Pertanika J. Sci. & Technol. 28 (3): 983 - 1008 (2020)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Modified Imperialistic Competitive Algorithm in Hopfield Neural Network for Boolean Three Satisfiability Logic Mining

Nur Ezlin Zamri[1], Alyaa Alway[1], Mohd. Asyraf Mansor[1], Mohd Shareduwan Mohd Kasihmuddin[2]* and Saratha Sathasivam[2]

[1]*School of Distance Education, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia*
[2]*School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia*

ABSTRACT

Artificial neural networks (ANNs) are actively utilized by researchers due to their extensive capability during the training process of the networks. The intricate training stages of many ANNs provide a powerful mechanism in solving various optimization or classification tasks. The integration of an ANN with a robust training algorithm is the supreme model to outperform the existing framework. Therefore, this work presented the inclusion of three satisfiability Boolean logic in the Hopfield neural network (HNN) with a sturdy evolutionary algorithm inspired by the Imperialist Competitive Algorithm (ICA). In general, ICA stands out from other metaheuristics as it is inspired by the policy of extending the power and rule of a government/country beyond its own borders. Existing models that incorporate standalone HNN are projected as non-versatile frameworks as it fundamentally employs random search in its training stage. The main purpose of this work was to conduct a comprehensive comparison of the proposed model by using two real data sets with an elementary HNN with exhaustive search (ES) versus a HNN with a standard evolutionary algorithm, namely- the genetic algorithm (GA). The performance evaluation of the proposed model was analyzed by computing plausible errors, such as root mean square error (RMSE), mean absolute error (MAE),

ARTICLE INFO

Article history:
Received: 1 January 2020
Accepted: 18 March 2020
Published: 16 July 2020

E-mail addresses:
ezlinzamri@student.usm.my (Nur Ezlin Zamri)
alyaalway@student.usm.my (Alyaa Alway)
asyrafman@usm.my (Mohd. Asyraf Mansor)
shareduwan@usm.my (Mohd Shareduwan Mohd Kasihmuddin)
saratha@usm.my (Saratha Sathasivam)
*Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

global minima ratio (R_m), computational time (CT) and accuracy (Q). The computational simulations were carried out by operating the different numbers of neurons in order to validate the efficiency of the proposed model in the training stage. Based on the simulations, the proposed model was found to execute the best performance in terms of attaining small errors and efficient computational time compared to other existing models.

Keywords: 3-satisfiability, Hopfield neural network, imperialist competitive algorithm, logic mining

INTRODUCTION

The inception of artificial neural networks (ANNs) has initiated a variety of capable models, which act as a useful tool in solving specific tasks such as classification, prediction, and pattern recognition (Ghaleini et al., 2019). Many of the recent developments have assembled different takes in refining the existing ANN models, specifically by integrating them with proficient searching techniques in order to intensify the quality of their standalone framework. In general, ANN possesses comprehensive structure of training and testing stages, thus emerging as one of the most efficient tools in finding patterns and extracting information to solve real-life applications. They are implemented in tasks such as solar radiation forecasting (Benali et al., 2019), risk analysis (Shi et al., 2019), fault detection (Dybkowski & Klimkowski, 2019), and quantitative analysis (Li et al., 2019a). Accordingly, ANNs can be described in many forms; one of them is the feedback-inducing recurrent networks. In particular, Hopfield neural network (HNN) is a recurrent neural network resembling the operations of human memory (Hureira & Vartanian, 2019). Proposed by Hopfield and Tank in 1985, its ability to manage nonlinear patterns by its training and testing capabilities is especially useful for interpreting complex real-life problems. In recent years, HNN has been widely used by many researchers as it has a deliberately sturdy component of content addressable memory (CAM) (Kong et al., 2019) and emits a degree of convergence by utilizing an energy function (Kasihmuddin et al., 2019). However, the fundamental HNN employs dated heuristics in its training stage, namely the exhaustive search (ES). Nievergelt (2000) had discovered that ES was not considered as a robust search technique as it exerted a random search mechanism, which increased the tendency of overfitting and showed the lack of variations (Lim & Bang, 2010). As such, Mansor et al. (2019) had proposed the incorporation of Elliot Hopfield Neural Network (EHNN) with a modified artificial immune system algorithm (AIS), thus ameliorating the performance of the HNN elementary model. Other than that, Genitha and Vani (2019) had proposed an integrated framework of modified genetic algorithm (GA) with the HNN approach for a super-resolution mapping of satellite images, thereby instigating greater accuracy compared to a primary HNN model. GA is a computational processing algorithm inspired by Darwin's model, namely a survival for the fittest model (Feng et al., 2019). Furthermore, Jayashree and Kumar (2019) had underlined mutation and crossover as the key traits of GA in order to extract information and prioritize feature selection. Consequently, it is one of the prevalent metaheuristics used by many neural networkers, substantiating its compatibility for a comparison with the Imperialist Competitive Algorithm (ICA) mechanism. The common ground in these works is the homogenization of the HNN framework with other evolutionary algorithms to enhance the HNN training stage in producing a better HNN mechanism. In order to introduce an all-rounded model, HNN can implement more vigorous metaheuristics in its training stage.

Generally, ICA is an evolutionary algorithm motivated by human socio-political behaviors. Imperialism is known as the practice of a government/country to grow stronger and rule beyond its territory, whereby the imperialist's main vision is to increase the number of colonies. The main components of ICA consist of the initial empires, assimilation, revolution, and imperialist competition (Li et al., 2019b). A work by Tashayo et al. (2019) had inaugurated its use to forecast the maximum surface settlement (MMS) induced by tunneling in civil projects. Meanwhile, Gerist and Maheri (2019) had proposed an approach to solve damage detection problems, specifically by utilizing ICA and resulting in a great performance of the convergence rate and better identification of the global optima both. Therefore, it can be concluded that employing ICA is an endless potential, ranging from industrial planning, scheduling, and decision-making to machine learning (Atashpaz-Gargari & Lucas, 2007). However, it is commonly used by researchers to explicitly acquire a solution to a problem, rather than making use of it to generate a learning model for the problem. A research by Abdechiri and Meybodi (2011) had emphasized the credibility of HNN in utilizing ICA to solve the propositional satisfiability (SAT) problem. However, the work is not suitable for solving real data sets. Therefore, in the current work, the proposed model employs ICA in the training stage to overcome the complexity of checking clause satisfaction, generate variation, and vast searching space in order to solve two real-life data sets acquired from UCI repository.

Due to the comprehensibility of the HNN framework, researchers have considered it as a black box model or a symbolic system. Taking this fact into consideration, the execution of logic learning in HNN has delineated many versatile models, primarily from the work of Abdullah (1992) that incorporates logic programming on HNN. The work presented an extensive HNN framework to cater for the optimization of logical consistency. Abdullah (1992) had accordingly proposed an optimized logic learning through synaptic weights, which was called the Abdullah (WA) method. In layman terms, logic programming illustrates the symbolic knowledge that will be "trained" by the HNN model. The primary work by Sathasivam (2012) had further fostered Abdullah's (1992) proposal by implementing first-order logic in a neuro-symbolic integration model, which attained more than 90% of global minimum energy (Kasihmuddin et al., 2019). Several compelling logical rules such as the 2-Satisfiability (2-SAT) (Kasihmuddin et al., 2017a) and Maximum Satisfiability (MAX-*k*SAT) (Mansor et al., 2017) have been successfully embedded in HNN. Its application with propositional satisfiability logic is boundless, ranging from Very Large Scale Integration (VLSI) circuit configuration (Mansor et al., 2016) and Bezier curves satisfiability model (Kasihmuddin et al., 2016). For this work, the incorporation of the 3-Satisfiability (3-SAT) propositional logic is utilized due to its ability to achieve a higher probability of satisfied interpretation compared to Horn Satisfiability (Horn-SAT) and 2-SAT. Thus, the proposed HNN-ICA model is incorporated with 3-SAT in order to solve real-life applications.

