Franklin Open 16 (2026) 100678

Contents lists available at ScienceDirect

Franklin Open

journal homepage: www.elsevier.com/locate/fraope

# Flexible logic mining model via conditional random 2 satisfiability in discrete Hopfield neural network

Nurshazneem Roslan [a], Saratha Sathasivam [b], Nur Ezlin Zamri [c,*]

[a] Department of Mathematical Sciences, Faculty of Intelligent Computing, Universiti Malaysia Perlis, Kampus Alam UniMAP, Pauh Putra, 02600 Arau, Perlis, Malaysia
[b] School of Mathematical Sciences, Universiti Sains Malaysia, 11800, USM, Penang, Malaysia
[c] Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

ARTICLE INFO

*Keywords:*
Conditional random 2 satisfiability
Non-systematic logic
Multi-objective optimization
Hybrid whale optimization
Multi-unit logic mining

ABSTRACT

The study of logic mining has gained significant attention for its ability to interpret behavior of the datasets. However, existing logic mining models face limitations such as reliance on single optimal logic, insufficient focus in attribute selection methods and a tendency to overfit especially when dealing with imbalanced datasets. To address these challenges, this paper introduces a versatile and flexible logic mining model based on Conditional Random 2 Satisfiability. The proposed model is structured into three main phases: pre-processing, learning and retrieval. By utilizing a multi-objective retrieval phase, the model improves its classification performance by retrieving a final neuron state that is optimally diversified. A different pathway in generating the best logic expands the search space leading to the production of a set of induced logics using different confusion matrixes. As a result, the logic mining model moves beyond a rigid dependency on the single best logic and thus offers flexibility in interpreting the behavior of the datasets. Additionally, the similarity analysis enhances the attribute selection method by calculating the distance between attributes and dataset outcomes. This evaluation considers the distribution of both positive and negative entries across independent and dependent attributes. Comparative experiments conducted on real-world datasets demonstrate the superiority of the proposed logic mining model, achieving an average $ACC = 0.8275$, $PRE = 0.9116$, $SPE = 0.9994$, $MCC = 0.5221$ and $MCR = 0.1758$. These results consistently outperform baseline logic mining models across all evaluation metrics.

## 1. Introduction

Artificial Intelligence (AI) focuses on creating systems capable of performing tasks with human-like intelligence. Artificial Neural Networks (ANNs) is a subset of AI which are inspired by the human brain and are designed to process data, learn from it and use the acquired knowledge to make decisions. Due to their adaptability and learning capabilities, ANNs are widely used across various fields including speech recognition [1], medical natural language processing [2], medical imaging [3] and financial forecasting [4]. Building on the strengths of ANNs, researchers have started combining symbolic logic with computational intelligence to tackle classification challenges. These challenges often involve handling complex datasets with different patterns, which makes tools like logic mining valuable for improving classification, accuracy and interpretability. This approach has opened new opportunities for developing models that are not only effective but also interpretable.

Among various types of ANNs, the Discrete Hopfield Neural Network (DHNN) is widely used for optimization problems. However, as a black box model, DHNN has limitations in terms of output interpretability. One of its well-known applications is the Travelling Salesman Problem (TSP) [5], which is historically regarded as a classical benchmark due to its suitability for combinatorial optimization. Although this demonstrates the effectiveness of DHNN in structured optimization tasks, its applications are primarily restricted to numerical optimization with limited interpretability. Therefore, implementing the symbolic logic into the DHNN model can enhance its interpretability by providing a deeper understanding of intelligence and a clearer interpretation of the output. According to [6], symbolic logic plays a key role in DHNN as it governs the neuron representation. In his early study, he proposed embedding satisfiability logical rules into DHNN and the value of the synaptic weight is obtained by comparing the cost function with the Lyapunov energy function. This approach is referred to as the Wan Abdullah method (WA method), which ensures the convergence of the network

