





Article

Amazon Employees Resources Access Data Extraction via Clonal Selection Algorithm and Logic Mining Approach

Nur Ezlin Zamri ¹, Mohd. Asyraf Mansor ^{1,*} , Mohd Shareduwan Mohd Kasihmuiddin ² ,
Alyaa Alway ¹, Siti Zulaikha Mohd Jamaludin ²  and Shehab Abdulhabib Alzaeemi ² 

¹ School of Distance Education, Universiti Sains Malaysia, Penang 11800, Malaysia; ezlinzamri@student.usm.my (N.E.Z.); alyaalway@student.usm.my (A.A.)

² School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia; shareduwan@usm.my (M.S.M.K.); szulaikha.szmj@usm.my (S.Z.M.J.); shehab_alzaeemi@yahoo.com (S.A.A.)

* Correspondence: asyrafman@usm.my; Tel.: +604-6535906

Received: 23 March 2020; Accepted: 20 April 2020; Published: 27 May 2020



Abstract: Amazon.com Inc. seeks alternative ways to improve manual transactions system of granting employees resources access in the field of data science. The work constructs a modified Artificial Neural Network (ANN) by incorporating a Discrete Hopfield Neural Network (DHNN) and Clonal Selection Algorithm (CSA) with 3-Satisfiability (3-SAT) logic to initiate an Artificial Intelligence (AI) model that executes optimization tasks for industrial data. The selection of 3-SAT logic is vital in data mining to represent entries of Amazon Employees Resources Access (AERA) via information theory. The proposed model employs CSA to improve the learning phase of DHNN by capitalizing features of CSA such as hypermutation and cloning process. This resulting the formation of the proposed model, as an alternative machine learning model to identify factors that should be prioritized in the approval of employees resources applications. Subsequently, reverse analysis method (SATRA) is integrated into our proposed model to extract the relationship of AERA entries based on logical representation. The study will be presented by implementing simulated, benchmark and AERA data sets with multiple performance evaluation metrics. Based on the findings, the proposed model outperformed the other existing methods in AERA data extraction.

Keywords: Boolean satisfiability; clonal selection algorithm; data extraction; human resources management; logic mining

1. Introduction

Amazon.com Inc. operates internationally by offering consumers products and subscriptions through more than 10 owned retail websites and physical stores in 600 locations across the United States of America (US). As reported in 2019, the company has increasing numbers of employees, and more than 600,000 employees worldwide [1]. Thus, with a large number of employees, there is always a risk of highly complicated employees and resources situations [2]. Within any company, new employees require a variety access of systems, portals or appliances related on the role, designation or unit of the employee. Technology companies like Amazon.com Inc. provide various types of resources; from computing to storage resources, accessible by their employees to be utilized optimally [3]. However, most of the time, employees encounter some complications prior to fulfilling their daily tasks. For example, computing resources opt to have Wi-Fi connection or they are unable to log in into Amazon.com Inc. human resources portal. Commonly, new resources applications are being processed and reviewed by distinct human administrators. It is worth mentioning that the downside of this common practice

involves a chain of human involvement that could lead to higher cost of resource maintenance and could be time-consuming. Therefore, Amazon.com Inc. made public their historical data from 2010–2011 of Amazon Employees Resources Access (AERA) data set, provisioned by Ken Montanez from Information Security of Amazon.com Inc. in partnership with Kaggle. Their motive is to seek alternative models that will prioritize the needs of employees and minimize manual resources access applications. A study by [4] proposed a forecasting model by using random forest (RF), logistic regression (LR) and gradient boosting (GB). However, the suggested approach was restricted to statistical linear classifiers and required a preprocessing step due to the imbalanced entries of AERA. One may question what makes this experiment significant from the work by [4]? In this paper, the main objective is to propose an alternative model in the field of data science by incorporating Artificial Neural Networks (ANNs) with Metaheuristic and Satisfiability representation (SAT). The proposed model act as a platform of knowledge extraction to handle big data which could benefit other big companies like Walmart Inc., Apple Inc., Samsung Electronics etc. in resources management.

ANN comprises parallel and nonparallel computing networks that are inspired by the mechanism of human biological brain [5]. ANN has several comprehensive architectures of feed-forward or feedback networks. Artificial Intelligence (AI) practitioners utilized ANN as a platform in applications such as entity classification problems [6], conducting analysis [7,8], pattern recognition [9,10], clustering problems [11,12] and circuits [13,14]. Nonetheless, another popular network of feedback ANN is the Hopfield Neural Network (HNN), which was formulated by [15] to solve optimization tasks. The extensive structure of HNN comprises energy function and associative property of content addressable memory (CAM). The work of [16] utilized HNN for transmitting binary amplitude modulated signals based on the potential energy function yielding lower probability of error. In addition, the work of [17] emphasized HNN as one of the most studied attractor-memory models due to the feature of useful Content Addressable Memory (CAM) for an optimization model. Note that HNN can be split into continuous HNN (CHNN) and discrete HNN (DHNN). The structure of DHNN consists of input and output neurons that store bipolar $\{1, -1\}$ or binary $\{1, 0\}$ pattern [18]. In addition, DHNN utilizes the Lyapunov energy function to determine degree of convergence of the solution [19]. This paper incorporates the Wan Abdullah (WA) method of finding the synaptic weights by comparing the Lyapunov energy function with the cost function [20]. The core impetus of the presented works is the relevancy of utilizing DHNN as a comprehensive model of AI as a platform to solve optimization tasks. Although DHNN is a “black box” model, the best way to observe DHNN behaviour is by implementing a systematic symbolic rule during the learning phase and a retrieved phase equation. Hence, one of the alternative ways to represent information theory is by the concept of satisfiability.

Satisfiability representation (SAT) is a logical and mathematical knowledge representation that plays a significant role in AI. SAT is utilized in various applications and areas such as quantum chemistry [21], approximation model [22], classification [23], chaos computing [24] and fault detection [25]. The SAT structure consists of clauses comprises of literals or variables. Why is SAT needed in DHNN? SAT is essential to provide symbolic instruction in attempt to represent the output of DHNN. Pioneer work by [26] showed the adaptability of Horn-SAT to represent information in executing the DHNN model that was improved later by [27]. The work improvised the existing model in neuro-symbolic integration model that gained more than 90% of global minimum energy. However, the restricted component in using Horn formula is limited in representing real-life data sets, which indicate not all real-life problem can be formulated in Horn-SAT [28]. Therefore, several researchers further extend the fundamental of Horn-SAT by proposing of DHNN model with different k -Satisfiability (k -SAT) logical representation [29–31]. These works emphasized on utilizing k -SAT, Maximum k -SAT (MAX k -SAT) and Maximum 2-SAT (MAX2-SAT) to investigate the ability of DHNN to process k -SAT patterns. In another development, data mining is a process of recognizing sequences or patterns in real-life data sets that involve various platforms. The difference between data mining and logic mining is that the logical rule mining utilizes logic to convey the information to the end user. Contingent upon that, the earliest logic mining method, the reverse analysis (RA) method, was introduced by [32] and it accommodates the combination of

RA and logic programming in DHNN to deduce the pattern and relationship of the real-life data sets. Subsequently, [33] utilized previous work of building a knowledge extraction tool by forming *k*-Satisfiability-based Reverse Analysis (*k*-SATRA). *k*-SATRA carries an important role in logic mining to display the true behaviour or pattern of a real-life data set by extracting the optimum logic that represents the relationship of the attributes. The extracted logic will represent information aligned with the specific classification tasks. An interesting application of *k*-SATRA is reported by the work of [34], which investigates students' performance in identifying related factors of underachievement students. The work entrenched several real-life data sets and obtained higher accuracy than two other existing educational data mining methods. Another development of utilizing *k*-SATRA was by [35] and [36] which exhibited the ability 2-SATRA to extract key findings of online games and football matches. The common denominator of these works exhibits the practicability of *k*-SATRA in extracting knowledge from a real-life data set. The extracted knowledge identifies relationships of attributes that affect the final outcome. However, there are no current works creating a platform to bridge logic and data mining methods with specific optimization tasks such as those encountered by Amazon.com Inc. of detecting which factor should be prioritized in order to grant or revoke employees resources applications. The incorporation of metaheuristics like the Clonal Selection Algorithm (CSA) in the training phase would capitalize better on the learning environment for an optimal optimization model.

The Metaheuristics Algorithm is a nonderivative method that searches near optimal solutions with specific constraints. [37] presented various applications of metaheuristics to find high-quality solutions to increasing number of ill-defined and complex real-world problems. Metaheuristics garnered much attention, especially from ANN practitioners, because metaheuristics provides a better learning mechanism of ANN networks by specifying the searching space of solutions and focusing on gradual solution improvement [38]. Conventionally, DHNN deployed the primitive learning rule of exhaustive search (ES), a trial and error mechanism to find solutions [39]. ES increases the probability of overfitting [40] and generates less variation of solutions [41]. CSA is an evolutionary algorithm, inspired by the natural phenomenon of the biological immune system, which defends the body against external microorganisms. [42] reviewed recent works by researchers implementing CSA into their proposed network to deal with constraint optimization tasks, such as pattern recognition [43], scheduling [44], fault detection [45] and dynamic optimization [46]. Mechanisms of CSA gives the inspiration of specific cells to recognize specific antigens which are later selected to proliferate. This resulted in a learning algorithm of evolving candidate solutions by selection, cloning and somatic hypermutation procedures, which established variation of solutions. Conjointly, the mechanism of CSA sets a new paradigm of solving optimization tasks. Pioneer work by [47] introduced the affinity-based interaction for modified CSA as a solver with the tabu search technique for the Maximum 3-SAT (MAX3-SAT) problem. The suggested model yielded quality solutions. Therefore, to predict the resources access applications for future sets of employees of AERA, this paper capitalizes on fundamental DHNN by incorporating CSA in the learning phase to overcome conventional metaheuristic drawbacks. The proposed model sets apart from previous literature due to different role of CSA to facilitate the learning phase of DHNN for 3-SAT logic, resulting a single intelligent unit that incorporates real-life data set to help Amazon.com Inc. resources access management.

To our best knowledge, no current work has proposed the incorporation of DHNN with CSA for 3-SATRA logic-mining methods. An optimal model may result in better management from Amazon.com Inc. in providing the best care for their employees. Subsequently, the contributions of this work are stated as follows: (1) To transform AERA into a 3-SAT logical representation to best represent the relationship of AERA. (2) To construct a modified DHNN model with CSA to enhance the learning phase of DHNN. (3) To utilize the 3-SATRA method into our proposed model as an alternative method to extract information from AERA in the form of logical representation. (4) To demonstrate the capability of our proposed model by conducting a simulated data set, benchmark data sets and the AERA data set in comparison with other existing methods. The comparison will be also evaluated by using appropriate performance evaluation metrics. The findings of this paper displayed the competency of our proposed model outperformed other

existing methods for all type of data sets. Figure 1 illustrates the implementation of our contribution in this paper.