Currently, no recent approach is available to thoroughly compare the performance of ICA with other metaheuristics in solving real-life data sets. This is crucial; as an evolutionary algorithm catering for variation and larger searching space in comparison with random search, ICA has to verify its distinctive features that can lead a better training model compared to other metaheuristics. Therefore, the contributions of this research are presented as follows: (1) to introduce the formulation of ICA with 3-SAT logic programming, (2) to initiate a model with the integration of HNN with ICA as a robust tool in order to solve optimization tasks by comparing it with two other searching techniques (i.e., GA and ES), (3) to implement reverse analysis with the proposed model of HNN-3SATICA in order to cater to real-life applications. The construction of the proposed model HNN-3SATICA shows better performance in the training stage and successfully interprets real-life datasets to detect the factors that are more prominent than others contributing to the optimization problems.

MATERIALS AND METHOD

3-Satisfiability Logic (3-SAT)

Propositional satisfiability or SAT logic is perceived as a logical rule that consists of clauses containing literals or variables. General satisfiability logic (*k*-SAT) can signify the capability to represent real-life applications (Kasihmuddin et al., 2019). This work utilized discrete HNN which catered neurons in bipolar representation $\{1,-1\}$ (Kasihmuddin & Sathasivam, 2016). Hansen et al. (2019) had emphasized on the generalization of *k*-SAT logical rule, P_{k-SAT} can be reduced to 3-SAT, P_{3-SAT} logical rule. The general formula of P_{3-SAT} is given as in Equation (1):

$$P_{3-SAT} = \wedge_{i=1}^{m} Z_i \tag{1}$$

where P_{3-SAT} is a 3-SAT that consists of clause Z_i shown in Equation (2):

$$Z_i = \vee_{j=1}^{3} \left(x_{ij}, y_{ij}, z_{ij} \right) \tag{2}$$

whereby n literals and m clauses denoted by Conjunctive Normal Form (CNF) formula. The general structure of P_{3-SAT} (Mansor et al., 2017) can be summarized as follows:

i. A set of m clauses in a Boolean formula, where $Z_1, Z_2, ..., Z_m$ and clauses will be connected with logical AND operator $\wedge$.
ii. Each clause consists of only literals will be combined by logical OR operator $\vee$. In the P_{3-SAT} formula, we only considered three literals in each clause.
iii. Boolean Satisfiability formula composes an array of n literals, $u_1, u_2, ..., u_n$, where $u_i \in \{1, -1\}$ in each clause. Note that in this work, n is equal to 3.
iv. The literals can be the variable itself or the negation of the variable, for example A or $\neg A$.

Further extension of P_{3-SAT}, an example of P_{3-SAT} is shown as follows:

$$P_{3-SAT} = (A \vee \neg B \vee C) \wedge (D \vee E \vee F) \wedge (\neg G \vee H \vee \neg I) \tag{3}$$

Equation (3) is satisfiable since it gives truth value resulting to $P_{3\text{-}SAT}$=1. According to Equation (3), if the neuron states read $(A, B, C, D, E, F, G, H, I) = (-1,1,1,1,1,1,1,-1,1)$, the formula will be unsatisfiable or $P_{3-SAT} = -1$. In this research, P_{3-SAT} will be embedded to the proposed model, HNN-3SATICA in comparison with different learning algorithms. P_{3-SAT} will cater the modified networks to unveil the true pattern or behaviour of the real data sets involved. Note that P_{3-SAT} is a symbolic form representation thus it is appropriate to be integrated in these networks as HNN is a non-symbolic platform.

Hopfield Neural Network (HNN)

HNN is a recurrent neural network, without hidden layer that mimics human biological brain. HNN structure of interconnected neurons and a powerful feature of CAM are crucial in solving various optimization and combinatorial tasks (Kong et al., 2019). The proposed model consists of structured N neurons, each of which is represented by an Ising variable. The neurons in discrete HNN are utilized in bipolar representation whereby $S_i \in \{1, -1\}$ (Sathasivam, 2010). The fundamental overview for the bipolar neuron state activation in HNN is shown in Equation (4):

$$S_i = \begin{cases} 1 & , \ if \ \sum_j W_{ij} S_j \geq \omega \\ -1 & , \text{otherwise} \end{cases} \tag{4}$$

where W_{ij} refers to synaptic weight of the neuron from unit j to i. S_j is the state of neuron j and ω is the predefined threshold value. Barra et al. (2018) specified that $\omega = 0$ to verify the network's energy decreases and ascertain our network to achieve plausible results. The connection in Hopfield net contains no connection with itself $W_{jj} = W_{ii} = 0$ (symmetrical). HNN model has similar intricate details to the Ising model of magnetism (Neelakanta & DeGroff, 1994). As the neuron states are termed in bipolar form, the neuron rotates towards the magnetic field, resulting in the neurons to rotate until the equilibrium is achieved. Hence, the dynamic of HNN (considering all the neurons involved) is asynchronously changed according to $S_i \rightarrow \text{sgn}\left[h_i(t)\right]$, where h_i is the local field of the neurons connection. Motivated by Sathasivam et al. (2011), the sum of the field induced by each neuron is given in Equation (5):

$$h_i = \sum_{k=1,k\neq j}^{N} \sum_{j=1,j\neq k}^{N} W_{ijk} S_j S_k + \sum_{j=1,j\neq i}^{N} W_{ij} S_j + W_i \tag{5}$$

Thus, the biggest task of local field is to evaluate the final bipolar state of neurons and generate all possible P_{3-SAT} induced logic that was obtained from the final state of neurons. One of the most prominent features of HNN is the fact that it always converges in some cases, as illustrated by the following theorem (Hopfield, 1982).

Theorem 1. Let N be a neural network of order n and be defined by $N = (W, T)$ where W is an $n \times n$ matrix with element W_{ij} and T is a vector of dimension n, where element t depicted as the threshold attached to node i. The network will always converge to a stable state when running in serial mode; only one neuron can change the state at any time instantly, if the diagonal elements of W are non-negative.

Moreover, the subsequent state updating rule can be represented as in Equation (6)

$$S_i(t+1) = \text{sgn}\left[h_i(t)\right] \tag{6}$$

whereby sgn represents the signum function to squash the output of neurons, where this paper utilized Hyperbolic Tangent Activation Function (HTAF) (Mansor & Sathasivam, 2016). The following Equation (7) represents the Lyapunov energy function in HNN (Mansor et al., 2018a).

$$L_{P_{3-SAT}} = -\frac{1}{3} \sum_{i=1,i\neq j\neq k}^{N} \sum_{j=1,i\neq j\neq k}^{N} \sum_{k=1,i\neq j\neq k}^{N} W_{ijk} S_i S_j S_k - \frac{1}{2} \sum_{i=1,i\neq j}^{N} \sum_{j=1,i\neq j}^{N} W_{ij} S_i S_j - \sum_{i=1}^{N} W_i S_i \tag{7}$$

The energy value computed from the Equation (7) will be authenticated as global or local minimum energy. The network will provide the filtering mechanism and produce the correct solution when the induced neurons state reached global minimum energy. There are limited works to combine HNN and ICA as a single computational network. Thus, the robustness of ICA helps to improve training process in HNN.

Imperialist Competitive Algorithm (ICA)

The pioneering work of ICA was presented by Atashpaz-Gargari and Lucas (2007), who stated that it was an algorithm inspired by imperialistic competition. Generally, all inspired countries are divided into two parts, namely the imperialist states and colonies, which tackle a list of operations such as initialization, assimilation, revolution, and imperialist competition. This can lead to a better searching technique compared to other metaheuristics. The main purpose of ICA is to drive the colonies to converge to a global minimum solution, which is believed to have shown vigorous mechanisms in solving optimization tasks (Mollajan et al., 2019). It is set to be different compared to other metaheuristics as its features function to ease the performing neighborhood movements in both the continuous and discrete search spaces (Hosseini & Al Khaled, 2014). The application of ICA is infinitely many, such as ship design optimization (Peri, 2019), engineering design optimization (Aliniya & Keyvanpour, 2019), slope stability prediction (Koopialipoor et al., 2019), and heat and power dispatch problem (Davoodi & Babaei, 2019).