* Corresponding author.
*E-mail addresses:* nurshazneem@unimap.edu.my (N. Roslan), saratha@usm.my (S. Sathasivam), ezlinzamri@upm.edu.my (N.E. Zamri).

https://doi.org/10.1016/j.fraope.2026.100678
Received 7 October 2025; Received in revised form 24 March 2026; Accepted 11 June 2026
Available online 15 June 2026

and eventually guarantees the global solution. Subsequent research such as [7] and [8] explored different logical rules like 2 Satisfiability (2SAT) and 3 Satisfiability (3SAT) to enhance neuron interpretability. The structure of 2SAT contains two literals per clause and achieved a high global minima ratio. While 3SAT with three literals per clause faced challenges due to exponential growth of associative memory in DHNN.

Recent advancements in DHNN have expanded the type of logical rules into non-systematic logical rules. Sathasivam et al. [9] introduced Random 2 Satisfiability (RAN2SAT) which combines first-order and second-order logic allowing for a flexible and dynamic number of literals per clause. However, experimental results showed that first-order clauses introduced more logical inconsistencies compared to second-order clauses. This limits the ability of DHNN to complete the learning phase and thus lead to a suboptimal retrieval phase. To address these challenges, [10] proposed Y-Type Random 2 Satisfiability (YRAN2SAT) a hybrid logical rule offering five flexible pathways. YRAN2SAT demonstrated its ability to optimize synaptic weights and increase neuron state variability across solution spaces. Despite these advances, the distribution of the negative literal in the logical rules remains random. Therefore, [11] introduced Weighted Random 2 Satisfiability (*r*2SAT) to control the distribution of negative literals through a logic phase that generates non-systematic structures based on a specified number of negative literals. However, this approach increased the complexity of the existing DHNN [12]. To address this, [12] proposed Weighted Systematic 2 Satisfiability (*r*S2SAT), which controls the distribution of negative literals, promotes diversification and encourages the learning of negative neuron connections. Meanwhile, [13] emphasized the high appearance of negative literals in their Negative-Based Higher-Order Systematic Satisfiability (NR3SAT), but their approach is limited to systematic logic. Further research is needed to explore non-systematic logical rules, particularly to understand the significance of the high appearance of negative literal without relying on an additional logic phase to strategically position these literals.

The study of DHNN in the field of AI has gained significant interest due to its unique capabilities including logic mining which extract logical rules and patterns [14] define logic mining as a subset of data mining that analyzes the behavior of dataset through logical rules, producing SAT-induced logic to enhance classification and decision-making. Logic mining relies on three key components: the computational power of ANNs, symbolic SAT rules and reverse analysis (RA) methods [15] pioneered this field by proposing RA method using HORNSAT logic to extract logical rules representing student performance. However, progress in identifying the most optimal induced logic remains limited. Building on earlier work, [16] and [17] introduced the 2 Satisfiability based Reverse Analysis method (2SATRA) and 3 Satisfiability based Reverse Analysis method (3SATRA), respectively. Furthermore, [18] developed the Energy-based 2 Satisfiability Reverse Analysis (E2SATRA) which uses an energy operator to transform final neuron states with global minimum energy into induced logic. A common issue across logic mining models is the random attribute selection, which reduces the interpretability of induced logic since the selected attributes may not represent the overall behavior of the dataset. To address this, [19]proposed 2 Satisfiability-based Reverse Analysis with a Permutation Operator (P2SATRA), introducing a logical permutation operator to improve compatibility, accuracy, and competitiveness over conventional 2SATRA. However, their approach focused solely on permutation without incorporating a specific method for attribute selection.