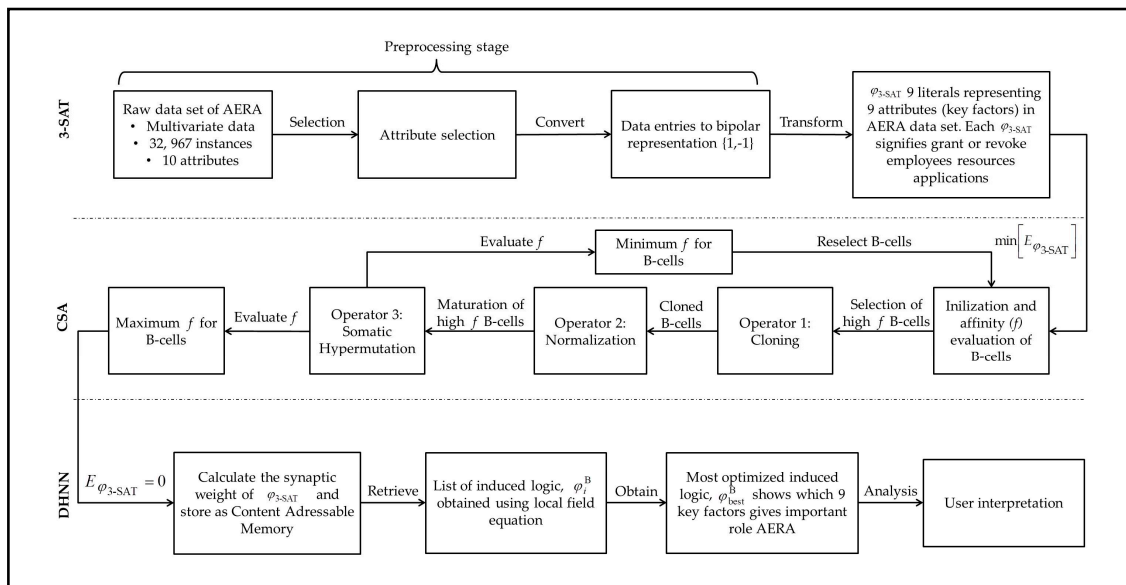


Figure 1. Implementation of the proposed model.

2. Boolean Satisfiability

Boolean satisfiability logic (SAT) represents a task in determining truth assignments that makes the logical rule satisfiable. SAT is a nondeterministic polynomial time, NP-complete problem where SAT can be solved in polynomial time by a nondeterministic Turing machine [48]. In this paper, SAT is represented in a conjunctive normal form (CNF) and composed of three significant elements [49]:

A group of m variables: $a_1 \vee a_2 \vee \dots \vee a_m$ where $a_i \in \{1, -1\}$.

A group of literals: A literal is a variable (a_1) or a negation of a variable (\bar{a}_1).

A group of n clauses: $A_1 \wedge A_2 \wedge \dots \wedge A_n$.

The above elements can be explicitly represented in the following Equation (1):

$$\varphi_{3-SAT} = \bigwedge_{i=1}^n A_i \text{ where } A_i = (a_i, b_i, c_i) \tag{1}$$

This paper utilized 3-Satisfiability (3-SAT) logical rule, φ_{3-SAT} , in each clause of which only exist three variables. Equation (2) governed an example of φ_{3-SAT} . Note that φ_{3-SAT} represents the objective or outcome of the logical rule.

$$\varphi_{3-SAT} = (\bar{P} \vee Q \vee R) \wedge (S \vee \bar{T} \vee \bar{U}) \wedge (\bar{V} \vee W \vee X) \tag{2}$$

Table 1 shows an example of cases for the φ_{3-SAT} logical rule. The outcomes of each case are known by substituting the values of $\{1, -1\}$ (neuron states) into Equation (2). For instance, case 1 is satisfiable since each clause gives a truth value. Besides that, case 3 is in full consistency since all literals give a truth value. Additionally, the work by [49] states that the algorithm needs to learn more inconsistent interpretations to obtain the satisfied φ_{3-SAT} , which we described as the checking clause satisfaction process. To undergo this process, a suitable metaheuristic algorithm is needed to attain $\varphi_{3-SAT} = 1$ [47]. In this paper, the φ_{3-SAT} logical rule is employed in our proposed model to govern our model and represent each entry of AERA.

Table 1. Example of cases for the 3-Satisfiability (3-SAT) logical rule, $\varphi_{3\text{-SAT}}$.

Case	$\varphi_{3\text{-SAT}}$ Instances	Outcome
1	$(P, Q, R, S, T, U, V, W, X) = (1, 1, 1, -1, -1, -1, 1, 1, -1)$	Satisfiable ($\varphi_{3\text{-SAT}} = 1$)
2	$(P, Q, R, S, T, U, V, W, X) = (1, -1, -1, 1, 1, -1, 1, 1, 1)$	Unsatisfiable ($\varphi_{3\text{-SAT}} = -1$)
3	$(P, Q, R, S, T, U, V, W, X) = (-1, 1, 1, 1, -1, -1, -1, 1, 1)$	Full consistency ($\varphi_{3\text{-SAT}} = 1$)
4	$(P, Q, R, S, T, U, V, W, X) = (1, -1, -1, -1, 1, 1, 1, -1, -1)$	Full inconsistency ($\varphi_{3\text{-SAT}} = -1$)

3. 3-Satisfiability in Discrete Hopfield Neural Network

The Discrete Hopfield Neural Network (DHNN) is another variant of the Hopfield Neural Network that is commonly utilized to solve practical optimization problems [50]. DHNN consists of interconnected neurons with no hidden layer. Each neuron in DHNN is bipolar $S_i \in \{-1, 1\}$, $i \in \mathbb{N}$, which exemplifies the interpretation of the defined problem. Several properties of DHNN include associative memory, fault tolerance and energy minimization as the neuron state changes. There are two types of neuron updates in DHNN: asynchronous and synchronous update. We limit our discussion to asynchronous update because we only consider one neuron state at the time. Each neuron spin resembles an Ising spin variable model [51], which contributes to updating neurons in each cycle. The general updating rule of the general DHNN is given as follows:

$$S_i = \begin{cases} 1, & \text{if } \sum_j^N W_{ij} S_j \geq \rho_i \\ -1, & \text{otherwise} \end{cases} \quad (3)$$

where W_{ij} and ρ_i are the synaptic weight and threshold of the constraints. It is worth mentioning that we consider $\rho_i = 0$ to ensure the energy of DHNN decreases uniformly [52]. W_{ij} in each neurons connection is formally defined in a matrix of $W_{ij}^{(2)} = \left[W_{ij}^{(2)} \right]_{N \times N}$ with the threshold of the neuron updates given by $[\rho_i]_{n \times 1} = [\rho_1, \rho_2, \rho_3, \dots, \rho_n]^T$. Note that DHNN has no self-looping $W_{ii}^{(2)} = W_{jj}^{(2)} = 0$ for all neurons and the connection is symmetrical $W_{ij}^{(2)} = W_{ji}^{(2)}$ which results in a matrix with zeros diagonal. The updating rule of general DHNN is important to ensure the neuron state will converge to the optimal solution. In this section, we capitalize the logical rule of $\varphi_{3\text{-SAT}}$ into the structure of DHNN by defining the cost function of the network. $\varphi_{3\text{-SAT}}$ can be implemented in DHNN by minimizing the cost function $E_{\varphi_{3\text{-SAT}}}$:

$$E_{\varphi_{3\text{-SAT}}} = \sum_{i=1}^{NC} \prod_{j=1}^N D_{ij} \quad (4)$$

where NC and N are the number of clauses and number of literals accordingly. D_{ij} is defined as follows:

$$D_{ij} = \begin{cases} \frac{1}{2}(1 + S_X), & \text{if } X \neq \bar{X} \\ \frac{1}{2}(1 - S_X), & \text{otherwise} \end{cases} \quad (5)$$

where X is one possible variable in $\varphi_{3\text{-SAT}}$. Note that the lowest value of the cost function is $E_{\varphi_{3\text{-SAT}}} = 0$ where all the inconsistencies of the $\varphi_{3\text{-SAT}}$ are minimized. Hence the updating rule or local field of the $\varphi_{3\text{-SAT}}$ in DHNN is given as follows:

$$h_i(t) = \sum_{k=1, k \neq j}^N \sum_{j=1, j \neq k}^N W_{ijk}^{(3)} S_k S_j + \sum_{j=1, j \neq i}^N W_{ij}^{(2)} S_j + W_i^{(1)} \quad (6)$$

$$S_i(t + 1) = \begin{cases} 1, & \sum_{k=1, k \neq j}^N \sum_{j=1, j \neq k}^N W_{ijk}^{(3)} S_k S_j + \sum_{j=1, j \neq i}^N W_{ij}^{(2)} S_j + W_i^{(1)} \geq 0 \\ -1, & \sum_{k=1, k \neq j}^N \sum_{j=1, j \neq k}^N W_{ijk}^{(3)} S_k S_j + \sum_{j=1, j \neq i}^N W_{ij}^{(2)} S_j + W_i^{(1)} < 0 \end{cases} \quad (7)$$

where $W_{ijk}^{(3)}$, $W_{ij}^{(2)}$ and $W_i^{(1)}$ are synaptic weight for the third, second and first order connection respectively. The threshold for the proposed DHNN is $\rho = 0$ and can be flexibly defined by the user. According to [53], the final neuron state, $S_i(t + 1)$, can be optimized by the usage of a squashing function such as a Hyperbolic Activation Function (HTAF). Interested readers on this aspect may refer to [49,53,54]. Furthermore, Equations (6) and (7) are vital to ensure the final neuron state always converges to $E_{\varphi_{3-SAT}} \rightarrow 0$. Theorem 1 explains the behaviour of the synaptic weight with respect to the final state of the neuron.

Theorem 1. *Let $N = (W, \rho)$ where ρ is the threshold of the model of DHNN. Assuming N operates in asynchronous mode and W is a symmetric matrix with the elements of the diagonal being nonnegative. Then DHNN will always converge to a stable state.*

In addition, the Lyapunov energy function $L_{\varphi_{3-SAT}}$ that corresponds to the φ_{3-SAT} rule is given as follows:

$$L_{\varphi_{3-SAT}} = -\frac{1}{3} \sum_{i=1, i \neq j \neq k}^N \sum_{j=1, i \neq j \neq k}^N \sum_{k=1, i \neq j \neq k}^N W_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2} \sum_{i=1, i \neq j}^N \sum_{j=1, i \neq j}^N W_{ij}^{(2)} S_i S_j - \sum_{i=1}^N W_i^{(1)} S_i \quad (8)$$

The value of $L_{\varphi_{3-SAT}}$ indicates the quality of the final state obtained from Equation (8). According to [20], the synaptic weight of the DHNN can be obtained by comparing Equations (4) and (6). The energy value of the φ_{3-SAT} , $L_{\varphi_{3-SAT}}^{\min}$, can be predetermined before the learning phase because the energy value from each clause in φ_{3-SAT} is always constant. It is worth mentioning that the optimal DHNN always converges to $L_{\varphi_{3-SAT}} \rightarrow L_{\varphi_{3-SAT}}^{\min}$ or $|L_{\varphi_{3-SAT}} - L_{\varphi_{3-SAT}}^{\min}| \leq \partial$, where ∂ is the tolerance value of the Lyapunov energy function. In this paper, the information from the data set will be represented in terms of φ_{3-SAT} and embedded into DHNN. The implementation of φ_{3-SAT} in DHNN is abbreviated as DHNN-3SAT. One of the main obstacles in implementing DHNN-3SAT is to find a set of S_i that corresponds to $E_{\varphi_{3-SAT}} = 0$. By that standard, optimal learning method is required to effectively minimize $E_{\varphi_{3-SAT}}$.