This work is focusing on utilizing ICA to find the maximum fitness of countries that will increase the number of satisfied clauses in the training stage. Its implementation with HNN is addressed as follows (Abdechiri & Meybodi, 2011):

Step 1. Forming initial empires (Initialization)
Each solution is shown by an array called country. Note that, in an N-dimensional optimization tasks, a country is denoted as $1 \times N$ array. This array predefined as Equation (8):

$$C_{Nj} = \begin{cases} 1 & , \ rand\left[0,1\right] \geq 0.5 \\ -1 & , \text{otherwise} \end{cases} \tag{8}$$

whereby C_i is the country and C_{Nj} is the number of variables to be considered of interest about a country. Each empire E_i comprises N number of countries which represents the state of P_{3-SAT} as shown in Equation (9):

$$E_i = \left(C_1^{E_i}, C_2^{E_i}, ..., C_N^{E_i}\right) \tag{9}$$

Step 2. Fitness Evaluation
Each country's fitness is calculated based on the clauses by using Equation (10):

$$f_{C_i^{E^i}} = \sum_{i=1}^{NZ} Z_i \tag{10}$$

$f_{C_i^{E^i}}$ denoted as fitness of each C_i in E^i, Z_i is the clause in P_{3-SAT} and NZ is the total number of clauses.

Step 3. Colonies moving towards imperialist (Assimilation)
We select the imperialist, T^{E_i} from the country, $C_i^{E_i}$ with the highest $f_{C_i^{E^i}}$ and the rest will remain as colonies by using Equation (11).

$$C_i^{E_i} = \begin{cases} T^{E_i}, & \max\left[f_{C_i^{E^i}}\right] \\ C_i^{E_i}, & \text{otherwise}, \ i = rand\left[1, N\right] \end{cases} \tag{11}$$

Other remaining countries and colonies will be allocated to respective empires where each colonies population is randomized.

Step 4. Revolution

When revolution occurs, the rest of the colonies, $C_i^{E_i}$ inside an empire E_i will be randomized according to the following Equation (12):

$$C_{ij}^{E_i} = \begin{cases} 1 & , \ rand[0,1] \geq 0.5 \\ -1 & , \text{ otherwise} \end{cases} \tag{12}$$

to offer colonies acquiring better position that will attain higher chances of redeeming its place to take over the reigning empire by replacing the current imperialist, T^{E_i}. In order for revolution to take place, the fitness for each $C_i^{E_i}$ will be computed by using Equation (10) and the new imperialist T^{E_i} will be selected based on Equation (11).

Step 5. Imperialist Competition

This feature of ICA sets apart ICA from other metaheuristics. In this process, imperialistic competition occurs among all imperialist in order to acquire power of each empires $V_N^{E^i}$ as in Equation (13):

$$V_{E_i} = f_{T^{E_i}} - \varepsilon f_{T^{E_i}} + \frac{1}{N-1}\sum_{i=1}^{N} f_{C_i^{E_i}} \tag{13}$$

whereby $f_{T^{E_i}}$ represents the fitness of each imperialist. According to Atashpaz-Gargari & Lucas (2007), $\varepsilon = 0.05$ is chosen as an optimal value for this study. Empires that show no power will fall in this competition and the surviving imperialist will be the one that has the highest power. Worth mentioning that, if the power for a particular empire is within $\left|V_{E_i} - V_\phi\right| \leq \lambda$, T^{E_i} will be chosen as the final neurons states for P_{3-SAT}. Note that λ will be predetermined by users. Step 4 and Step 5 will be repeated until termination criteria $f_{T^{E_i}} = NZ$ is been met. The state of T^{E_i} will be stored as CAM. In this paper, we modified the ICA of the pioneering work (Atashpaz-Gargari & Lucas, 2007) that utilized ICA to solve a continuous problem into bipolar representation to solve NP problems. A work on implementing ICA with HNN for solving Satisfiability problem was executed by Abdechiri & Meybodi (2011), however it only catered to simulated data set. Figure 1 depicts the summary of ICA from Step 1 until Step 5. This research extends the work of ICA by implementing HNN and P_{3-SAT} logical rule with reverse analysis in solving real data sets.

Genetic Algorithm (GA)

In this paper, GA will be applied into HNN-3SAT or abbreviated as HNN-3SATGA in order to compare with other searching techniques, HNN-3SATICA and HNN-3SATES. GA is a popular optimization algorithm that was inspired by Darwin's evolutionary theory to find a formula or optimized answer in order to predict or match patterns (Esfe et al., 2019).

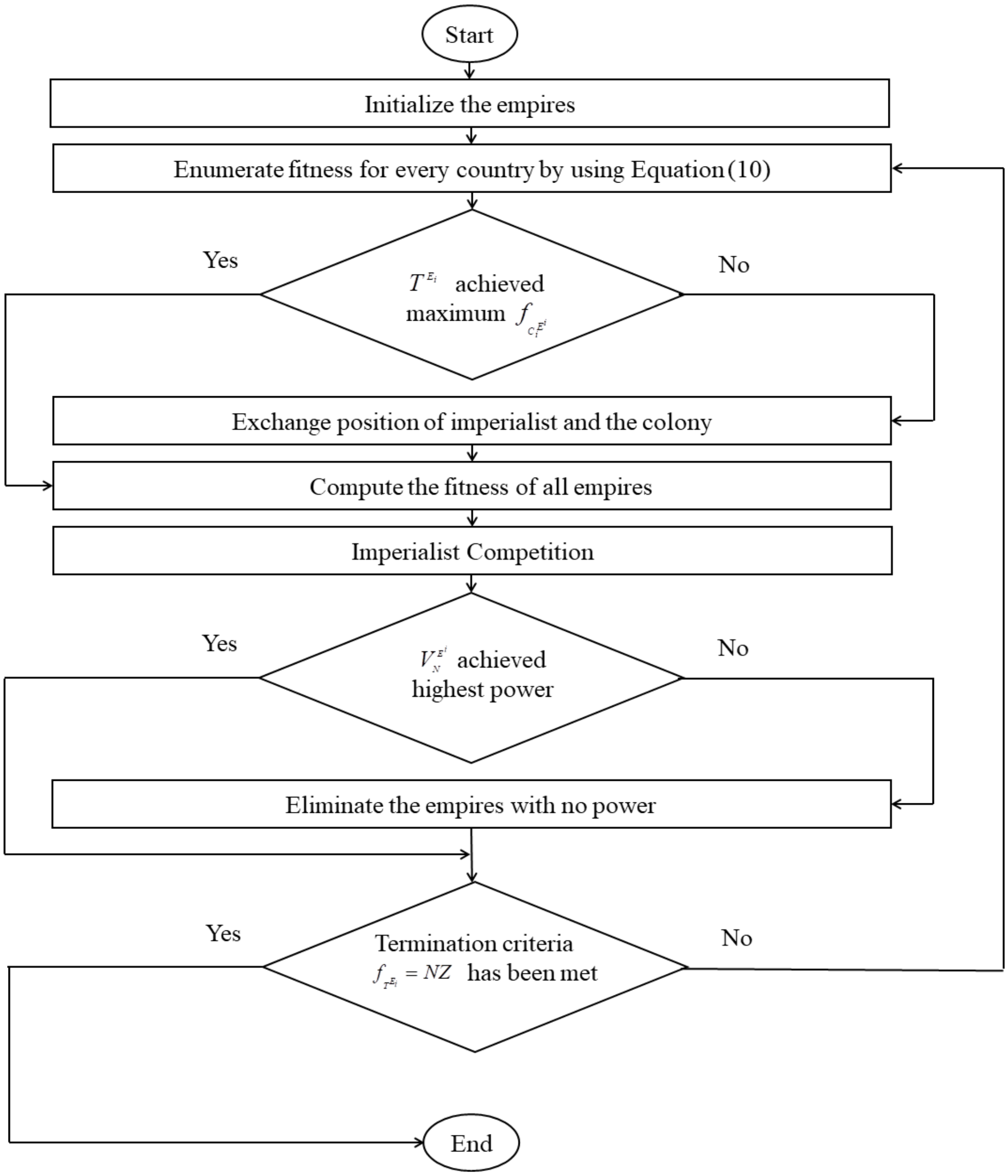

Figure 1. Summary of ICA

The phases of HNN-3SATGA are as follows (Mansor et al., 2019):

Step 1. Initialization
Initialize 100 random chromosomes (Cr_i) as bipolar interpretation where each element of $\{1,-1\}$ is denoted by True and False. Every Cr_i portrays the possible interpretation for P_{3-SAT}.