Previous logic mining models often rely on random attribute selection methods, which can lead to unreliable and insignificant patterns in datasets. To address this issue, several studies have focused on improving attribute selection methods. For example, [20] introduced the Supervised Logic Mining model with Correlation analysis (S2SATRA) which uses correlation to identify the best attributes for dataset representation. By eliminating unnecessary attributes before embedding them into DHNN, S2SATRA achieved superior accuracy and precision compared to existing models. Furthermore, [21] proposed the Supervised Logic Mining model with Log-linear analysis (A2SATRA) to select optimal attributes. However, supervised methods do not account for similarity between attributes and logical outcomes. To overcome this, [14] employed the Jaccard similarity index for feature selection. This method is effective but only considers positive entries and may overlook significant relationships in negative entries. Hence, incorporating both positive and negative entries is essential for a more comprehensive understanding of dataset relationships. Another issue with logic mining models is their reliance on generating the best logic based solely on true positive outcomes. This approach often prioritizes clauses with the highest frequency of positive results which lead to biased solutions particularly for imbalanced datasets.

The best logic refers to the optimal logic used to be trained by the network to learn the behaviour of the dataset [22] and [23] introduced multi-unit logic mining models that select the best logic based on true positive and true negative outcomes which outperform existing logic mining models. However, these models still face limitations during the learning phase, as they rely on trial-and-error algorithms which remain suboptimal learning phase. Current logic mining models such 2 Hybrid Exhaustive Search Reverse-based Analysis (2HESRA) by [24] and Weighted Random 2 Satisfiability Modified Reverse-based Analysis (*r*2SATMRA) by [14] include modifications in the learning phase by utilizing multi-objective learning algorithms and multiple CAMs to improve the quality of CAM. However, these models primarily focus on maximizing true positive and true negative outcomes to generate the best logic and produce a single induced logic for knowledge extraction. To address these limitations, [12]introducing the Weighted Systematic 2 Satisfiability based Reverse Analysis Method (U2SATRA) which provides an alternative pathway for generating the best logic. U2SATRA learns the behaviors of the dataset from various perspectives and generates different sets of induced logics that consequently offer flexibility in doing classification tasks. The effectiveness of U2SATRA is further enhanced by an optimized learning phase and a multi-objective retrieval phase. Despite these advancements, U2SATRA remains most effective in the context of systematic logic and has limited capacity to independently validate patterns of the datasets. In contrast, [25] introduced a new flexible logic mining model. However, the evaluation of best logic focuses primarily on maximizing true positive and true negative outcomes.

Motivated by the identified limitations and informed by insights from the prior literature review, the following research questions are proposed for further investigation.

1. What is an alternative method for attribute selection that can effectively measure the relationship between positive and negative distributions?
2. How does the implementation of a multi-objective function impact the quality of the final neuron state?
3. What approach can be implemented into the current logic mining model to ensure that a set of induced logics obtained from different confusion matrixes is effective in doing classification tasks?
4. How effectively does the proposed logic mining model generate a set of induced logics and perform on benchmark datasets based on various performance metrics?

In response to these research questions, we propose a versatile logic mining model that offers flexibility in performing classification tasks and extracting knowledge from various datasets domains. This proposed model is based on a new method for attribute selection and implementing multi-objective optimization during the retrieval phase. We also introduce different pathways for generating the best logic and produces a set of induced logic based on different confusion matrixes that accurately represent classification outcomes for both positive and negative classes. Thus, the main contributions of this work are outlined as follows:

To propose a new method for attribute selection by introducing Rogers Tanimoto Feature Selection Method. This method utilizes similarity analysis to compute the distance between attributes and dataset outcomes which focus on the distribution of both negative and positive states.
To enhance the retrieval phase by implementing a multi-objective function for the Hybrid Binary Whale Optimization Algorithm. This approach aims to optimize the final neuron states with aims of retrieved diversified and optimal final neuron states with more negative states which will be good influenced in the context of logic mining.
To propose a multi-unit Conditional Random 2 Satisfiability Reverse Analysis method which consider five different confusion matrixes in generating the best logic and produce a set on induced logics capable of performing classification and knowledge extraction from the datasets.
To evaluate the performance of the proposed logic mining model and analyze the significance of generated induced logics using the Friedman Test.