4. Clonal Selection Algorithm

The learning phase of an ANN can be further improved via metaheuristics to provide more global solutions, a better learning mechanism and to ascertain the convergence of the ANN models [55]. A work proposed by [56] indicated that these algorithms required less execution time to complete the training process. Generally, metaheuristics have two types of searching algorithms, trajectory-based and population-based. The work is focusing on the population-based nature-inspired algorithm of evolutionary algorithms (EA). CSA is a class of Artificial Immune System (AIS) algorithms that is motivated by the natural immune system process that build particular antibodies against antigens. B-cells (β) will produce specific antibodies once a new antigen is identified. Through the cloning process, the chosen β will proliferate to form a clone of β and fight against antigens [57]. The cloned β developed into two types of cell, memory cells and plasma cells. Memory cells are recognized as long-lived cells that can react instantly to any illness. As for the plasma cells, they are active and able to secrete specific antibodies for the antigens, but they do not last long.

The findings by Layeb [47] presented modified CSA with the tabu search method to resolve the satisfiability problem. The affinity computation in [47] utilized the adaptive affinity function, which considers the summation of weight with the clauses and complies with the binary vector form

of MAX-SAT logical representation. On the other hand, our proposed CSA model complies to bipolar representation of φ_{3-SAT} as the affinity function being formulated in terms of clause representation that corresponds to $E_{\varphi_{3-SAT}} = 0$. The operations involved in the CSA mechanism are where β produces a specific antibody to destroy a specific antigen, which signifies the adaptive system of CSA principle. Proliferation, normalization and somatic hypermutation processes ensure a better variation of the β population. This paper implements the CSA mechanism to provide an optimal learning model, where CSA helps to achieve maximum number of satisfied clauses from the affinity or fitness of β . The implementation of CSA in the proposed model (DHNN3-SATCSA) is presented as follows [58]:

Stage 1: Initialization of β

$\beta = 100$ (interpretations) were initialized.

$$\beta_{ij} = \begin{cases} 1, rand[0, 1] \geq 0.5 \\ -1, otherwise \end{cases} \tag{9}$$

Stage 2: Affinity Evaluation

Compute affinity of all β in the entire population, f_{β_i} . The f_{β_i} examines the number of clauses satisfied in φ_{3-SAT} . A_i is the number of clauses learned by CSA and NC is the number of clauses in φ_{3-SAT} .

$$f_{\beta_i} = \sum_{i=1}^{NC} A_i \tag{10}$$

where

$$A_i = \begin{cases} 1, True \\ 0, False \end{cases} \tag{11}$$

given that $f_{\beta_i} \in \mathbb{Z}$.

Stage 3: Proliferation via Cloning

The top five β with higher affinity were chosen to proliferate in cloning process. In this process, β will be duplicated by applying the roulette wheel mechanism [59]. The number of cloned β , N_ϵ will be computed by using Equation (12).

$$N_\epsilon = \frac{\lambda f_i}{\sum_{i=1}^{NC} f_{\beta_i}} \tag{12}$$

where λ , known as initial affinity, is the population clone size which the software seeks to implement into the searching space. [58] suggested selecting $\lambda = 200$.

Stage 4: Normalization

Equation (13) shows the list of cloned β ; $1 \leq \beta_i^C \leq N_\epsilon$.

$$\beta_i^C = \{\beta_1^C, \beta_2^C, \beta_3^C, \dots, \beta_{N_\epsilon}^C\} \tag{13}$$

Normalization of β_i^C ($\widetilde{\beta}_i^C$) is often called immune response maturation throughout the system. It is important to normalize the β_i^C before proceeding to the next step. Next, we calculate the affinity for each $\widetilde{\beta}_i^C$, which is abbreviated as $f_{\widetilde{\beta}_i^C}$.

$$f_{\widetilde{\beta}_i^C} = \frac{f_{\beta_i^C} - \min|f_{\beta_i^C}|}{\max|f_{\beta_i^C}| - \min|f_{\beta_i^C}|} \tag{14}$$

where

$$\max |f_{\beta_i^C}| - \min |f_{\beta_i^C}| > 0 \tag{15}$$

Note that $\max |f_{\beta_i^C}| \neq \min |f_{\beta_i^C}|$ because the probability of getting $f_{\beta_i^C} = 0$ is almost zero.

Stage 5: Somatic Hypermutation

The somatic hypermutation process is significant since it will ensure the β to achieve highest affinity which results in a feasible solution. Equation (16) shows the calculation of the number of mutations (N_{ζ}) for each β_i^C .

$$N_{\zeta} = \left(\frac{f_{\beta_i^C}}{\eta} \right) + \theta (1 - f_{\beta_i^C}) \tag{16}$$

where η is the number of variables in φ_{3-SAT} , $\theta = 0.01$ and $\eta \neq 0$ [58]. For every mutation that occurs in N_{ζ} , one or more β will be flipped from 1 to -1 or vice versa.

Finally, the f_{β_i} of mature population will be computed and we will choose the best β as the candidate cell to be kept in the memory cell. The solution will be selected if $f_{\beta_i^C} = NC$. On the other hand, the process will repeat from stages 2 to 5 if $f_{\beta_i^C} \neq NC$. Figure 2 shows the summary of all steps involved in CSA.

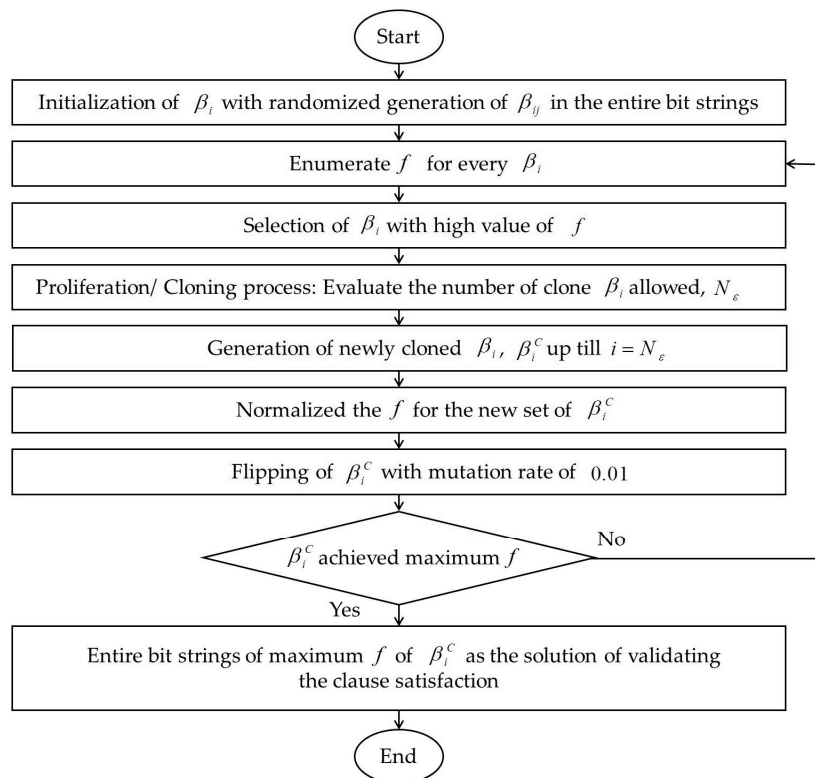


Figure 2. Summary of Clonal Selection Algorithm (CSA).

5. 3-Satisfiability Based Reverse Analysis Method

Logic mining is a process that utilizes logic programming to extract information from a data set. In this regard, this section will explain how the logic mining tool named 3-Satisfiability-based Reverse Analysis Method (3-SATRA) is implemented in our DHNN3-SATCSA model to extract the relationship of AERA entries. Consider n attributes of the data sets $(S_1, S_2, S_3, \dots, S_n)$, where $S_i \in \{1, -1\}$. Note that all binary representations must be represented in terms of bipolar states. Since this paper investigates

$\varphi_{3\text{-SAT}}$, the arrangement of each A_m consists of S_i, S_j, S_k where $i \neq j \neq k$. Note that NC is the number of clauses in $\varphi_{3\text{-SAT}}$. For A_m that leads $\varphi_{3\text{-SAT}}^{\text{learn}} = 1$, we assign

$$A_m = \left(S_i^{\max[n(S_i)]} \vee S_j^{\max[n(S_j)]} \vee S_k^{\max[n(S_k)]} \right) \tag{17}$$

Note that $\max[n(S_i)]$ signifies the highest frequency of S_i . In this case, each S_i is given as follows:

$$S_i = \begin{cases} S_i, & S_i = 1 \\ \bar{S}_i, & S_i = -1 \end{cases} \tag{18}$$

By using the obtained A_m , we can formulate $\varphi_{3\text{-SAT}}^{\text{best}}$:

$$\varphi_{3\text{-SAT}}^{\text{best}} = \bigvee_{m=1}^{NC} A_m \tag{19}$$

For example, we will choose $A_1 = (S_1 \vee \bar{S}_2 \vee \bar{S}_3)$ if $S_1^{\max[n(S_1)]} = S_1, S_2^{\max[n(S_2)]} = \bar{S}_2$ and $S_3^{\max[n(S_3)]} = \bar{S}_3$. Next, $\varphi_{3\text{-SAT}}^{\text{best}}$ will be embedded into DHNN. Henceforth, we will obtain the states of S_i that correspond to $E_{\varphi_{3\text{-SAT}}^{\text{best}}} = 0$. By comparing Equation (4) with Equation (8), the corresponding W_{ij} will be obtained. During the testing phase, the induced states, S_i^B , will be obtained by using Equation (6). Subsequently, the induced logic, φ_i^B will be constructed based on the rule given in Equation (2). Finally, the chosen induced logic is obtained based on $\varphi_i^B = \varphi_i^{\text{test}}$ (testing data). Figure 3 demonstrates how 3-SATRA was implemented in the DHNN model. In this paper, we will represent each neuron with entries of AERA.

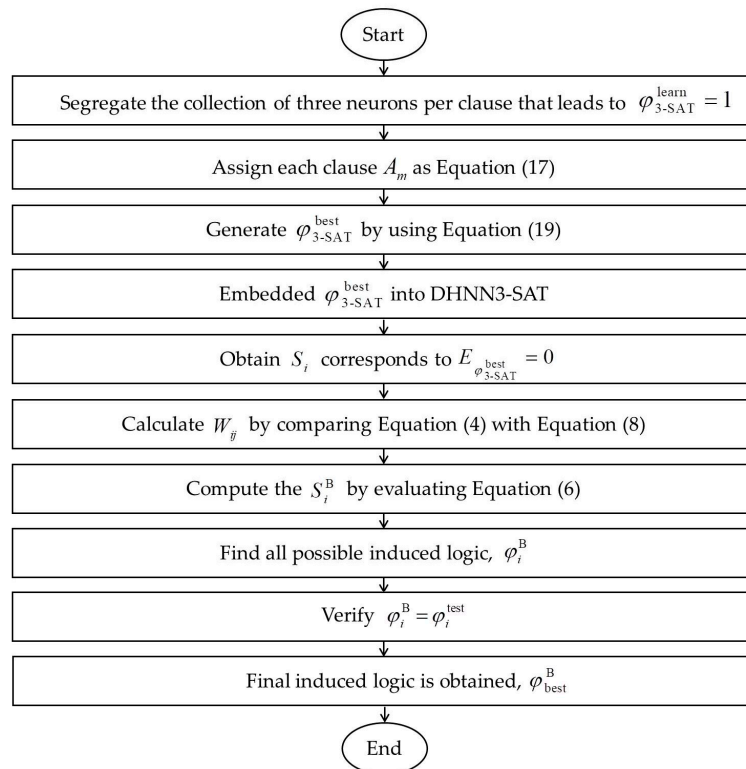


Figure 3. Implementation of 3-Satisfiability Reverse Analysis (3-SATRA) in the Discrete Hopfield Neural Network (DHNN).

