embedded Exact kSAT in HN model.

This result showed a giant leap in the implementation of the HN model itself, particularly in terms of producing the correct ssynaptic weight for effectively connecting the input to output weights with minimal errors. The influence of the optimized output weights produced by the HN model has therefore successfully optimized the training phase with varying neurons levels of complexity. It is possible to attain the ability of the Exact kSAT logical rule on ABM in HN model. based on this study, we can conclude that the conventional learning approach is not inherently efficient in output weight optimization in ANN. Our proposed model, HN model in Exact kSAT logical representation, has good agreement with the HN model in kSAT logic. The emergence of the Satisfiability Logic Rule in computation is still at an early stage and necessitates more in-depth research, particularly in the other variant of the ANN model, such as the feed-forward neural network. We will be venturing the convolution neural network convolution in the future and optimizing the potential HN model parameters. The approach would potentially open up various possibilities for real-life applications. Additionally, by incorporating robust metaheuristic algorithms such as Dragonfly algorithm, Harmony Search, Bees Algorithm, modified Flower Pollination Algorithm, Ant lion optimizer and particle swarm optimization algorithm and fine-tuning metaheuristics, the HN model learning parameters can be greatly optimized.

References

1. Oulasvirta, A., Dayama, N. R., Shiripour, M., John, M., & Karrenbauer, A. (2020). Combinatorial optimization of graphical user interface designs. *Proceedings of the IEEE*, *108*(3), 434-464
2. Schulz, A. S., Skutella, M., Stiller, S., & Wagner, D. (Eds.). (2015). *Gems of Combinatorial Optimization and Graph Algorithms*. Springer.
3. Aliano Filho, A., Moretti, A. C., Pato, M. V., & de Oliveira, W. A. (2019). An exact scalarization method with multiple reference points for bi-objective integer linear optimization problems. *Annals of Operations Research*, 1-35.
4. Yahaya Pudza, M.; ZainalAbidin, Z.; Abdul Rashid, S.; MdYasin, F.; Noor, A.S.M.; Issa, M.A. Sustainable Synthesis Processes for Carbon Dots through Response Surface Methodology and Artificial Neural Network. *Processes,* 2019, *7*, 704
5. Shang, Y.; Wah, B.W. Global Optimization for Neural Networks Training. *IEEE Computer, 1996, 29, 45-54.*
6. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 1982, *79*, 8, 2554-2558.
7. Hopfield, J. Neurons with graded response have collective computational properties like those of two-state neurons, Proc. *Natl. Acad. Sci.* USA 81, 1984, 3088-3092.
8. Hopfield, J.; Tank, D. &Neural' computation of decisions in optimization problems, *Biol. Cybernet.* 52, 1985, 141-152
9. Joya, G.; M.A. Atencia, M.A.; Sandoval, F. Hopfeld neural networks for optimization, study of the different dynamics. *Neurocomputing* 43,2002, 219-237
10. Kobuchi, Y. State evaluation functions and Lyapunov functions for neural networks, *Neural Networks* 4, 1991, 505-510.
11. Takefuji, Y., & Lee, K. C. (1991). Artificial neural networks for four-coloring map problems and K-colorability problems. *IEEE transactions on circuits and systems*, *38*(3), 326-333.
12. Wen, U.P.; Lan, K.M.; Shih, H. S. A review of Hopfield neural networks for solving mathematical programming problems. *European Journal of Operational Research*, 198, 3, 2009, 675–687.Pa
13. Park, J. H., Kim, Y. S., Eom, I. K., & Lee, K. Y. (1993). Economic load dispatch for piecewise quadratic cost function using Hopfield neural network. *IEEE transactions on power systems*, *8*(3), 1030-1038.
14. Nguyen MQ, Atkinson PM, Lewis HG. Superresolution mapping using a Hopfield neural network with fused images. *IEEE Transactions on Geoscience and Remote Sensing*. 2006 Feb 21;44 (3):736-49.
15. Fung, C.H.; Wong, M.S.; Chan, P.W. Spatio-Temporal Data Fusion for Satellite Images Using Hopfield Neural Network. *Remote Sens.* 2019, *11*, 2077.