This paper is structured as follows: Section 2 discusses the motivation and related work that support the contributions outlined above. Section 3 explains the logical structure which acts as the neuron representation in DHNN and forms the main component of the logic mining model. Sections 4 and 5 provide a detailed description of the proposed methodology. Section 6 outlines the experimental setup, including the repository datasets used in the experiments. Section 7 discusses the findings from all logic mining models and Section 8 concludes the paper by summarizing the key findings.

## 2. Motivation and related work

The current literature highlights several limitations, creating opportunities for the development of a new logic mining model. Based on the identified motivations and related works, this paper addresses key challenges in current research to develop an effective logic mining model that captures significant relationships between attributes and outcomes.

### 2.1. Lack of understanding significance of the negative literal in non-systematic logic

According to [26], SAT tends to produce positive literals which limits the appearance of negative literals in the logical rule. Although various types of non-systematic logic have been proposed such as RAN2SAT [9] and YRAN2SAT [10], the significance of a high appearance of negative literals within the second-order clauses remains uncertain. This uncertainty arises from the random generation of positive and negative literals in logical rules. Some studies, such as *r*2SAT [11] and S-TypeRAN2SAT [26] focus on distributing negative literals. For instance, *r*2SAT uses a metaheuristic algorithm to distribute negative literals, but this increases the complexity of DHNN [12]. S-TypeRAN2SAT employs a probabilistic logic phase to position positive and negative literals within clauses. Despite these efforts, the significance of having a high appearance of negative literals within the second-order clauses remains uncertain and requires further investigation. Therefore, it is essential to implement a mechanism that ensure the appearance of at least one negative literal in second-order clauses. In this paper, we introduce restricted conditions within non-systematic second-order logic to closely examine the role of negative literals.

### 2.2. Limitations in diversifying negative final neuron state

During the retrieval phase, the final neuron state generated by the DHNN is evaluated to determine whether the solution is a global or local minima solution. The retrieved final neuron state is updated asynchronously using the synaptic weights stored in the CAM. However, most studies overlooked the quality assessment of the final neuron state because the primary focus is typically on reaching the final state [27] enhanced the retrieval phase using the Estimation of Distribution Algorithm (EDA). However, as the number of neurons increases, the final neuron state is trapped in suboptimal states due to the use of Exhaustive Search (ES) during the learning phase of DHNN. Additionally, [27] has concentrated solely on the quality of the final neuron state without considering the variability in negative states. Although [28] improved the updating rule of DHNN by implementing Smish activation function, but the impact on the variability of negative state remains unclear. Similarly, although [13] proposed multi-objective functions during the retrieval phase, their approach was limited to systematic logic. In this paper, we propose utilizing the Election Algorithm (EA) from Sathasivam et al. (2020) to achieve an optimal learning phase. The quality of the retrieved final neuron state will be optimized to improve diversification and achieve optimal states with a specific focus on increasing negative states. This enhancement will significantly benefit the field of logic mining.

### 2.3. Limitation of non-systematic logic mining in generating the best logic

In logic mining models, generating the best logic is crucial because these logical rules are trained by the network to extract the behaviour of the datasets. Additionally, the best logic also represents the optimal connections between neurons with the correct synaptic weight. In previous logic mining models [20,21] the best logic was typically based on clauses with the highest frequency of positive outcomes in the learning data. However, this approach can lead to bias solutions especially when dealing with imbalanced datasets. To address these issues, some researchers have introduced new approaches that consider the summation of both true positive and true negative classifications when generating the best logic [14,24]. However, relying on single induced logic may limit the search space for finding an optimal best logic that fully represents the behaviour of the dataset. To overcome this limitation, [23] and [22] proposed a multi-unit DHNN logic mining model that generates ten best logics to be learned by the network. However, their logic mining model only focuses on maximizing the summation of both true positive and true negative classifications. Since most datasets suffer from imbalances, [12] proposed generating multiple best logics that consider both true and false classifications to obtain significant induced logic that accurately represents the overall behavior of the dataset. However, the proposed logic mining model by [12] is based on systematic logic. In this paper, we propose a different pathway for generating the best logic based on non-systematic logic. This represents the first attempt to implement non-systematic logic as a rule for logic mining.