6. Experimental Setup

A standard procedure among ANN practitioners is to investigate the proposed model with other comparative studies. Therefore, the simulation process is divided into three sections. Firstly, the performance of DHNN3-SATCSA is analyzed by using simulated data sets. In this case, the ability of CSA in the learning phase of the proposed model will be compared with other existing methods [58,60]. Secondly, several benchmark data sets will be implemented into DHNN3-SATCSA. The comparison of the retrieval properties of DHNN3-SATCSA will be also evaluated based on $\varphi_i^B = \varphi_i^{\text{test}}$. The third section presents the implementation of AERA into the proposed model. All real-life data sets that were converted to bipolar representation and information extraction will be conducted via 3-SATRA and incorporated with DHNN3-SAT models.

In the first section, DHNN with linearized initial neuron states might result in the biasedness of the retrieval state because the network simply memorizes the final state without producing a new state [61]. Therefore, possible positive and negative biases can be reduced by generating all the neuron states randomly as in Equation (20):

$$S_i(t) = \begin{cases} 1, & \text{rand}[0, 1] < 0.5 \\ -1, & \text{otherwise} \end{cases} \quad (20)$$

where S_i is defined as in Equation (3). The simulated data set will be initiated by generating randomized clauses and literals for each $\varphi_{3\text{-SAT}}$. A similar approach has been implemented in several studies such as [19,60] in generating the initial neuron states. It is worth mentioning that all simulations will be measured against existing methods by evaluating appropriate performance evaluation metrics. By quoting several relevant studies that implemented such experimentations [35,49,62], the proposed performance metrics in this experiment are mean absolute error (MAE), sum of square error (SSE), global minima ratio (ω), accuracy in percentage (α) and computational time in SI unit of second (CT). According to [63], MAE computes the average absolute error of the fitness during the learning phase in our proposed model. The formulation of MAE is as follows:

$$MAE = \sum_{i=1}^n \frac{1}{n} |\tau - v| \quad (21)$$

where τ and v are the total number of clauses and the number of satisfied clauses in $\varphi_{3\text{-SAT}}$ respectively. In relation to Equation (21), the accumulation of errors in each model can be also effectively evaluated by using SSE. The formulation of SSE is described by the following equation:

$$SSE = \sum_{i=1}^n (v - \tau)^2 \quad (22)$$

On the other hand, we examine the final neuron states of the proposed model via ω .

$$\omega = \frac{1}{ab} \sum_i^n N_{L\varphi_{3\text{-SAT}}} \quad (23)$$

According to [52], if the final neuron states of the proposed model is $E_{\varphi_{3\text{-SAT}}} \rightarrow 0$, the model will prone to $\omega \rightarrow 1$. Hence, the best model will attain the lowest value of MAE and SSE with $\omega \rightarrow 1$. Notably, $\omega \rightarrow 1$ indicates $n \left(\left| L_{\varphi_{3\text{-SAT}}}^{\min} - L_{\varphi_{3\text{-SAT}}} \right| \leq \lambda \right) \rightarrow ab$ where a and b are number of trials and neuron combinations, respectively. In another development, we utilize the value of α and CT to investigate the

effectiveness and efficiency of 3-SATRA in the testing phase of DHNN3-SATCSA. We describe two formulations:

$$\alpha = \frac{\varphi_i^B}{N_{\varphi_i^{\text{test}}}} \times 100\% \quad (24)$$

$$CT = \text{LearningTime} (s) + \text{RetrievalTime} (s) \quad (25)$$

Note that $\alpha \rightarrow 100$ if $n(\varphi_i^{\text{test}} = \varphi_i^B) \rightarrow G$, where G in our case is depicted by 40% of instances in a data set. In practice, the best model requires $\alpha \rightarrow 100$ and the minimum value of CT . Learning time and retrieval time are denoted as total time executed by DHNN3-SAT models in the learning phase and retrieval phase respectively. Tables 2 and 3 discuss the parameters involved in hybrid Hopfield Neural Network with Exhaustive Search (DHNN3-SATES) and hybrid model with Clonal Selection Algorithm (DHNN3-SATCSA) respectively.

Table 2. List of parameters in DHNN3-SATES [52].

Parameter	Parameter Value/Remarks
a	100
b	100
λ	0.001
CT	24 h
NN	$9 \leq NN \leq 72$
Selection Rate	0.1
Number of Strings	100
Type of Selection	Random

Table 3. List of parameters in DHNN3-SATCSA.

Parameter	Parameter Value/Remarks
$n(\beta)$	100
γ	200
θ	0.01
λ	0.001
CT	24 h
NN	$9 \leq NN \leq 72$
Type of Selection	Roulette Wheel Selection [59]
Learning Method	WA Method [20]

The choice of β is important as a large population size requires a large searching space of the solutions, which may increase the computational cost. On the other hand, a small β can lead to local minima solutions. According to [64], we should choose $\beta = 100$ as it is repeated to achieve a good result. The general implementation of the proposed model in a simulated data set can be summarized in Figure 4. 3-SATRA is implemented to show the level of connectedness between W_{ij} and neurons. Overall, simulated and real-life data sets will be implemented into DHNN3-SATCSA. The computational simulation for both data sets was conducted on Dev C++ Version 5.11 for Windows 7 in 2GB RAM with Intel Core I3. As for the simulated data set, the Dev C++ program will generate the initial bipolar data randomly. Throughout the simulations, the same device is being used to avoid any biases. On the whole, all simulations are utilized with different number of neurons (NN), which is within the bound of not exceeding the threshold time of 24 h [35]. Note that the proposed model will randomly select nine attributes for the real-life data set as well as their arrangements in $\varphi_{3\text{-SAT}}$ logical rule.

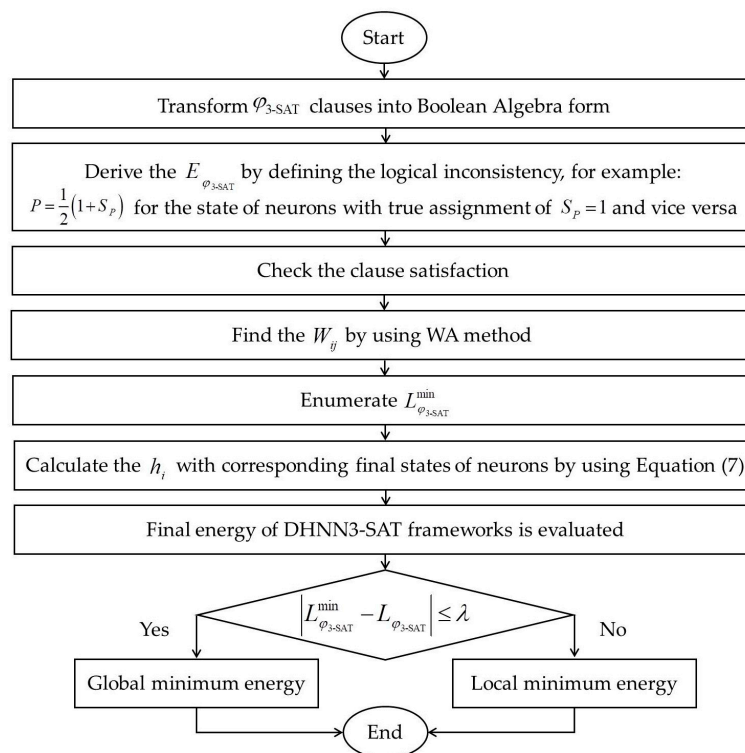


Figure 4. The implementation of DHNN3-SAT models in a simulated data set.

7. Results and Discussion

7.1. Simulated Data Set

The first section of the experiment was carried out by using simulated data. This section evaluates the performance of CSA as the learning rule in the DHNN model in comparison with ES. The findings of simulated data set for both models are presented as follows:

According to Figures 5 and 6, DHNN3-SATCSA accumulated fewer errors compared to DHNN3-SATES due to CSA's ability to learn and train the network effectively. However, ES incorporates random search which makes the complexity of the learning phase increase. As illustrated in Figure 7, DHNN3-SATCSA achieved a consistent value of $\omega = 1$, from $NN = 9$ to $NN = 72$ whereas DHNN3-SATES only gained a better value of ω after processing 62.5% of the total NN . ES projected unnecessary projection due to the "trial-and-error" feature that does not aid the proposed model to improve the solutions. CSA can manage a large number of constraints compared to ES. CSA made this possible because CSA showcased the ability of β in fighting the pathogens and improving the affinity values in the entire bit strings to help DHNN3-SATCSA search for ideal solutions. In this experiment, we did not consider α because the value of ω corresponds to the number of global minimum energy achieved by DHNN3-SAT models. Hence, the value of ω is adequate to represent the effectiveness of the retrieval phase of both models. The main distinction between these models with [47] is the formulation of fitness function. The cost function in [47] is $E_{\varphi_{3-SAT}} \neq 0$ because the structure of φ_{3-SAT} is not satisfiable. CSA reduces the number of iterations because the CSA optimization operator, particularly somatic hypermutation, will allow the solution to attain $E_{\varphi_{3-SAT}} = 0$ faster than ES. In general, CSA will reduce learning time, which will elongate relaxation time within the ideal rate and, we believe, result in less neuron oscillation. It is worth noting that the probability for somatic hypermutation to flip the neurons entirely is approaching to 0. Thus, the chances for the solution to achieve nonimproving fitness will reduce drastically compared to conventional ES. The Wan Abdullah method is chosen because this method is reported to contribute less neuron oscillation as compared to other methods such as Hebbian learning [65]. Uncontrollable neuron oscillation via other

methods such as Hebbian learning will lead to more local minimum energy or $|L_{\varphi_{3-SAT}}^{\min} - L_{\varphi_{3-SAT}}| > \lambda$. This comparison is vital to validate the learning capabilities of CSA. The limitation of DHNN3-SATCSA is the use of bipolar neuron states instead of another neuron representation (ternary), $S_i \in \{1, -1, 0\}$. Ternary representation can provide more analysis since it has another vector of 0 which indicates no response or meaningless result. In another perspective, the proposed model only considers satisfiable SAT logic. Other SAT representations such as MAXk-SAT [18] require major restructuring, especially in terms logical redundancy. Furthermore, this experiment only employs a nonrestricted learning environment where the CSA and ES will iterate until $E_{\varphi_{3-SAT}} = 0$. Finally, this work only embeds φ_{3-SAT} in terms of CNF form. According to [66], CNF representation is more compatible to the WA method compared to Disjunctive Normal Form (DNF) representation.

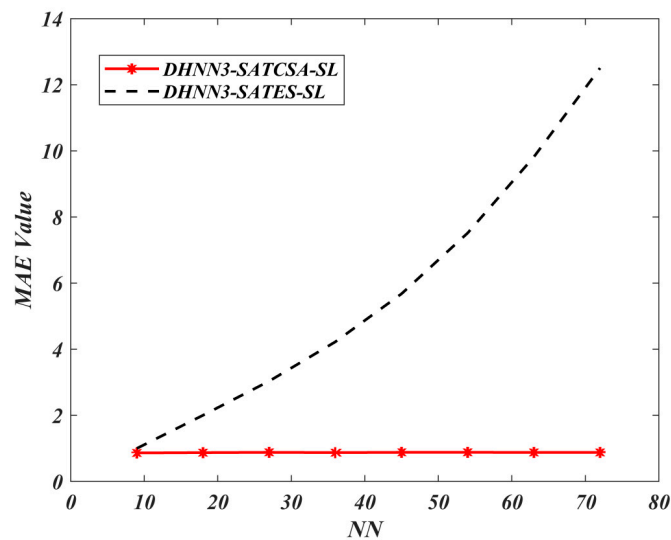


Figure 5. Mean absolute error (MAE) value of DHNN3-SAT models.