16. Duong, T. L.; Nguyen, P. D.; Phan, V. D.; Vo, D. N.; Nguyen, T. T. Optimal Load Dispatch in Competitive Electricity Market by Using Different Models of Hopfield Lagrange Network. *Energies*. 2019, *12*, 15, 2932.
17. Kzar, A.; Jafri, M.; Mutter, K.; Syahreza, S. A Modified Hopfield Neural Network Algorithm (MHNNA) Using ALOS Image for Water Quality Mapping. *International journal of environmental research and public health*, 2016, *13*, 1-92.
18. Salcedo-Sanz, S,; R. Santiago-Mozos, R.; and Bousono-Calzon, C. "A hybrid Hopfield network-simulated annealing approach for frequency assignment in satellite communications systems," *in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* 34, 2004, 2, 1108-1116
19. Chen, X.; Zhan, F.B. Agent-based modelling and simulation of urban evacuation: Relative effectiveness of simultaneous and staged evacuation strategies. J. Oper. Res. Soc. 2008, 59, 25–33
20. Chen, B.; Cheng, H.H. A review of the applications of agent technology in traffic and transportation systems. IEEE Trans. Intell. Transp. Syst. 2010, 11, 485–497.
21. Bouarfa, S., Blom, H. A., & Curran, R. (2016). Agent-based modeling and simulation of coordination by airline operations control. *IEEE Transactions on Emerging Topics in Computing*, *4*(1), 9-20.
22. Balmer, M.; Cetin, N.; Nagel, K.; Raney, B. Towards truly agent-based traffic and mobility simulations. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1; IEEE Computer Society: Washington, DC, USA, 2004; pp. 60–67.
23. Hofer, C.; Jäger, G.; Füllsack, M. Large scale simulation of CO2 emissions caused by urban car traffic: An agent-based network approach. J. Clean. Prod. 2018, 183, 1–10
24. Grignard, A., Alonso, L., Taillandier, P., Gaudou, B., Nguyen-Huu, T., Gruel, W., & Larson, K. (2018, July). The impact of new mobility modes on a city: A generic approach using abm. In *International Conference on Complex Systems* (pp. 272-280). Springer, Cham.
25. Davidsson, P. (2002). Agent based social simulation: A computer science view. *Journal of artificial societies and social simulation*, *5*(1).

26. Grand, J. A., Gupta, P., & Trinh, M. P. (2019, July). Application of Agent-Based Modeling (ABM) in Organizational Research on Teams and Groups. In *Academy of Management Proceedings* (Vol. 2019, No. 1, p. 15008). Briarcliff Manor, NY 10510: Academy of Management.
27. Nicholls, S., Amelung, B., & Student, J. (2017). Agent-based modeling: A powerful tool for tourism researchers. *Journal of Travel Research*, *56*(1), 3-15.
28. Bonabeau, E. Agent-based modeling: Methods and techniques for simulating human systems. Proc. Natl. Acad. Sci. USA 2002, 99, 7280–7287.
29. Alijani, Z., Delbari, M., & Sarmadian, F. (2017, June). Dealing with Land Degradation: Application of Agent Based Modeling. In *79th EAGE Conference and Exhibition 2017* (Vol. 2017, No. 1, pp. 1-3). European Association of Geoscientists & Engineers
30. Mandes, A., & Winker, P. (2017). Complexity and model comparison in agent based modeling of financial markets. *Journal of Economic Interaction and Coordination*, *12*(3), 469-506
31. Babic, J., & Podobnik, V. (2016, July). A review of agent-based modelling of electricity markets in future energy eco-systems. In *2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech)* (pp. 1-9). IEEE.
32. Kaplar, A., Vidaković, M., Luburić, N., & Ivanović, M. (2017, May). Improving a distributed agent-based ant colony optimization for solving traveling salesman problem. In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1144-1148). IEEE
33. Alzaeemi, S. A. S., & Sathasivam, S. (2018). Hopfield neural network in agent based modeling. *MOJ App Bio Biomech*, *2*(6), 334-341.