### 2.4. Challenges in generating induced logic with high similarity between attributes and outcome

Induced logic is crucial in logic mining where it is used to extract patterns from real-life datasets particularly for classification tasks. While most previous research has focused on producing a single best-induced logic, [12] attempted to produce a set of induced logics. However, the interpretability of these logics has been questioned due to a lack of consideration for the similarity between attributes and outcomes. Despite retrieving a set of induced logics through various metrics, their interpretability remains limited due to weak relationships between attributes and the outcomes. While generating a set of induced logic is essential, optimal attribute selection is equally important for ensuring relevant classification tasks. Existing logic mining models use both supervised and unsupervised attributes selection methods [12,14,20,21], but these methods often overlook the relationship between positive and negative attributes and the outcome thus limit the comprehensiveness of the relationships. Even though logic mining models can produce sets of induced logic, selecting optimal attributes that accurately represent the

behavior of the dataset remains crucial. In this paper, we propose using similarity analysis for attribute selection to ensure that only significant attributes with strong relationships to the outcome are selected. This approach helps eliminate irrelevant attributes and extract relevant patterns, making it particularly suitable for complex and imbalanced real-life datasets.

Based on these identified limitations, this paper proposes a logic mining model designed to address the limitations of existing approaches. The proposed framework focuses on improving the representation of negative literals, enhancing the quality of retrieved neuron states through multi-objective optimization, and enabling more flexible logic generation using multiple evaluation criteria. In addition, a similarity-based attribute selection strategy is incorporated to ensure more relevant feature representation. These improvements collectively enhance the effectiveness of the model, particularly for imbalanced and real-world datasets.

## 3. Conditional random 2 satisfiability into DHNN

This section provides the theoretical background of all the elements involved in the proposed logic mining, including the formulation of non-systematic logic, the implementation of non-systematic logic into DHNN as well as the utilization of multi-objective function that optimize the retrieved final neuron state. First, this section discusses the formulation of the non-systematic structure of Conditional Random 2 Satisfiability (CRAN2SAT). The basic components of the proposed CRAN2SAT are as follows:

(a) A set of $m$ second-order clauses where each clause contains exactly two literals is represented as $C_1^{(2)}$, $C_2^{(2)}$, ..., $C_m^{(2)}$ such that $C_m^{(2)} = (B_i \vee D_i)$, $m \in Z^+$.
(b) A set of $n$ first-order clauses where each clause contains exactly one literal is represented as $C_1^{(1)}$, $C_2^{(1)}$, ..., $C_n^{(1)}$ such that $C_n^{(1)} = (A_i)$, $n \in Z^+$.

Note that $i$ represents the independent literals. The $C_m^{(2)}$ and $C_n^{(1)}$ denote the second-order and first-order clauses, respectively where a set of literals can be either positive or negative literal such $B_i \in \{B_i, \neg B_i\}$, $D_i \in \{D_i, \neg D_i\}$ and $A_i \in \{A_i, \neg A_i\}$. CRAN2SAT is a logical formulation that introduces conditional randomness in the generation of clauses within the DHNN framework. According to [28], the structure of CRAN2SAT allows DHNN to govern at least one negative neuron connection within the second-order clauses. This is because CRAN2SAT excludes both positive literals in the second-order clauses and does not impose any restrictions on the first-order clauses. As a result, possible combinations of second-order clauses that consist of both positive literals are disregarded from the logical structure of CRAN2SAT. From a logical perspective, classical logical structures such as 2SAT and RAN2SAT, which allow all possible combinations of literals. In contrast, CRAN2SAT imposes a structural constraint that excludes second-order clauses containing only positive literals. The general formulation of CRAN2SAT denoted as $P_{CRAN2SAT}$ is presented in Eq. (1) and the definition of the clauses in $P_{CRAN2SAT}$ is provided in Eq. (2):

$$P_{CRAN2SAT} = \wedge_{i=1}^{m} C_i^{(2)} \wedge_{i=1}^{n} C_i^{(1)}, \tag{1}$$