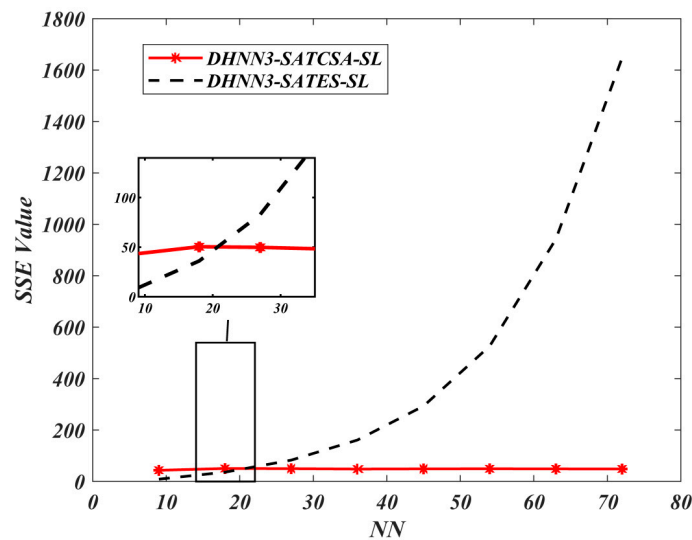


Figure 6. Sum of square error (SSE) value of DHNN3-SAT models.

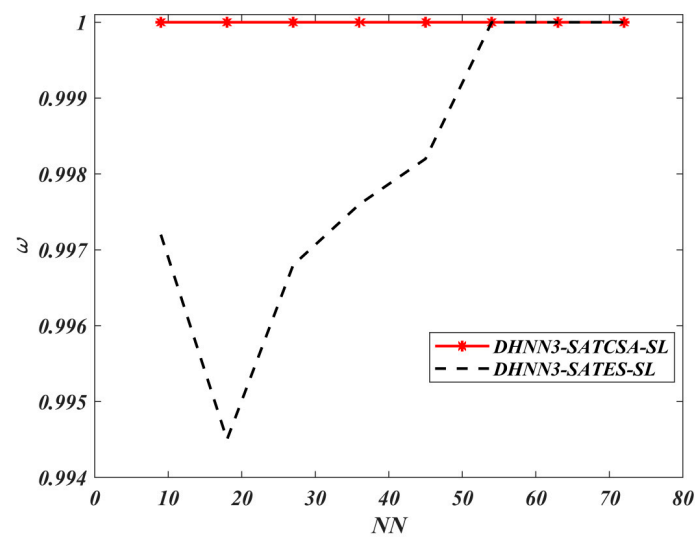


Figure 7. ω value of DHNN3-SAT models.

7.2. Benchmark Data Sets

For the second part of the experiment, the simulation is carried out over a set of four widely used benchmark real-life data sets [67] listed in Table 4. Note that this section evaluates the performance of DHNN-3SAT models in doing real life data sets. A benchmark data set is reported in this paper because these structured data sets are validation performances of the DHNN-3SAT models.

All attributes (consisting of nine literals for each data set) listed in Table 4 will be embedded into 3-SATRA using implementation in Figure 3. We choose data set from the different disciplines because each data set has different clustering behaviour. The objective of each DHNN3-SAT model in this section is to induce the best φ_i^B that classifies the outcome of the data sets. In general, the choice of outcome for each data set is given as follows:

1. BDMC: Client will subscribe a term deposit where 1 and -1 signify nonsubscription and subscription respectively.
2. CCDP: Response to default payment of credit card customers where 1 and -1 signifies nonpaymaster and paymaster respectively.
3. DRDD: Signs of diabetic retinopathy where 1 and -1 signifies the sign exist and nonexist respectively.
4. FLST: Customers interest where 1 and -1 signifies the show of noninterest and interest towards the product respectively.

In this section, we only evaluate the performance of the induced φ_i^B and disregard the result from learning error. The instances of the data sets will be divided into φ_i^{learn} (60%) and φ_i^{test} (40%) which follows the procedure of the logic-mining model proposed by Kho [35]. We found that more capacity for learned data (than the proposed proportion) will result in data overfitting. Thus, the best DHNN-3SAT model is measured based on the highest value of α .

The value of α for all models is shown in Figures 8–11. A higher value of α indicates the optimality of the model in retrieving φ_i^B . The results of the analyses discussed in Figures 8–11 are all based on the assumption that φ_i^B for all data sets had achieved $\omega = 1$. According to Figure 8, both models demonstrate the similar maximum value of $\alpha = 74\%$ for $NN = 72$ in the BDMC data set. Despite the similar value of α for both models at $NN = 72$, DHNN-3SATES reported high learning error compared to DHNN-3SATCSA. The small value of α for DHNN-3SATCSA at $10 \leq NN \leq 60$ is due to overfitting the solution during the retrieval phase of DHNN-3SAT. As seen in Figure 9, the overall trend of α is distinct where the proposed model achieved consistent value of 77% and 36% for all NN respectively. In Figures 10 and 11, the proposed DHNN-3SATCSA is reported to not be effective when the NN is small, although the α reached the maximum value at $NN \geq 15$. Despite a similar value of α for

both models in Figure 11, DHNN-3SATCSA achieved similar α in lower learning error. As observed in Figures 8–11, the proposed DHNN-3SATCSA in 3-SATRA exhibits competitive performance with respect to the learning error and α . The innovation of DHNN-3SATCSA lies in the solution diversity of β that prevents CSA from getting trapped in local minima energy. In this case, a promising β will be improved via hypermutation strategy during the learning phase of DHNN. In contrast, DHNN-3SATES has no optimization layer and in most cases will contribute to suboptimal φ_i^B (see Figure 9). We expect that DHNN-3SATES will exceed the threshold computational time when $NN > 88$ due to the structural limitation of ES. Hence, we can further agree that generally DHNN3-SATCSA is a better model in terms of α and the capability of its mechanism to employ different sizes and natures of real-life data sets. We can further improve the retrieval property of DHNN-3SATCSA by implementing a mutation operator such as in [19].

Table 4. List of benchmark data sets information.

Benchmark Data Sets/Field	Attributes	Instances	Sources
Bank Direct Marketing Campaign (BDMC)/Marketing	<i>P</i> : Age <i>Q</i> : Job <i>R</i> : Credit card status <i>S</i> : Housing loan <i>T</i> : Personal loan <i>U</i> : Last contact day of the month <i>V</i> : Last contact duration <i>W</i> : Number of days passed by after the client was last contacted from a previous campaign <i>X</i> : Number of contacts performed before this campaign	45,211	UCI Machine Learning Repository
Credit Card Default Payment (CCDP)/Finance	<i>P</i> : Amount of limit balance <i>Q</i> : Education <i>R</i> : Marital status <i>S</i> : History of repayment status in Month I <i>T</i> : History of repayment status in Month II <i>U</i> : Amount of bill statement in Month I <i>V</i> : Amount of bill statement in Month II <i>W</i> : Amount of previous payment in Month I <i>X</i> : Amount of previous payment in Month II	3000	UCI Machine Learning Repository
Diabetic Retinopathy Debrecen Disease (DRDD)/Health	<i>P</i> : Result of quality assessment <i>Q</i> : Result of pre-screening <i>R</i> : Features detection I <i>S</i> : Features detection II <i>T</i> : Features detection for exudates I <i>U</i> : Features detection for exudates II <i>V</i> : Affected patient condition according to the Euclidean Distance (center of the macula and the center of the optic disc) <i>W</i> : Diameter of the optic disc <i>X</i> : Result of the AM/FM-based classification	1151	UCI Machine Learning Repository
Facebook Live Sellers in Thailand (FLST)/Marketing	<i>P</i> : Status type <i>Q</i> : Number of comments <i>R</i> : Number of shared post <i>S</i> : Number of likes <i>T</i> : Number of reaction; Love emoticon <i>U</i> : Number of reaction; Wow emoticon <i>V</i> : Number of reaction; “Haha” emoji <i>W</i> : Number of reaction; “Sad” emoji <i>X</i> : Number of reaction; “Angry” emoji	7050	UCI Machine Learning Repository

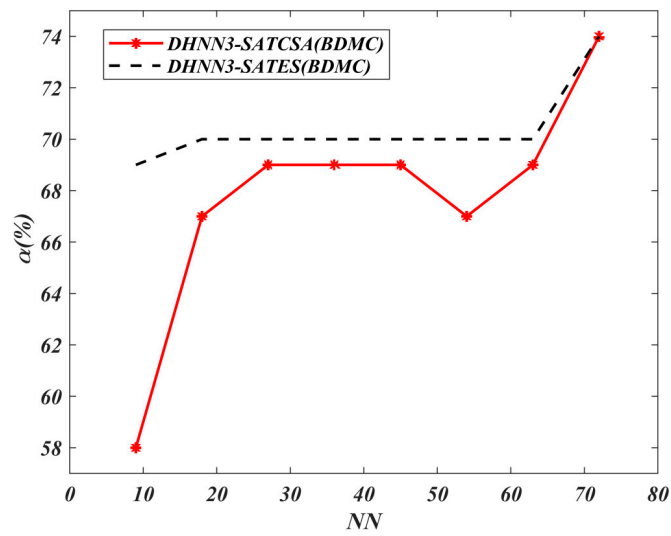


Figure 8. α (%) value of DHNN3-SAT models in the BDMC data set.

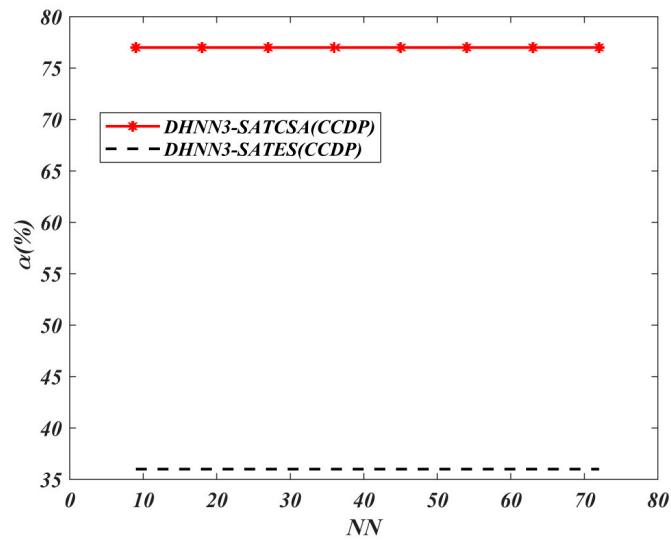


Figure 9. α (%) value of DHNN3-SAT models in the CCDP data set.

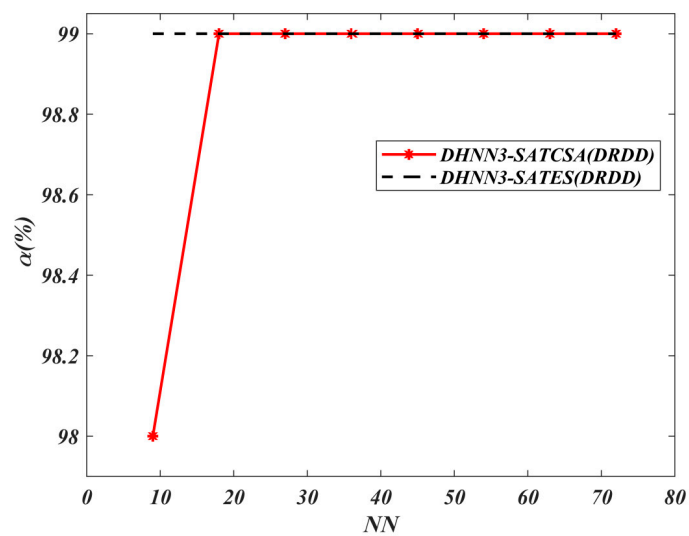


Figure 10. α (%) value of DHNN3-SAT models in the DRDD data set.