Step 2. Cr_i Fitness Computation
Fitness of the Cr_i in GA is typically being calculated by using fitness function. Thus, the fitness computation is based on the satisfiability of the clauses in P_{3-SAT}. Hence, the maximum fitness manifests the effectiveness of the training process.

Step 3. Cr_i Selection

Select y Cr_i (in our case $y=10$) containing P_{3-SAT} information with the highest fitness out of 100 Cr_i to undergo the crossover process.

Step 4. Crossover
Two Cr_i are selected and separated during the crossover phase from 10 selected Cr_i. Furthermore, by joining both parts of the paternal Cr_i, a child Cr_i is produced. Hence, the second child is therefore generated by the addition of the primary and secondary segment and vice versa (Luke, 2013).

For example:
Before crossover

$$Cr_1 = -1-1-1111-1-1-1$$

$$Cr_2 = -1-1-11-11-1-1-1$$

After crossover

$$Cr_1 = -1-1-1-1-1-1-1-1-1$$

$$Cr_2 = -1-1-11-11111$$

Therefore, the Cr_i fitness for the new generated Cr_i can be determined.

Step 5. Mutation
Mutation is an integral optimization operator in the GA which shifts Cr_i patterns, ensuring that the population is not trapped at local minima. This is the random process of altering specific genes of the Cr_i. As an illustration:

Before mutation

$$Cr_3 = -1-1-11-11111$$

After mutation

$$Cr_3 = 1-1-11-11111$$

The first part of the Cr_i was flipped from -1 to 1. Thus, a better Cr_i might be generated after mutation. The fitness value for a newly formed Cr_i will be computed. The current Cr_i will repeat the first step if the fitness value does not achieve maximum fitness.

IMPLEMENTATION

Performance Evaluation Metrics

The performance of the proposed model in executing logic mining will implement several performance metrics such as root mean square error (RMSE), mean absolute error (MAE), global minima ratio (R_m), computational time (CT) and accuracy (Q).The list of parameter names in performance evaluation metrics is shown in Table 1. Meanwhile, the list of parameter values used in R_m depicts in Table 2.

Root Mean Square Error (RMSE)

Overall, RMSE is a prediction metric utilized by neural networkers to enumerate the predicted value of a model with observed value. RMSE formulation utilized in this work is shown in Equation (14):

$$RMSE = \sum_{i=1}^{n} \sqrt{\frac{1}{n}(f_x - f_y)^2} \tag{14}$$

Mean Absolute Error (MAE)

According to Willmott and Matsuura (2005), MAE capable of evaluating good error estimation by showing a uniformly distributed error. MAE formulation consists of the absolute value of the difference between the estimated values and the actual values (Chai & Draxler, 2014). A good model will attain low values of RMSE and MAE. The equation of MAE is shown in Equation (15):

$$MAE = \sum_{i=1}^{n} \frac{1}{n}\left|f_x - f_y\right| \tag{15}$$

Global Minima Ratio (R_m)

The global minimum ratio is the generalized metric for evaluating the efficiency of the solutions. Tolerance value will filter each calculated final energy of the neurons in HNN. The final energy is assumed as global minimum energy if the final energy of the model within the tolerance value (Sathasivam, 2010). The Equation (16) of R_m is shown as follows:

$$R_m = \frac{1}{ab}\sum_{i}^{n} N_{L_{P_{3-SAT}}} \tag{16}$$

Computational Time (CT)

Computational time is used to determine the effectiveness of the proposed models. The value for CT will be measured in SI unit of second (s). The Equation (17) of CT is shown below:

$$CT = TrainingTime(s) + TestingTime(s) \tag{17}$$

whereby $TrainingTime$ and $TestingTime$ are depicted as the total time to execute the HNN-3SAT models in training and testing phase respectively. In the work by Kho et al. (2020), CT was utilized for HNN-2SAT model because it implies the capability and stability of the model.

Accuracy (Q)

The accuracy is used to assess the models ability to train the data set. The Equation (18) of Q is shown below:

$$Q = \frac{P_{induced}^{Correct}}{N_{P_{test}}} \times 100\% \tag{18}$$

Table 1

List of parameters in performance evaluation metrics

Parameter	Parameter name
f_x	Total number of clauses
f_y	Number of satisfied clause
n	Number of iteration before $f_x = f_y$
a	Number of trials
b	Number of neuron combination

Parameter	Parameter name
$N_{L_{P_{3-SAT}}}$	Number of global minimum energy
$N_{L_{P_{3-SAT}}}$	Number of global minimum energy
$N_{P_{test}}$	Number of testing data
$P_{induced}^{Correct}$	Correct induced logic

Table 2

List of parameters in R_m

Parameter	Parameter value
a	100
b	100
Tolerance value	0.001 (Sathasivam, 2010)

Experimental Setup

The utilization of P_{3-SAT} will aid in discovering valuable information on real data sets. Many works of literature have surfaced in explaining that information and patterns can be represented in a logical form. The use of reverse analysis method to extract significant information from a particular data set has been introduced by Sathasivam and Abdullah (2011) by considering the CNF logical rule. Motivated by the work of Mansor et al. (2018b), more systematic logic mining techniques incorporating P_{3-SAT} in HNN have been proposed. Following this, the 3-SAT-based Reverse Analysis Method (3-SATRA) is employed in HNN-3SATICA to generate an optimized induced logical rule from several prominent data sets. In this case, raw data are translated into P_{3-SAT} and then embedded and processed by HNN-3SAT. By pursuing this, the induced P_{3-SAT} will be used to classify the outcome of the dataset. The comprehensive 3-SATRA via HNN-3SATICA is depicted in Figure 2.

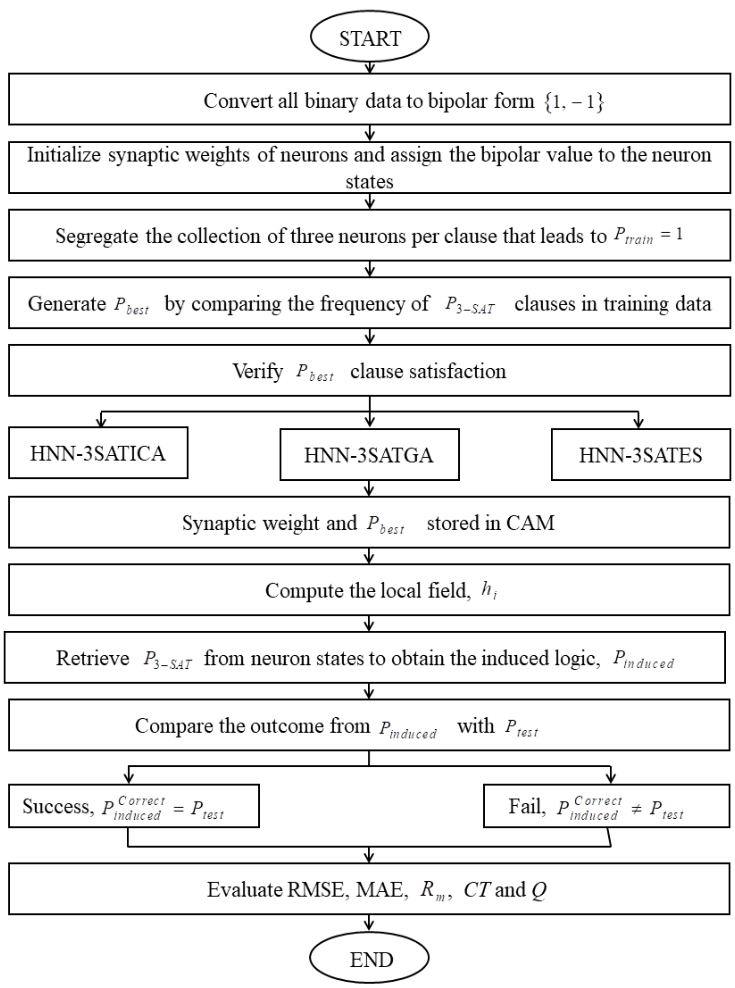

Figure 2. Implementation of 3-SATRA in HNN-3SATICA

In this paper, two different real data sets from different fields occupied the constructed 3-SATRA, namely the Bach Choral Harmony data set (BCHDS) and German Credit data set (GCDS). All data sets were taken from the UCI machine learning repository website and each of them had different purposes. Nine attributes were used in this paper and all real data sets used multivariate data. In this experiment, the aim was to analyze a comprehensive comparison of accuracy between HNN-3SATICA and other existing models such as GA, ES, and researchers utilizing the same data sets. The information about BCHDS and GCDS is shown in Table 3 and 4.