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

Note that $m$ represents the total number of the second-order clauses, $n$ denotes the total number of first-order clauses, and $k$ indicates the order of the clauses. In this paper, the randomized distribution structure of CRAN2SAT will act as a neuron representation in DHNN. Thus, the general implementation of CRAN2SAT into DHNN (or referred as DHNN—CRAN2SAT) is described based on the process involved during the learning and retrieval phase of DHNN.

During the learning phase, the objective is to determine the optimal value of the synaptic weight which represent the strength of neuron connections. This phase focuses on minimizing the logical inconsistency of $P_{CRAN2SAT}$ which corresponds to the minimization of the cost function of CRAN2SAT($E_{P_{CRAN2SAT}}$). According to [30], the value of $E_{P_{CRAN2SAT}}$ is proportional to the number of unsatisfied clauses in $P_{CRAN2SAT}$. Thus, an increase in unsatisfied clauses results in a higher value of $E_{P_{CRAN2SAT}}$. Achieving a zero-cost function ($E_{P_{CRAN2SAT}} = 0$) indicates that the logical inconsistency of $P_{CRAN2SAT}$ has been fully minimized show that all clauses in $P_{CRAN2SAT}$are satisfied. The cost function is formulated as shown in Eqs. (3) and (4):

$$E_{P_{CRAN2SAT}} = \sum_{i=1}^{m} \left( \prod_{j=1}^{2} Q_{ij} \right) + \sum_{i=1}^{n} \left( \prod_{j=1}^{1} Q_{ij} \right), \tag{3}$$

$$Q_{ij} = \begin{cases} \frac{1}{2}(1 - S_X), & \text{if } \neg X \\ \frac{1}{2}(1 + S_X), & \text{if } X \end{cases}. \tag{4}$$

The notation of $S_X$ in Eq. (4) represents the neuron state in a clause where $X$ and $\neg X$ are positive and negative literal, respectively for which $X$consist of arbitrary literals of $\{A_i, B_i, D_i\}$ within $P_{CRAN2SAT}$. The computation of the synaptic weights is obtained by using WA method [6] by comparing the coefficient of $E_{P_{CRAN2SAT}}$ with the Lyapunov energy function of DHNN $(H_{P_{CRAN2SAT}})$ formulated as in Eq. (5):

$$H_{P_{CRAN2SAT}} = -\frac{1}{2} \sum_{i=1, i \neq j}^{N} \sum_{j=1, i \neq j}^{N} W_{ij}^{(2)} S_i S_j - \sum_{i=1}^{N} W_i^{(1)} S_i \tag{5}$$

The factor $\frac{1}{2}$ prevents counting the same neuron interaction twice because the synaptic weights are symmetric $(W_{ij} = W_{ji})$. Under this condition, the Lyapunov energy function decreases monotonically during neuron updates, ensuring that the DHNN converges to a stable equilibrium state. These support the convergence of the network to a stable state [5]. Based on Eq. (5), $W_{ij}^{(2)}$ and $W_i^{(1)}$ represent synaptic weights for the second-order and first-order, respectively while$S_j$ refers to neuron state in $P_{CRAN2SAT}$. In this paper, the Election Algorithm (EA) is utilized to ensure that the network achieves an optimal learning phase which consequently leads to optimal synaptic weights [13]. Two important characteristics of synaptic weights in DHNN are symmetric weights such that $W_{ij} = W_{ji}$ and there is no self-connection such that $W_{ii} = W_{jj} = 0$. Furthermore, the generated values of $W_{ij}^{(2)}$ and $W_i^{(1)}$ are stored in Content Addressable Memory (CAM) in matrix form which allows DHNN to memorize the optimal neuron connections. These weights are subsequently used in local field computations during the retrieval phase.