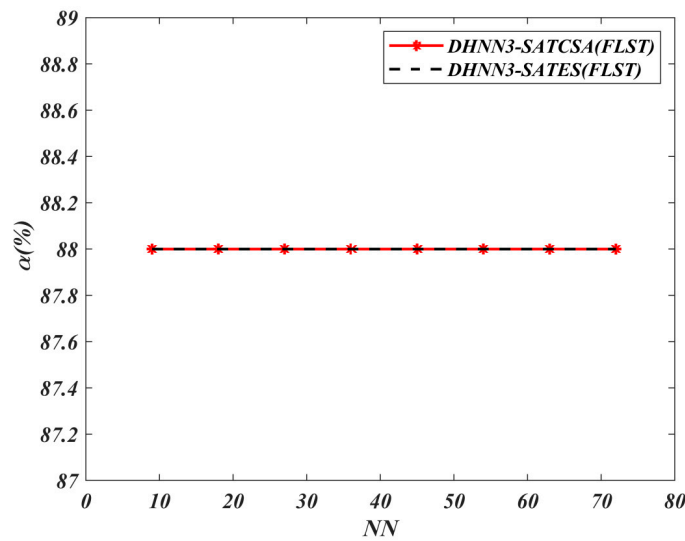


Figure 11. α (%) value of DHNN3-SAT models in the FLST data set.

Table 5 extends the experiment by comparing benchmark data sets with other existing methods which comprises conventional statistical methods such as decision tree (DT), naïve Bayes (NB), support vector machine (SVM). The work of [68] utilized BDMC to predict the successful direct marketing campaign that ensures customers subscribing to a term deposit plan by using DT analysis. Our proposed model achieved better α with differences of 12.73%. On the other hand, the α attained by [69] for CCDP is excessively lower than our proposed model. [69] applied an NB classifier to provide information for risk management of handling customers with credit risks. In [70], this work applies SVM analysis with a confusion matrix to accentuate feature selection and classification. However, the α gained for the SVM method is 25.3% lower than DHNN3-SATCSA. As for FLST, there is no comparable recent work that utilizes this data set.

Table 5. α of DHNN3-SATCSA in comparison with other existing methods.

Data Set	DHNN3-SATCSA	ES	α /Method
BDMC	74%	74%	61.27%/DT [68]
CCDP	77%	36%	66.32%/NB [69]
DRDD	99%	99%	73.7%/SVM [70]
FLST	88%	88%	-

Note that the proposed model does not consider the effect of attribute permutation. This straightforward φ_i^{learn} implementation helps us to effectively determine attributes in the induced logic φ_i^{B} whenever we convert it to other logic programming form. It is worth mentioning that this simulation only considers attributes that lead to $\varphi_{3\text{-SAT}}^{\text{learn}} = 1$, because the proposed model aims to minimize the $E_{\varphi_{3\text{-SAT}}} = 0$. Since there are no redundant attributes in 3SATRA, the satisfiability aspect of φ_i^{B} can be guaranteed. In this case, the structure of φ_i^{learn} should be modified into nonsatisfiability logic such as maximum satisfiability [58]. By that standard, we expected DHNN-3SATCSA will outperform DHNN-3SATES if $E_{\varphi_{3\text{-SAT}}} \neq 0$ is considered in 3-SATRA. In addition, the proposed DHNN-3SAT model does not consider noise function such as in the work of [19]. Thus, the result from this section is important as the φ_i^{B} can be easily analysed by the practitioners as compared to relying entirely on error analysis. Through our findings of simulated and benchmark data sets, we further experiment the competency of DHNN3-SATCSA in entrenching AERA by analysing several performance evaluation metrics.

7.3. Amazon Employees Resources Access Data Set

7.3.1. Performance of DHNN3-SAT in Learning and Testing Phase

From the previous section, we can conclude that the proposed model is suitable to implement in AERA. Therefore, this section investigates the behaviour of DHNN3-SATCSA analysing AERA to help benefit Amazon.com Inc. Both models utilized the 3-SATRA method. However, our main contribution is to investigate CSA capability to enhance the learning mechanism of DHNN. This is to ensure an optimal learning environment. Relative to the experiment, the key findings of the attained φ_{best}^B will also be presented in this section.

In Figures 12 and 13, both the MAE and SSE of DHNN3-SATCSA attained consistent value of errors, approaching to 0, whereas, for DHNN3-SATES, the errors are gradually increasing. Particularly for DHNN3-SATCSA, less accumulation of errors is due to the CSA mechanism improving the quality of solutions in order to attain $E_{\varphi_{3\text{-SAT}}} = 0$. However, ES generated larger value of errors because the ES mechanism is only effective with low NN. We illustrate the capability of the retrieval properties of DHNN3-SAT models based on Figures 14 and 15. Overall, the value of α obtained by DHNN3-SATCSA is relatively higher by at most 3% compared to DHNN3-SATES. Conversely, we also compare the α obtained by other existing work such as [4] that also utilized AERA with conventional statistical methods such as LR, GB and RF. Due to the imbalanced entries of AERA, the work also mentioned their efforts of constructing a prediction model by trying out single models on categorical data. Subsequently, it was improved by introducing various modified methods of decision trees to finally get the desired α . A summary of α achieved by DHNN3-SATCSA with all comparative methods are shown in Table 6. From $NN = 9$ up to $NN = 36$, the CT recorded for both models have a similar rate. However, from $NN = 45$ onwards, DHNN3-SATES required more CT. The apparent reason of why DHNN3-SATES needed more time compared to DHNN3-SATCSA is because the ES mechanism leads to a property of entire bit strings of logical rule collapsing when any of the clauses is not satisfied, thus more iterations are required to produce a plausible solution. That is unlike CSA's ability to minimize iterations in the completion of learning process due to its optimization operator [64].

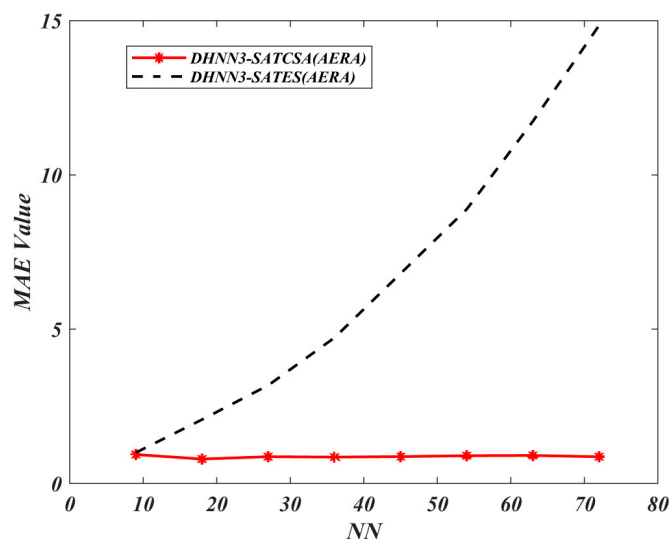


Figure 12. MAE value of DHNN3-SAT models.

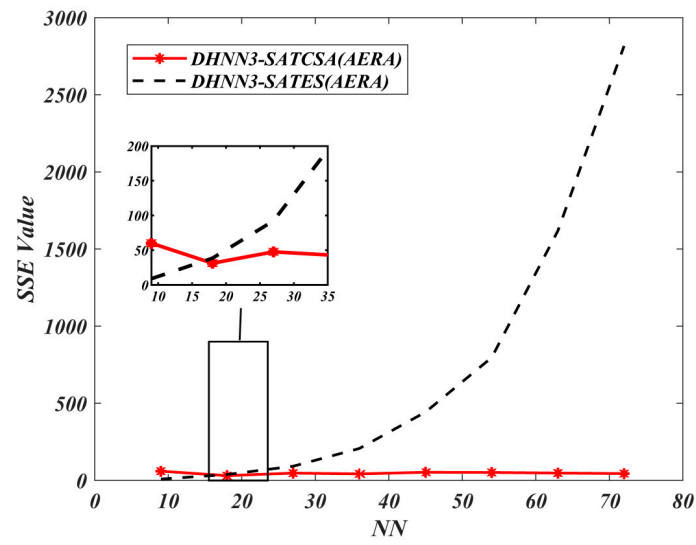


Figure 13. SSE value of DHNN3-SAT models.

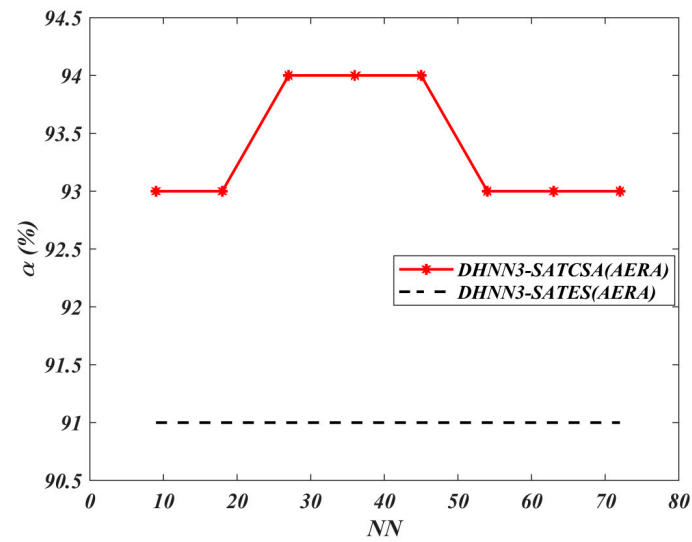


Figure 14. α (%) value of DHNN3-SAT models.

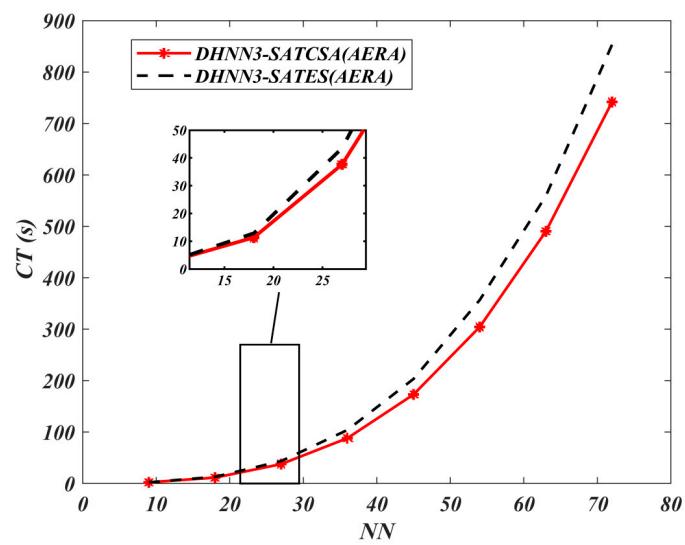


Figure 15. CT value of DHNN3-SAT models.

Table 6. α of DHNN3-SATCSA model in comparison with other existing methods.