34. Abubakar, H., Yusuf, S., & Abdurrahman, Y. (2020). Discrete Artificial Dragonflies Algorithm in Agent Based Modelling for Exact Boolean kSatisfiability Problem. *Journal of Advances in Mathematics and Computer Science*, 115-134.
35. Azim, M. A., Sathasivam, S., Alzaeemi, S. A. S., & Mahmood, M. (2019). Agent Based Modeling for Comparing the Performances of Hyperbolic and Zeng and Martinez Activations Functions. *International Journal of Computer Networks and Communications Security*, *7*(12), 250-257.
36. Li, W., Bai, Q., & Zhang, M. (2019). A multi-agent system for modelling preference-based complex influence diffusion in social networks. *The Computer Journal*, *62*(3), 430-447.
37. Simon, H. A. (1972). Theories of bounded rationality. *Decision and organization*, *1*(1), 161-176.
38. Carastan-Santos, D., de Camargo, R. Y., Martins Jr, D. C., Song, S. W., & Rozante, L. C. (2017). Finding exact hitting set solutions for systems biology applications using heterogeneous GPU clusters. *Future Generation Computer Systems*, *67*, 418-429
39. Shi, L., & Cai, X. (2010, May). An exact fast algorithm for minimum hitting set. In *2010 Third International Joint Conference on Computational Science and Optimization* (Vol. 1, pp. 64-67). IEEE
40. Bisoyi, S. K., & Reza, H. (2017). Toward Securing Cyber-Physical Systems Using Exact Cover Set. *J Inform Tech Softw Eng*, *7*(198), 2.
41. Fomin, F. V., Gaspers, S., Lokshtanov, D., & Saurabh, S. (2019). Exact algorithms via monotone local search. *Journal of the ACM (JACM)*, *66*(2), 1-23
42. Korula, N., & Pál, M. (2009, July). Algorithms for secretary problems on graphs and hypergraphs. In *International Colloquium on Automata, Languages, and Programming* (pp. 508-520). Springer, Berlin, Heidelberg.
43. Guo, J., Moalic, L., Martin, J. N., & Caminada, A. (2018, January). Exact Graph Coloring Algorithms of Getting Partial and All Best Solutions. In *ISAIM*.
44. Xiao, M., & Nagamochi, H. (2017). Exact algorithms for maximum independent set. *Information and Computation*, *255*, 126-146.
45. Björklund, A., Husfeldt, T., & Koivisto, M. (2009). Set partitioning via inclusion-exclusion. *SIAM Journal on Computing*, *39*(2), 546-563.
46. Dahllöf, V., Jonsson, P., & Beigel, R. (2004). Algorithms for four variants of the exact satisfiability problem. *Theoretical Computer Science*, *320*(2-3), 373-394.
47. Schaefer, T. J. (1978, May). The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing* (pp. 216-226).
48. Porschen, S., Schmidt, T., Speckenmeyer, E., & Wotzlaw, A. (2014). XSAT and NAE-SAT of linear CNF classes. *Discrete Applied Mathematics*, *167*, 1-14.
49. Cook, S. A. (1971, May). The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing* (pp. 151-158).
50. Barra, A.; Beccaria, M.; Fachechi, A. A new mechanical approach to handle generalized Hopfield neural networks. *Neural Netw*, 2018, 106, 205–222.
51. Sathasivam, S., Mansor, M., Kasihmuddin, M. S. M., & Abubakar, H. (2020). Election Algorithm for Random k Satisfiability in the Hopfield Neural Network. *Processes*, *8*(5), 568.
52. Abdullah, W. A. T. W. (1992). Logic programming on a neural network. *International journal of intelligent systems*, *7*(6), 513-519.
53. Sathasivam, S. (2010). Upgrading logic programming in Hopfield network. *Sains Malaysiana*, *39*(1), 115-118.
54. Abubakar, H., Sabri, SRM,. Abdu, SM, Yusuf,S. (2020). Modified election algorithm in hopfield neural network for optimal random *k* satisfiability representation, Int. J. Simul. Multidisci. Des. Optim. **11**, 16.