Table 3
List of attributes for each data sets

Data Set	Details of each attributes			Output $P_{3\text{-}SAT}$
BCHDS (Haque et al., 2019)	A	:	Pitch for Key "C-M"	To identify the distinction between good harmony and bad harmony for musician.
	B	:	Pitch for Key "E-m"	
	C	:	Pitch for Key "B-m"	
	D	:	Pitch for Key "A-m"	
	E	:	Pitch for Key "F-m"	
	F	:	Pitch for Key "G-M"	
	G	:	Pitch for Key "A-M"	
	H	:	Bass	
	I	:	Meter	
GCDS (Liu et al., 2019)	A	:	Credit History	To distinguish bank's customer of having a good or bad credit risks.
	B	:	Status of existing checking account	
	C	:	Saving account/bonds	
	D	:	Personal status	
	E	:	Age	
	F	:	Housing	
	G	:	Number of existing credits at this bank	
	H	:	Telephone	
	I	:	Foreign worker	

As mentioned earlier, the literals can be the variable itself or the negation of the variable; for example, A or $\neg A$. Thus, as the work implemented real-life data sets, the negation literals represented attributes that did not affect the output of P_{3-SAT} and *vice versa* for non-negate literals.

Table 4
Method used in existing model

Data Set	Method
BCHDS	AdaBoost
GCDS	Support Machine Vector (SVM)

The source code was developed through Dev C++ Version 5.11 in 8GB RAM with Intel Core i5 to acquire a good comparison between all proposed models. The same device was used during simulation to avoid bias; using the same device rendered the simulation comparable as the memory (RAM) and processor had the same power. Another factor that can trigger bias is a different compiler for each learning algorithm, which may lead to different computational times or anomaly in error values. Moreover, to avoid bias, the number of neurons chosen for the simulations showed the same combination of $9 \leq NN \leq 72$ for all learning algorithms. The threshold for simulation was set to 24 hours according to Kasihmuddin et al. (2016). All outputs exceeding 24 hours of computation time were omitted due to the simulation, which would eventually break down in finding the satisfied clause interpretation, $E_{P_{3-SAT}} = 0$. Table 5, 6 and 7 show all parameters involved in the proposed models.

Table 5
List of parameters in HNN-3SATES (Sathasivam, 2012)

Parameter	Parameter value
Neuron Combination (*b*)	100
Number of Trial (*a*)	100
Tolerance Value (*Tol*)	0.001
Number of String	100
Selection Rate	0.1

Table 6
List of parameters in HNN-3SATGA (Kasihmuddin et al., 2017b)

Parameter	Parameter value
Neuron Combination (*b*)	100
Number of Trial (*a*)	100
Tolerance Value (*Tol*)	0.001
Selection Rate	0.1
Mutation Rate	0.01
Generation	1000

Table 7
List of parameters in HNN-3SATICA

Parameter	Parameter value
Neuron Combination (b)	100
Number of Trial (a)	100
Tolerance Value (Tol)	0.001
Initial Empires	10
Parameter (ε)	0.05
Termination Value (λ)	0.05

RESULT AND DISCUSSION

The effectiveness of implementing ICA in the training stage of the HNN-3SAT model was investigated in this paper. The proposed mode of HNN-3SATICA in comparison with existing models that utilized GA (Kasihmuddin et al., 2017b) and ES (Sathasivam, 2012) embedded two real-life data sets, which were retrieved from the UCI Machine Learning Repository platform. The investigation of a model's performance can be separated into two parts. The first important part is to examine the quality of the solution generated by different searching techniques, specifically by employing the suitable training errors. Second, one should analyse the robustness and efficiency of the proposed model by comparing the CT and Q needed to execute the models' respective mechanisms. Accordingly, five performance evaluation metrics were involved in analyzing the training and testing stages of the modified models, as presented in the Performance Evaluation section. Therefore, this research's main contribution was the display of HNN-3SATICA competency in outperforming the existing models.

In this section of result analysis for BCHDS, it could be concluded that the outcomes attained by HNN-3SATICA for RMSE and MAE showed a consistent value of zero (Figure 3 & 4), thereby indicating ICA providing a better and well-trained HNN framework. However, the results were identical to HNN-3SATGA. Similarly, HNN-3SATES projected larger errors as the number of neurons increased. The incorporation of ES underlined the lack of modification undertaken in the training stage of the fundamental HNN. For the R_m, it is likely to be a better model when it is prone to the value of 1 (Sathasivam, 2011). From Figure 5, all models are approaching to 1 even by manipulating different numbers of neurons. The trend showed the capability of the training methods deployed by HNN-3SAT to attain the global minimum energy by having R_m closer to 1. From Figure 6, it can be deduced that all models exhibit less CT compared to the assigned threshold time. Regardless, HNN-3SATICA executed less CT compared to other models, thereby displaying

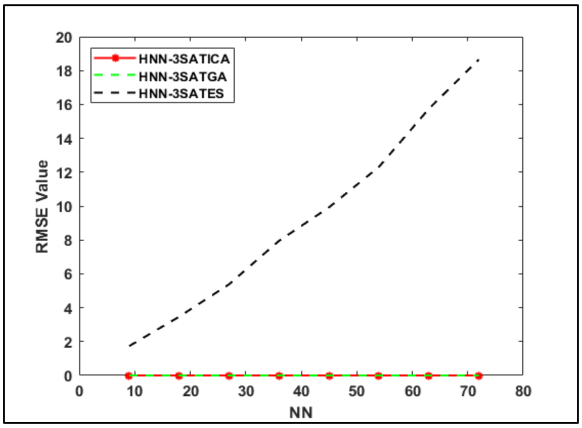

Figure 3. RMSE value of HNN-3SAT models for BCHDS

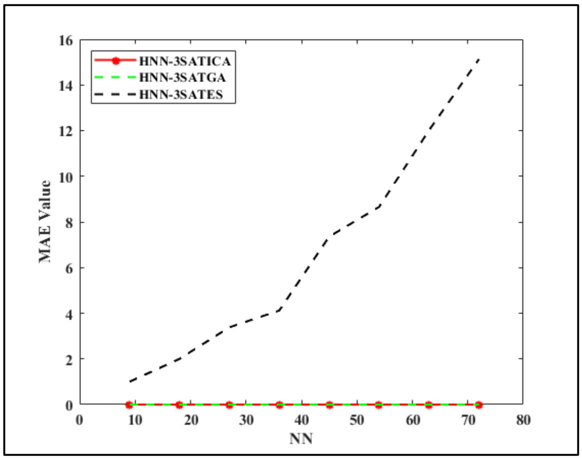

Figure 4. MAE value of HNN-3SAT models for BCHDS

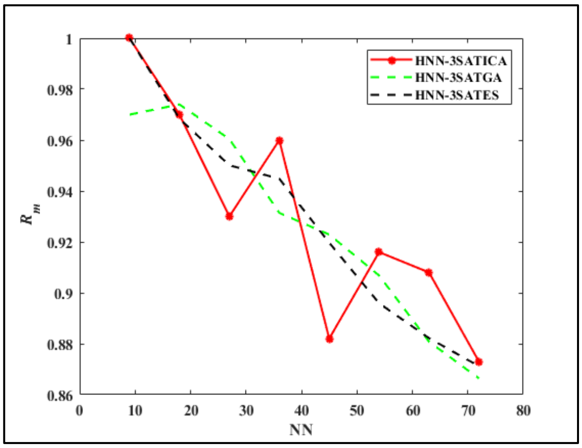

Figure 5. R_m of HNN-3SAT models for BCHDS

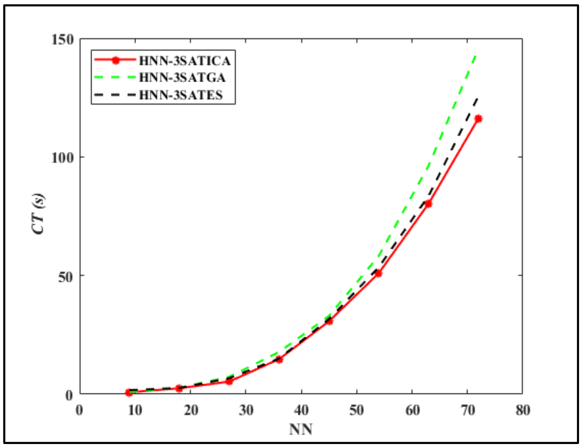

Figure 6. CT of HNN-3SAT models for BCHDS

Table 8
Accuracy of HNN-3SATICA with other existing models

Data Set	HNN-3SATICA	Sathasivam (2012)	Kasihmuddin et al. (2017b)	Haque et al. (2019)
BCHDS	61%	54%	61%	56.72%