Method	α
DHNN3-SATCSA	94%
ES [65]	91%
LR [1]	87.21%
RF [1]	85.58%
GB [1]	85.14%

7.3.2. Key Findings of φ_{best}^B

Equation (26) shows the attained φ_i^B at the highest $\alpha(\varphi_{best}^B)$ by DHNN3-SATCSA. The generated φ_{best}^B will help Amazon.com Inc. in identifying insignificant factors to improve the human resources management. Table 7 shows the details of AERA utilized in this experiment.

$$\varphi_{best}^B = (P \vee \bar{Q} \vee R) \wedge (\bar{S} \vee T \vee \bar{U}) \wedge (\bar{V} \vee W \vee \bar{X}) \tag{26}$$

Table 7. List of information on Amazon Employees Resources Access (AERA) 2010–2011 data set.

Attributes	Example	Instances/Sources
<i>P</i> : An ID for each resources	Types of resources (computer, laptops, software)	32,769/Kaggle Machine Learning and Data Science Community [71]
<i>Q</i> : Manager employee ID	Supervised or not supervised employee	
<i>R</i> : Company role up category ID 1	US Data Analyst	
<i>S</i> : Company role up category ID 2	US Manufacturing	
<i>T</i> : Company role department	Manufacturing	
<i>U</i> : Company role business title	Junior Data Analyst, Senior Manufacturing Staff	
<i>V</i> : Company role family extended description	Security Data Analyst, Product fault detection manufacturing staff	
<i>W</i> : Company role family description	Security Data Analyst	
<i>X</i> : Company role code (unique to each role)	Data Analyst	

Equation (26) gives information of which attributes carry a trivial role in Amazon.com Inc. employees resources applications. We recognized the negation of literals in Equation (26) as a factor that does not affect the problem faced by Amazon.com Inc. For example, \bar{Q} indicates a manager’s unnecessary role to grant resources application. This is believed to add more pointless human administration to solve the employees’ complications regarding their resources. In addition, attributes like *P* will influence the application process as the availability of resources is required to be known for sufficient needs of all employees. *P* also provides resources information to other departments like the operations and maintenance departments to manage defective equipment and appliances. *R* and \bar{S} are correlated. However, it shows clearly different major levels of management, such as top-level, middle-level and first-level are crucial in deciding which resources are first in line. The example of roles related to *R* and \bar{S} are engineers and retailers respectively, thus Equation (26). We can conclude that engineers should be prioritized first compared to retailers as Amazon.com Inc. emerges as a well-known tech giant. Furthermore, Amazon.com Inc. should prioritize *T* to decide which departments are more vital and need new resources to accomplish their tasks in the company. Amazon.com Inc. have to underline certain standards to maintain the quality of work from certain departments that hold a greater role in the norm business of Amazon.com Inc. Consequently, factors like \bar{U} clearly show the insignificant need of considering the business title of an employee in order to grant or revoke their employees resources applications. Top management of Amazon.com

Inc. could improve business personnel in a value-added business to other employees in their business duties. The attributes of \bar{V} and W are related to one another, however, the difference is the specification of an employee role. \bar{V} is the extended version or additional role given to an employee, by referring to the attained φ_{best}^B . We can deduce that Amazon.com Inc. should only consider employees' main role in the company to prioritize the resources applications. Attribute \bar{X} is only essential when Amazon.com Inc. recollects which role is open to vacancy and does not affect much in resources management of Amazon.com Inc.

In line with the no free lunch theorem [72], it is impractical to propose a specific algorithm or model which claims to solve all real-life applications. Thus, new developments on improving metaheuristics and optimization model are continuously needed to handle particular optimization tasks. This work focused on DHNN3-SATCSA transforming AERA into 3-SAT logic representation with 3-SATRA to generate optimum φ_{best}^B to extract information from AERA. On the other hand, [73] reports that the CSA mechanism computational time may take longer because the number of affinity evaluations is increasing as the population of β increases. Nonetheless, the φ_{best}^B attained from DHNN3-SATCSA may help provide Amazon.com Inc. an alternative model to predict resources applications of the future set of employees. Furthermore, DHNN3-SATCSA could be tested by implementing other types of optimization problems from other companies, such as Walmart's efforts to reduce food waste through distribution processes or Ikea's attempts to scale up their system of product fault detection. The implementation of DHNN3-SATCSA will provide beneficial information to a company that wishes to know which factors are more significant than others and which could lead to better control and management of their production.

8. Conclusions

In conclusion, we believe the findings of this study will broaden fundamental optimization methods, such as statistical methods or conventional evolutionary algorithms. In this experiment, the incorporation of 3-SAT in DHNN was crucial to exhibit the relationship and behaviour of AERA symbolically. In addition, 3-SATRA was developed in this study to extract information from AERA, despite its large size with imbalanced entries. Subsequently, 3-SATRA is vital to generate induced logics which displayed insignificant factors in AERA that lead to the problem faced by Amazon.com Inc. Also, the construction of our modified DHNN3-SAT model integrated with modified CSA was revealed to be useful to improve the traditional learning phase of DHNN. In addition, we demonstrated the competency of our hybrid DHNN model of DHNN3-SATCSA by entrenching three different data sets: simulated, benchmark and AERA, in comparison with other existing methods. The comparative investigation was executed by employing various performance evaluation metrics. The findings showed DHNN3-SATCSA outperformed other existing methods. In order to construct a possible model that can cater to all optimization tasks, further improvement of the proposed model could be done to improve the performance and mechanism of the model by implementing a mutation feature in the testing phase of DHNN. Therefore, the exploration of the testing phase in DHNN is worthy of attention, alongside future research addressing the variability of implementing other algorithms to enhance the mechanism of modified DHNN models.

Author Contributions: Conceptualization, methodology, resources, A.A.; validation, writing—original draft preparation, project administration, N.E.Z.; formal analysis, M.S.M.K.; writing—review and editing, S.Z.M.J., visualization, S.A.A.; funding acquisition, M.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Fundamental Research Grant Scheme (FRGS), Ministry of Education Malaysia, grant number 203/PJJAUH/6711751 and Universiti Sains Malaysia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Clement, J. Number of Amazon.com Employees from 2007 to 2019. Available online: <https://www.statista.com/statistics/234488/number-of-amazon-employees/> (accessed on 9 April 2020).
2. Marques, P.H.; Atouguia, J.; Marques, F.H.; Palhais, C.; Pinto, A.R.; Silva, L.A. (Eds.) A study of associations of occupational accidents to number of employees, and to hours worked. In *Occupational Safety and Hygiene II*; CRC: London, UK, 2014; pp. 663–667.
3. Gupta, A.K.; Singhal, A. Managing human resources for innovation and creativity. *Res.-Technol. Manag.* **1993**, *36*, 41–48. [[CrossRef](#)]
4. Tang, S.; Han, J.B.; Zhang, Y. Amazon Employee Access Control System. Available online: <https://www.semanticscholar.org/paper/Amazon-Employee-Access-Control-System-Tang-Han/e419470b5d9808a8a8bfee41f4e7d8ffff15eb53> (accessed on 27 January 2020).
5. Gonzalez-Fernandez, I.; Iglesias-Otero, M.A.; Esteki, M.; Moldes, O.A.; Mejuto, J.C.; Simal-Gandara, J. A critical review on the use of artificial neural networks in olive oil production, characterization and authentication. *Crit. Rev. Food Sci. Nutr.* **2019**, *59*, 1913–1926. [[CrossRef](#)]
6. Parsaeimehr, E.; Fartash, M.; Torkestani, J.A. An Enhanced Deep Neural Network-Based Architecture for Joint Extraction of Entity Mentions and Relations. *Int. J. Fuzzy Log. Intell. Syst.* **2020**, *20*, 69–76. [[CrossRef](#)]
7. Sameen, M.I.; Pradhan, B.; Lee, S. Application of convolutional neural networks featuring Bayesian optimization for landslide susceptibility assessment. *Catena* **2020**, *186*, 104249. [[CrossRef](#)]
8. Su, L.; Deng, L.; Zhu, W.; Zhao, S. Statistical detection of weak pulse signal under chaotic noise based on Elman neural network. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 9653586. [[CrossRef](#)]
9. Mansor, M.A.; Kasihmuddin, M.S.M.; Sathasivam, S. Enhanced Hopfield network for pattern satisfiability optimization. *Int. J. Intell. Syst. Appl.* **2016**, *8*, 27. [[CrossRef](#)]
10. Fu, Y.; Aldrich, C. Flotation froth image recognition with convolutional neural networks. *Miner. Eng.* **2019**, *132*, 183–190. [[CrossRef](#)]
11. Hou, J.; He, Y.; Yang, H.; Connor, T.; Gao, J.; Wang, Y.; Zeng, Y.; Zhang, J.; Huang, J.; Zheng, B.; et al. Identification of animal individuals using deep learning: A case study of giant panda. *Biol. Conserv.* **2020**, *242*, 108414. [[CrossRef](#)]
12. Benitez-Garcia, G.; Haris, M.; Tsuda, Y.; Ukita, N. Finger gesture spotting from long sequences based on multi-stream recurrent neural networks. *Sensors* **2020**, *20*, 528. [[CrossRef](#)] [[PubMed](#)]
13. Choi, H.S.; Park, Y.J.; Lee, J.H.; Kim, Y. 3-D synapse array architecture based on charge-trap flash memory for neuromorphic application. *Electronics* **2020**, *9*, 57. [[CrossRef](#)]
14. Yang, C.; Kim, H.; Adhikari, S.P.; Chua, L.O. A circuit-based neural network with hybrid learning of backpropagation and random weight change algorithms. *Sensors* **2017**, *17*, 16. [[CrossRef](#)] [[PubMed](#)]
15. Hopfield, J.J.; Tank, D.W. “Neural” computation of decisions in optimization problems. *Biol. Cybern.* **1985**, *52*, 141–152. [[PubMed](#)]
16. Duan, L.; Duan, F.; Chapeau-Blondeau, F.; Abbott, D. Stochastic resonance in Hopfield neural networks for transmitting binary signals. *Phys. Lett. A* **2020**, *384*, 126143. [[CrossRef](#)]
17. Kong, D.; Hu, S.; Wang, J.; Liu, Z.; Chen, T.; Yu, Q.; Liu, Y. Study of recall time of associative memory in a memristive Hopfield neural network. *IEEE Access* **2019**, *7*, 58876–58882. [[CrossRef](#)]
18. Kasihmuddin, M.S.M.; Mansor, M.A.; Sathasivam, S. Discrete Hopfield neural network in restricted maximum k-satisfiability logic programming. *Sains Malays.* **2018**, *47*, 1327–1335. [[CrossRef](#)]
19. Kasihmuddin, M.S.M.; Mansor, M.A.; Basir, M.F.M.; Sathasivam, S. Discrete mutation Hopfield neural network in propositional satisfiability. *Mathematics* **2018**, *7*, 1133. [[CrossRef](#)]
20. Abdullah, W.A.T.W. The logic of neural networks. *Phys. Lett. A* **1993**, *176*, 202–206. [[CrossRef](#)]
21. Soeken, M.; Meuli, G.; Schmitt, B.; Mozafari, F.; Riener, H.; De Micheli, G. Boolean satisfiability in quantum compilation. *Philos. Trans. R. Soc. A* **2020**, *378*, 20190161. [[CrossRef](#)]
22. Gebregiorgis, A.; Tahoori, M.B. Test Pattern Generation for Approximate Circuits Based on Boolean Satisfiability. In Proceedings of the 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 25–29 March 2019; pp. 1028–1033.
23. Hireche, C.; Drias, H.; Moulai, H. Grid based clustering for satisfiability solving. *Appl. Soft Comput.* **2020**, *88*, 106069. [[CrossRef](#)]