the effectiveness of ICA during the training stage with different levels of complexity. Table 8 outlines the accuracy of all models that utilized BCHDS, whereby HNN-3SATICA and HNN-3SATGA are observed to enumerate the same value of accuracy. Both models achieved 61% of accuracy, which demonstrated their respective capability in attaining an optimized induced logic for BCHDS. Besides, HNN-3SATES and the work by Haque et al. (2019) both generate an accuracy value less than 60%. The accuracy for both methods was not promising, particularly for HNN-3SATES. It is due to the non-optimized induced logic generated at the end of the executions. By executing the simulation of HNN-3SATICA for BCHDS, the induced logic attained by HNN-3SATICA can be observed as per Equation (19) below:

$$P_{induced}^{Correct} = (\neg A \vee B \vee \neg C) \wedge (D \vee E \vee \neg F) \wedge (G \vee H \vee I) \tag{19}$$

whereby from Equation (19), in order to produce a good harmony, attributes A, C, and F are insignificant to be scrutinize. However, such attributes like B, D, E, G, H and I would lead the harmonization to be out of tune.

In comparison with the result analysis of BCHDS, GCDS showed better result of HNN-3SATICA with other existing models. Figure 7 reveals the RMSE values attained by HNN-3SATGA and HNN-3SATES as relatively increasing as the number of neurons increased. However, HNN-3SATICA persistently achieved zero RMSE values despite the incremental number of neurons. Furthermore, the MAE analysis for all models yielded comparatively similar outcomes to the illustration of RMSE analysis. From Figures 7 and 8, the execution of the ICA mechanism is significant to aid the standalone HNN with a vigorous training stage to achieve $E_{P_{3-SAT}} = 0$. Unfortunately, both HNN-3SATGA and HNN-3SATES revealed an inclining trend of errors for their MAE and RMSE outcomes both, showing the incompetency of the GA and ES mechanisms to accommodate a higher number of neurons as the complexity increased. The analysis of R_m in GCDS (Figure 9) shows that most of the models are able to generate at most 100% of global minimum solutions, except for HNN-3SATGA. Although two different data sets from different fields were utilized, the findings of R_m achieved by all models depicted indistinguishable results. Figure 10 displays that the *CT* for HNN-3SATGA and HNN-3SATES requires more time compared to HNN-3SATICA. This was due to the proposed model offering a larger search space, which contributed to a more efficient HNN framework. In Table 9, the induced logic generated by HNN-3SATICA in the testing stage records an accuracy of 83%. This finding set forth its ability to acquire an optimized induced logic that could best represent the GCDS data set. Contrary to this, HNN-3SATGA, HNN-3SATES, and SVM methods deployed by Liu et al. (2019) have achieved lower accuracy compared to HNN-3SATICA. This is attributable to ICA mechanism's function to attain $E_{P_{3-SAT}} = 0$, which can lead to a better training stage in the resulting construction of an optimized induced logic. Equation (20) displays the induced logic attained by HNN-3SATICA at $NN = 54$ until $NN = 72$:

$$P_{induced}^{Correct} = \left(\neg A \vee B \vee \neg C\right) \wedge \left(D \vee E \vee \neg F\right) \wedge \left(G \vee H \vee I\right) \tag{20}$$

From Equation (20), one can distinguish whether a customer is a good credit risk or not, whereby attributes such as *A* and *D* can exhibit a fair credit status. Here, the credit history and personal status were very important to know their credit management and liability. Other than that, the induced logic could reveal insignificant and trivial attributes such as *C* and *I* in order to sort out which customer was a good or bad credit risk.

The training stage played a prominent role in enhancing the standalone HNN framework. From the HNN-3SATICA results obtained in Figure 3 and 4 and Figure 7 and 8, the capability of ICA is portrayed by improving the HNN training stage to attain a good solution. The deviation of error was generally smaller than the other counterparts due to the optimization operator employed in ICA. Assimilation and revolution in ICA played a big role to generate fewer iterations in obtaining $E_{P_{3-SAT}} = 0$. Apart from this, the fewer iterations indicated that less RMSE and MAE were generated from the model. However, the results attained for HNN-3SATGA and HNN-3SATES were larger than the proposed model, particularly in GCDS. They both employed undeniably ineffective searching method compared to ICA, especially ES. In particular, ES used random search where the number of neurons increases, the complexity also increases. Thus, ES contributes to

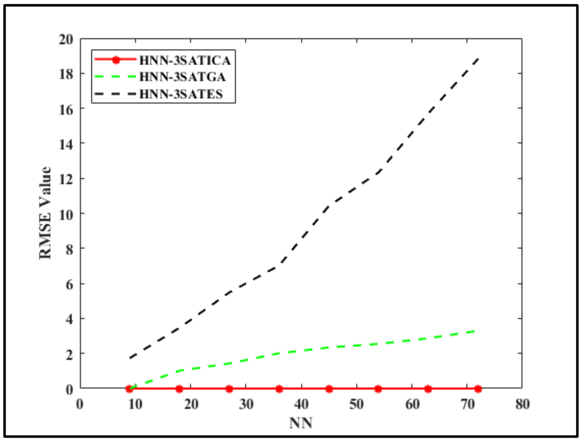

Figure 7. RMSE value of HNN-3SAT models for GCDS

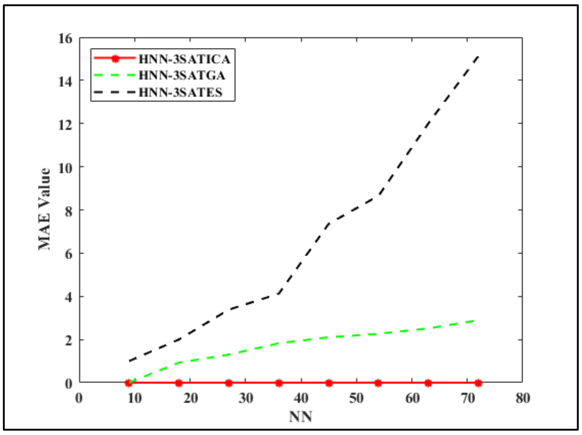

Figure 8. MAE value of HNN-3SAT models for GCDS

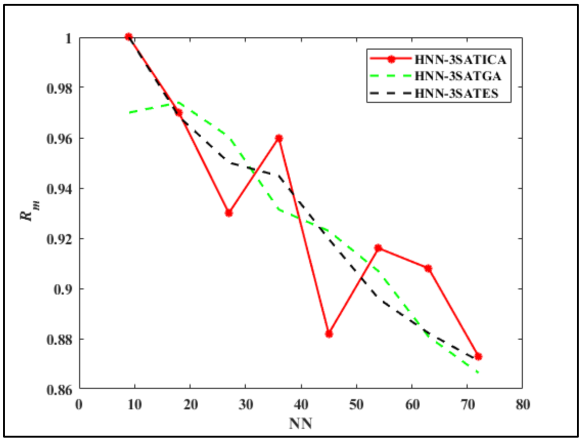

Figure 9. R_m of HNN-3SAT models for CGDS

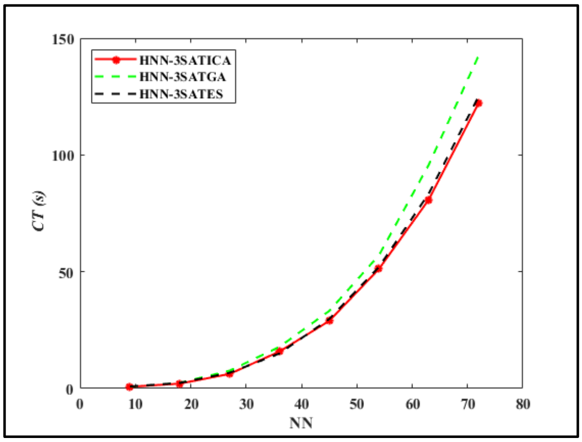

Figure 10. CT of HNN-3SAT models for BCHDS

Table 9

Accuracy of HNN-3SATICA with other existing models

Data Set	HNN-3SATICA	Sathasivam (2012)	Kasihmuddin et al. (2017b)	Liu et al. (2019)
GCDS	83%	71%	82%	75.7%