24. Yamashita, H.; Aihara, K.; Suzuki, H. Timescales of Boolean satisfiability solver using continuous-time dynamical system. *Commun. Nonlinear Sci. Numer. Simul.* **2020**, *84*, 105183. [[CrossRef](#)]
25. Weissenbacher, G.; Malik, S. Post-silicon fault localization with satisfiability solvers. In *Post-Silicon Validation and Debug*; Mishra, P., Farahmandi, F., Eds.; Springer: Cham, Switzerland, 2019; pp. 255–273.
26. Abdullah, W.A.T.W. Logic programming on a neural network. *Int. J. Intell. Syst.* **1992**, *7*, 513–519. [[CrossRef](#)]
27. Sathasivam, S. First order logic in neuro-symbolic integration. *Far East J. Math. Sci.* **2012**, *61*, 213–229.
28. Čeppek, O.; Kučera, P. Known and new classes of generalized Horn formulae with polynomial recognition and SAT testing. *Discret. Appl. Math.* **2005**, *149*, 14–52. [[CrossRef](#)]
29. Kasihmuddin, M.S.M.B.; Mansor, M.A.B.; Sathasivam, S. Genetic algorithm for restricted maximum k-Satisfiability in the Hopfield network. *Int. J. Interact. Multimed. Artif. Intell.* **2016**, *4*, 52–60.
30. Kasihmuddin, M.S.M.; Mansor, M.A.; Sathasivam, S. Hybrid genetic algorithm in the Hopfield network for logic satisfiability problem. *Pertanika J. Sci. Technol.* **2017**, *25*, 139–152.
31. Kasihmuddin, M.S.M.; Sathasivam, S.; Mansor, M.A. Hybrid Genetic Algorithm in the Hopfield Network for Maximum 2-Satisfiability Problem. In Proceedings of the 24th National Symposium of Mathematical Sciences: Mathematical Sciences Exploration for the Universal Preservation, Kuala Terengganu, Malaysia, 27–29 September 2016; p. 050001.
32. Sathasivam, S.; Abdullah, W.A.T.W. Logic mining in neural network: Reverse analysis method. *Computing* **2011**, *91*, 119–133. [[CrossRef](#)]
33. Sathasivam, S. Application of Neural Networks in Predictive Data Mining. In Proceedings of the 2nd International Conference on Business and Economic, Kedah, Malaysia, 14–16 March 2011; pp. 371–376.
34. Kasihmuddin, M.S.M.; Mansor, M.A.; Sathasivam, S. Students' Performance via Satisfiability Reverse Analysis Method with Hopfield Neural Network. In Proceedings of the International Conference on Mathematical Sciences and Technology 2018 (MATHTECH2018): Innovative Technologies for Mathematics and Mathematics for Technological Innovation, Penang, Malaysia, 10–12 December 2018; p. 060035.
35. Kho, L.C.; Kasihmuddin, M.S.M.; Asyraf, M. Logic mining in league of legends. *Pertanika J. Sci. Technol.* **2020**, *28*, 211–225.
36. Kho, L.C.; Kasihmuddin, M.S.M.; Mansor, M.A.; Sathasivam, S. Logic mining in football matches. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *17*, 1074–1083. [[CrossRef](#)]
37. Gendreau, M.; Potvin, J.Y. *Handbook of Metaheuristics*, 2nd ed.; Springer: New York, NY, USA, 2010.
38. Göçken, M.; Özçalıcı, M.; Boru, A.; Dosdoğru, A.T. Integrating metaheuristics and artificial neural networks for improved stock price prediction. *Expert Syst. Appl.* **2016**, *44*, 320–331. [[CrossRef](#)]
39. Nievergelt, J. Exhaustive Search, Combinatorial Optimization and Enumeration: Exploring the Potential of Raw Computing Power. In Proceedings of the 27th Conference on Current Trends in Theory and Practice of Informatics, Milovy, Czech Republic, 25 November–2 December 2000; pp. 18–35.
40. Reunanen, J. Overfitting in making comparisons between variable selection methods. *J. Mach. Learn. Res.* **2003**, *3*, 1371–1382.
41. Hooker, J.N. Unifying Local and Exhaustive Search. In Proceedings of the ENC 2005—Sixth Mexican International Conference on Computer Science, Puebla, Mexico, 26–30 September 2005; pp. 237–243.
42. Luo, W.; Lin, X. Recent Advances in Clonal Selection Algorithms and Applications. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–8.
43. Mohammed, M.A. Compare between genetic algorithm and clonal selection algorithm to pattern recognition Latin's numbers. *J. Educ. Sci.* **2019**, *28*, 300–315. [[CrossRef](#)]
44. Dasgupta, K.; Roy, P.K.; Mukherjee, V. Power flow based hydro-thermal-wind scheduling of hybrid power system using sine cosine algorithm. *Electr. Power Syst. Res.* **2020**, *178*, 106018. [[CrossRef](#)]
45. Silva, G.C.; Carvalho, E.E.; Caminhas, W.M. An artificial immune systems approach to case-based reasoning applied to fault detection and diagnosis. *Expert Syst. Appl.* **2020**, *140*, 112906. [[CrossRef](#)]
46. Zhang, W.; Zhang, W.; Yen, G.G.; Jing, H. A cluster-based clonal selection algorithm for optimization in dynamic environment. *Swarm Evol. Comput.* **2019**, *50*, 100454. [[CrossRef](#)]
47. Layeb, A. A clonal selection algorithm based tabu search for satisfiability problems. *J. Adv. Inf. Technol.* **2012**, *3*, 138–146. [[CrossRef](#)]
48. Cook, S.A. The Complexity of Theorem-Proving Procedures. In Proceedings of the STOC '71: Third Annual ACM Symposium on Theory of Computing, Shaker Heights, OH, USA, 3–5 May 1971; pp. 151–158.

49. Mansor, M.A.; Kasihmuddin, M.S.M.; Sathasivam, S. Modified artificial immune system algorithm with Elliot Hopfield neural network for 3-satisfiability programming. *J. Inform. Math. Sci.* **2019**, *11*, 81–98.
50. Fitzsimmons, M.; Kunze, H. Combining Hopfield neural networks, with applications to grid-based mathematics puzzles. *Neural Netw.* **2019**, *118*, 81–89. [[CrossRef](#)]
51. Katayama, K.; Horiguchi, T. Generalization ability of Hopfield neural network with spin-S Ising neurons. *J. Phys. Soc. Jpn.* **2000**, *69*, 2816–2824. [[CrossRef](#)]
52. Sathasivam, S. Upgrading logic programming in Hopfield network. *Sains Malays.* **2010**, *39*, 115–118.
53. Velavan, M.; Yahya, Z.R.B.; Halif, M.N.B.A.; Sathasivam, S. Mean field theory in doing logic programming using Hopfield network. *Mod. Appl. Sci.* **2016**, *10*, 154. [[CrossRef](#)]
54. Kasihmuddin, M.S.B.M.; Sathasivam, S. Accelerating Activation Function in Higher Order Logic Programming. In Proceedings of the 23rd National Symposium of Mathematical Sciences (SKSM23), Johor Bahru, Malaysia, 24–26 November 2015; p. 030006.
55. Moreno-Armendáriz, M.A.; Hagan, M.; Alba, E.; Rubio, J.D.J.; Cruz-Villar, C.A.; Leguizamón, G. Advances in neural networks and hybrid-metaheuristics: Theory, algorithms, and novel engineering applications. *Comput. Intell. Neurosci.* **2016**, *2016*, 3263612. [[CrossRef](#)] [[PubMed](#)]
56. Rojas-Delgado, J.; Trujillo-Rasúa, R.; Bello, R. A continuation approach for training artificial neural networks with meta-heuristics. *Pattern Recognit. Lett.* **2019**, *125*, 373–380. [[CrossRef](#)]
57. Castro, L.N.D.; Zuben, F.J.V. The Clonal Selection Algorithm with Engineering Applications. In Proceedings of the GECCO, Las Vegas, NV, USA, 8–12 July 2000; pp. 36–37.
58. Mansor, M.A.B.; Kasihmuddin, M.S.B.M.; Sathasivam, S. Robust artificial immune system in the Hopfield network for maximum k-satisfiability. *IJIMAI* **2017**, *4*, 63–71. [[CrossRef](#)]
59. Goldberg, D.E.; Deb, K. A comparative analysis of selection schemes used in genetic algorithms. *Found. Genet. Algorithms* **1991**, *1*, 69–93.
60. Sathasivam, S.; Mamat, M.; Mansor, M.A.; Kasihmuddin, M.S.M. Hybrid discrete Hopfield neural network based modified clonal selection algorithm for VLSI circuit verification. *Pertanika J. Sci. Technol.* **2020**, *28*, 227–243.
61. Ong, P.; Zainuddin, Z. Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction. *Appl. Soft Comput.* **2019**, *80*, 374–386. [[CrossRef](#)]
62. Kathirvel, V.; Mansor, M.A.; Kasihmuddin, M.S.M.; Sathasivam, S. Hybrid imperialistic competitive algorithm incorporated with hopfield neural network for robust 3 satisfiability logic programming. *IAES Int. J. Artif. Intell.* **2019**, *8*, 144. [[CrossRef](#)]
63. Wang, X.; Wang, J.; Fečkan, M. BP neural network calculus in economic growth modelling of the group of seven. *Mathematics* **2020**, *8*, 37. [[CrossRef](#)]
64. Mansor, M.A.; Kasihmuddin, M.S.M.; Sathasivam, S. Artificial immune system paradigm in the Hopfield network for 3-satisfiability problem. *Pertanika J. Sci. Technol.* **2017**, *25*, 1173–1188.
65. Sathasivam, S. Learning rules comparison in neuro-symbolic integration. *Int. J. Appl. Phys. Math.* **2011**, *1*, 129. [[CrossRef](#)]
66. Sathasivam, S. Clauses Representation Comparison in Neuro-Symbolic Integration. In Proceedings of the World Congress on Engineering 2010 Vol 1 (WCE 2010), London, UK, 30 June–2 July 2010; pp. 34–37.
67. Lichman, M. UCI Machine Learning Repository: University of California, School of Information and Computer Science. Available online: <http://archive.ics.uci.edu/ml> (accessed on 9 April 2020).
68. Rogic, S.; Kascelan, L. Customer Value Prediction in Direct Marketing Using Hybrid Support Vector Machine Rule Extraction Method. In Proceedings of the European Conference on Advances in Databases and Information Systems (ADBIS 2019), Bled, Slovenia, 8–11 September 2019; Springer: Cham, Switzerland, 2019; pp. 283–294.
69. Singh, B.E.R.; Sivasankar, E. Enhancing Prediction Accuracy of Default of Credit Using Ensemble Techniques. In Proceedings of the First International Conference on Artificial Intelligence and Cognitive Computing (AICC 2018), London, UK, 7 December 2018; Springer: Singapore, 2019; pp. 427–436.
70. Oladele, T.O.; Ogundokun, R.O.; Kayode, A.A.; Adegun, A.A.; Adebisi, M.O. Application of Data Mining Algorithms for Feature Selection and Prediction of Diabetic Retinopathy. In Proceedings of the International Conference on Computational Science and Its Application (ICCSA) 2019, Saint Petersburg, Russia, 1–4 July 2019; Springer: Cham, Switzerland, 2019; pp. 716–730.

71. Amazon.com Employee Access Challenge. Available online: <https://www.kaggle.com/c/amazon-employee-access-challenge> (accessed on 9 April 2020).
72. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE* **1997**, *1*, 67–82. [[CrossRef](#)]
73. Wang, Y.; Li, T. Local feature selection based on artificial immune system for classification. *Appl. Soft Comput.* **2020**, *87*, 105989. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).