generate larger errors. Figures 5 and 9 showcase the R_m obtained by all models, which are relatively identical even as the number of neurons increased. Therefore, this ensured the proposed model reached the global minimum energy without any complications. This fact displayed the versatility and relevancy of the ICA mechanism to accomplish close achievement as the existing models.

CT is predefined as the expanse of time needed for the network to complete the overall computational process. Therefore, it is significant to identify a model's performance efficiency affirmation (Xiao et al., 2017). From Figures 6 and 10, it is discovered that CT needed to execute HNN-3SATICA is lesser than other existing models. In contrast, with HNN-3SATES, HNN-3SATICA attained faster execution due to the ICA algorithm that had fewer parameters compared to ES. However, for ES, the number of neurons was gradually increasing and thus more CT was required due to its nature of brute force that needed more iterations. Thereby resulting in overfitting the solution. HNN-3SATICA was set apart from other models as it achieved a satisfactory percentage of accuracy for both BCHDS and GSDS data sets. Based on the results in Table 9, it can be concluded that ICA plays a vital role in the proposed model to generate better-induced logic, which reflects the precision of the modified network. The reason for ICA being able to fulfill such factor is due to the optimization operator in it, which introduces a larger search space and variations of solution that contribute to an effective training stage.

The limitation of HNN-3SATICA is that this modified framework only caters to bipolar representation and only utilizes the multivariate type of data sets. Further extension can be done by utilizing the binary form of entries and incorporating other types of data sets, such as time-series. Other than that, this modified network only utilized one type of SAT, namely 3-SAT. Therefore, one can incorporate more than one or other types of SAT, such as 2-SAT, MAX-*k*SAT, and *k*-SAT. The main drawback of HNN-3SATICA is its tendency of overfitting, which lacks the variability of generated induced logic. In the current context, overfitting indicates that as the search algorithm becomes more complex in the training phase, the solution will produce inaccurate and biased results (Reunanen, 2003). To overcome such aspect, an alteration of the data sets is crucial. Rearrangement and permutation of the attributes with a randomized selection of attributes should be implemented. The no-free lunch theorem (Wolpert & Macready, 1997) states that there are no absolute or specific algorithms that can be utilized to solve any problems. However, in this research, it was found that HNN-3SATICA worked exceptionally well for GCDS. On the contrary, the HNN-3SATICA and HNN-3SATGA worked well for BCHDS.

CONCLUSION

In this research, the formulation of constructing ICA in 3-SAT proved to be adequate to represent the mechanism that involves in ICA and the implementation of ICA with standalone HNN framework with reverse analysis proved to be effective in solving real life data sets. Through this approach, we have successfully presented a modified model of HNN-3SATICA which presented ICA role in generating variations and broad searching space. Particularly in this research, assimilation and revolution components of ICA provided a better solution in checking $E_{P_{3-SAT}} = 0$ which modify the training stage of a fundamental HNN framework. The proposed HNN-3SATICA model was trained and tested by using two real life data sets, in comparison with two different searching techniques; HNN-3SATGA and HNN-3SATES. The performance evaluation between these models was analysed by employing different performance metrics such as RMSE, MAE, global minima ratio, computational time and accuracy. The analysis of the results displayed the competency of the proposed model, HNN-3SATICA for all chosen performance evaluation metrics. We have successfully accomplished all objectives as presented based on the performance of our proposed model. Computational simulations for all models were specifically difficult towards larger number of neurons. However, the constructed proposed model, HNN-3SATICA showed good potential and a better training mechanism compared other metaheuristics and regular HNN framework. Extended research is required to further enhance HNN-3SATICA model by employing the same mechanism to other types of recurrent neural networks, such as Elman and Kohonen neural network. Other than that, we can utilize other types of propositional logic such as *k*-SAT and MAX-*k*SAT.

ACKNOWLEDGEMENT

This research was funded by Fundamental Research Grant Scheme (FRGS), Ministry of Education Malaysia, grant number 203/PMATHS/6711804 and Universiti Sains Malaysia.

REFERENCES

Abdechiri, M., & Meybodi, M. R. (2011, February 26-28). A hybrid Hopfield network-imperialist competitive algorithm for solving the SAT problem. In *3rd International Conference on Signal Acquisition and Processing (ICSAP 2011)* (Vol. 2, pp. 37-41). Singapore.

Abdullah, W. A. T. W. (1992). Logic programming on a neural network. *International Journal of Intelligent Systems*, *7*(6), 513-519.

Aliniya, Z., & Keyvanpour, M. R. (2019). CB-ICA: A crossover-based imperialist competitive algorithm for large-scale problems and engineering design optimization. *Neural Computing and Applications*, *31*(11), 7549-7570.

Atashpaz-Gargari, E., & Lucas, C. (2007, September 25-28). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In *2007 IEEE Congress on Evolutionary Computation* (pp. 4661-4667). Singapore.

Barra, A., Beccaria, M., & Fachechi, A. (2018). A new mechanical approach to handle generalized Hopfield neural networks. *Neural Networks*, *106*, 205-222.

Benali, L., Notton, G., Fouilloy, A., Voyant, C., & Dizene, R. (2019). Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components. *Renewable Energy*, *132*, 871-884.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), 1247-1250.

Davoodi, E., & Babaei, E. (2019). A modified imperialist competitive algorithm for combined heat and power dispatch. *Computational Intelligence in Electrical Engineering, 10*(1), 1-18.

Dybkowski, M., & Klimkowski, K. (2019). Artificial neural network application for current sensors fault detection in the vector controlled induction motor drive. *Sensors*, *19*(3), 1-16.

Esfe, M. H., Amiri, M. K., & Bahiraei, M. (2019). Optimizing thermophysical properties of nanofluids using response surface methodology and particle swarm optimization in a non-dominated sorting genetic algorithm. *Journal of the Taiwan Institute of Chemical Engineers*, *103*, 7-19.

Feng, L., Ong, Y. S., & Gupta, A. (2019). Genetic algorithm and its advances in embracing memetics. In *Evolutionary and Swarm Intelligence Algorithms* (pp. 61-84). Cham, Switzerland: Springer.

Genitha, C. H., & Vani, K. (2019). A hybrid approach to super-resolution mapping of remotely sensed multi-spectral satellite images using genetic algorithm and Hopfield neural network. *Journal of the Indian Society of Remote Sensing*, *47*(4), 685-692.

Gerist, S., & Maheri, M. R. (2019). Structural damage detection using imperialist competitive algorithm and damage function. *Applied Soft Computing*, *77*, 1-23.

Ghaleini, E. N., Koopialipoor, M., Momenzadeh, M., Sarafraz, M. E., Mohamad, E. T., & Gordan, B. (2019). A combination of artificial bee colony and neural network for approximating the safety factor of retaining walls. *Engineering with Computers*, *35*(2), 647-658.

Hansen, T. D., Kaplan, H., Zamir, O., & Zwick, U. (2019, June 23-26). Faster k-SAT algorithms using biased-PPSZ. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (pp. 578-589). Phoenix, Arizona, USA.

Haque, M. R., Saud, A. A., Annajiat Yasmin, B., & Hossain, S. (2019). *Solving multi-class classification tasks with classifier ensemble based on clustering* (Bachelor Degree). United International University, Bangladesh.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, *79*(8), 2554-2558.

Hopfield, J. J., & Tank, D. W. (1985). "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, *52*(3), 141-152.

Hosseini, S., & Al Khaled, A. (2014). A survey on the imperialist competitive algorithm metaheuristic: implementation in engineering domain and directions for future research. *Applied Soft Computing*, *24*, 1078-1094.

Hureira, D., & Vartanian, C. (2019). Machine learning and neural networks for real-time scheduling. *Recent Advances in Real-Time Systems as of 2019, 1*, 1-7.

Jayashree, J., & Kumar, S. A. (2019). Evolutionary correlated gravitational search algorithm (ECGS) with genetic optimized Hopfield neural network (GHNN)–A hybrid expert system for diagnosis of diabetes. *Measurement*, *145*, 551-558.

Kasihmuddin, M. S. B. M., & Sathasivam, S. (2016). Accelerating activation function in higher order logic programming. In *AIP Conference Proceedings* (Vol. 1750, No. 1, p. 030006). New York, USA: AIP Publishing.

Kasihmuddin, M. S. M., Mansor, M. A., & Sathasivam, S. (2016). Bezier curves satisfiability model in enhanced Hopfield network. *International Journal of Intelligent Systems and Applications*, *8*(12), 9-17.

Kasihmuddin, M. S. M., Mansor, M., & Sathasivam, S. (2017a). Robust artificial bee colony in the Hopfield network for 2-satisfiability problem. *Pertanika Journal of Science and Technology*, *25*(2), 453-468.

Kasihmuddin, M. S. M., Mansor, M. A., & Sathasivam, S. (2017b). Hybrid genetic algorithm in the Hopfield network for logic satisfiability problem. *Pertanika Journal of Science and Technology*, *25*(1), 139-152.

Kasihmuddin, M. S. M., Mansor, M. A., Basir, M. F. M., & Sathasivam, S. (2019). Discrete mutation Hopfield neural network in propositional satisfiability. *Mathematics*, *7*(11), 1-21.

Kho, L. C., Kasihmuddin, M. S. M., Mansor, M., & Sathasivam, S. (2020). Logic mining in league of legends. *Pertanika Journal of Science and Technology*, *28*(1), 211-225.

Kong, D., Hu, S., Wang, J., Liu, Z., Chen, T., Yu, Q., & Liu, Y. (2019). Study of recall time of associative memory in a memristive Hopfield neural network. *IEEE Access*, *7*, 58876-58882.

Koopialipoor, M., Armaghani, D. J., Hedayat, A., Marto, A., & Gordan, B. (2019). Applying various hybrid intelligent systems to evaluate and predict slope stability under static and dynamic conditions. *Soft Computing*, *23*(14), 5913-5929.

Li, F., Ge, L., Dan, W., Gu, Y., He, Q., & Sun, K. (2019a, April 12-15). Application of improved variable learning rate back propagation neural network in energy dispersion x-ray fluorescence quantitative analysis. In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)* (pp. 428-432). Chengdu, China.

Li, M., Lei, D., & Xiong, H. (2019b). an imperialist competitive algorithm with the diversified operators for many-objective scheduling in flexible job shop. *IEEE Access*, *7*, 29553-29562.

Lim, S., & Bang, H. (2010). Waypoint planning algorithm using cost functions for surveillance. *International Journal of Aeronautical and Space Sciences*, *11*(2), 136-144.

Liu, Y., Ghandar, A., & Theodoropoulos, G. (2019, May 4-5). A metaheuristic strategy for feature selection problems: Application to credit risk evaluation in emerging markets. In *2019 IEEE Conference on Computational Intelligence for Financial Engineering and Economics (CIFEr)* (pp. 1-7). Shenzhen, China.

Luke, S. (2013). *Essentials of metaheuristics*. Raleigh, USA: Lulu Press.

Mansor, M. A. B., Kasihmuddin, M. S. B. M., & Sathasivam, S. (2017). Robust artificial immune system in the Hopfield network for maximum k-satisfiability. *International Journal of Interactive Multimedia and Artificial Intelligence*, *4*(4), 63-71.

Mansor, M. A., & Sathasivam, S. (2016). Performance analysis of activation function in higher order logic programming. In *AIP Conference Proceedings* (Vol. 1750, No. 1, p. 030007). New York, USA: AIP Publishing.

Mansor, M. A., Kasihmuddin, M. S. M., & Sathasivam, S. (2016). VLSI circuit configuration using satisfiability logic in Hopfield network. *International Journal of Intelligent Systems and Applications (IJISA)*, *8*(9), 22-29.

Mansor, M. A., Kasihmuddin, M. S. M., & Sathasivam, S. (2019). Modified artificial immune system algorithm with Elliot Hopfield neural network for 3-satisfiability programming. *Journal of Informatics and Mathematical Sciences*, *11*(1), 81-98.

Mansor, M. A., Sathasivam, S., & Kasihmuddin, M. S. M. (2018a). Artificial immune system algorithm with neural network approach for social media performance metrics. In *AIP Conference Proceedings* (Vol. 1974, No. 1, p. 020072). New York, USA: AIP Publishing.

Mansor, M. A., Sathasivam, S., & Kasihmuddin, M. S. M. (2018b). 3-satisfiability logic programming approach for cardiovascular diseases diagnosis. In *AIP Conference Proceedings* (Vol. 1974, No. 1, p. 020022). New York, USA: AIP Publishing.

Mollajan, A., Memarian, H., & Quintal, B. (2019). Imperialist competitive algorithm optimization method for nonlinear amplitude variation with angle inversion. *Geophysics*, *84*(3), 81-92.

Neelakanta, P. S., & DeGroff, D. (1994). *Neural network modeling: Statistical mechanics and cybernetic perspectives*. Florida, USA: CRC Press.

Nievergelt, J. (2000). Exhaustive search, combinatorial optimization and enumeration: Exploring the potential of raw computing power. In *International Conference on Current Trends in Theory and Practice of Computer Science* (pp. 18-35). Heidelberg, Germany: Springer.

Peri, D. (2019). Hybridization of the imperialist competitive algorithm and local search with application to ship design optimization. *Computers and Industrial Engineering*, *137*, 1-11.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, *3*, 1371-1382.

Sathasivam, S. (2010). Upgrading logic programming in Hopfield network. *Sains Malaysiana*, *39*(1), 115-118.

Sathasivam, S. (2011). Learning rules comparison in neuro-symbolic integration. *International Journal of Applied Physics and Mathematics*, *1*(2), 129-132.

Sathasivam, S. (2012). First order logic in neuro-symbolic integration. *Far East Journal of Mathematical Sciences*, *61*(2), 213-229.

Sathasivam, S., & Abdullah, W. A. T. W. (2011). Logic mining in neural network: Reverse analysis method. *Computing 91*(2), 119-133.

Sathasivam, S., Hamadneh, N., & Choon, O. H. (2011). Comparing neural networks: Hopfield network and RBF network. *Applied Mathematical Sciences*, *5*(69), 3439-3452.

Shi, J., Zhu, Y., Khan, F., & Chen, G. (2019). Application of Bayesian regularization artificial neural network in explosion risk analysis of fixed offshore platform. *Journal of Loss Prevention in the Process Industries*, *57*, 131-141.

Tashayo, B., Behzadafshar, K., Tehrani, M. S., Banayem, H. A., Hashemi, M. H., & Nezhad, S. S. T. (2019). Feasibility of imperialist competitive algorithm to predict the surface settlement induced by tunneling. *Engineering with Computers*, *35*(3), 917-923.

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research 30*(1), 79-82.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67-82.

Xiao, L., Wang, X. H., Wang, B. Z., Zheng, G., & Chen, P. (2017). An efficient hybrid method of iterative mom-po and equivalent dipole-moment for scattering from electrically large objects. *IEEE Antennas and Wireless Propagation Letters*, *16*, 1723-1726.