During the retrieval phase, the aims is to retrieve an optimal final neuron state. To achieve this, the computation of local field $(h_i)$ of DHNN which utilizes the value of $W_{ij}^{(2)}$ and $W_i^{(1)}$ stored in CAM is calculated using Eq. (6):

$$h_i = \sum_{j=1, j \neq i}^{N} W_{ij}^{(2)} S_j + W_i^{(1)} \tag{6}$$

Subsequently, the activation function is applied to squash the value of $h_i$ into bipolar state, either 1 or −1 as illustrated in Eq. (7):

$$S_i^f(t) = \begin{cases} 1, & \text{if } g(h_i) \geq 0 \\ -1, & \text{otherwise} \end{cases}. \tag{7}$$

Based on the Eq. (7), the $g(h_i)$is computed using the Smish activation function which is shown in Eq. (8) where the values of $\alpha$ and $\beta$ are learnable parameters. Based on work by [28], Smish activation function able to maintain partial sparsity and thus increase the variation between

the retrieved final neuron state. The parameters $\alpha$ and $\beta$ follow the configuration adopted in the previous CRAN2SAT–DHNN [28] and remain fixed throughout the experiments to ensure consistency with the original model.

$$g(h_i) = \alpha h_i \cdot \tanh[\ln(1 + sigmoid(\beta h_i))] \tag{8}$$

When the final neuron states are updated correctly, these states will converge towards the global minimum energy which consequently leads to a global minima solution. Therefore, the retrieved final neuron state $\left(S_i^f(t)\right)$ will be used to evaluate the final energy $\left(H_{P_{CRAN2SAT}}\right)$ of the network by using Eq. (9). The $H_{P_{CRAN2SAT}}$ is calculated by using the updated value of $S_i^f(t)$ and $S_j$ represents the neuron state in $P_{CRAN2SAT}$ as shown in Eq. (9):

$$H_{P_{CRAN2SAT}} = -\frac{1}{2}\sum_{i=1,i\neq j}^{N}\sum_{j=1,i\neq j}^{N} W_{ij}^{(2)} S_i^f S_j - \sum_{i=1}^{N} W_i^{(1)} S_i^f \tag{9}$$

Furthermore, the minimum energy of the network denoted as $H_{P_{CRAN2SAT}}^{\min}$ is computed using Eq. (10) where $m$ and $n$ represent the number of second-order clauses and first-order clauses in $P_{CRAN2SAT}$, respectively.

$$H_{P_{CRAN2SAT}}^{\min} = -\left(\frac{m+2n}{4}\right) \tag{10}$$

The minimum energy in Eq. (10) is obtained from the energy contribution of each clause in $P_{CRAN2SAT}$. Each satisfied second-order clause contributes $-\frac{1}{4}$ to the network energy, while each satisfied first-order clause contributes $-\frac{1}{2}$. The quality of the $S_i^f(t)$ that is retrieved by DHNN—CRAN2SAT model is examined using Eq. (11), whereby *Tol* is the tolerance value used as an indicator in evaluating the quality of $S_i^f(t)$.

$$\left|H_{P_{CRAN2SAT}} - H_{P_{CRAN2SAT}}^{\min}\right| \leq Tol. \tag{11}$$

If Eq. (11) is satisfied, $S_i^f(t)$ that is retrieved by the DHNN—CRAN2SAT is considered a global minima solution, otherwise $S_i^f(t)$is classified as a local minima solution. Although EA ensures the optimal state of $S_i^f(t)$ and Smish activation function enhances diversification between the states, the retrieved states may present high similarity to the benchmark states. To address this, implementing a multi-objective function during the retrieval phase of DHNN could further improve the quality of the retrieved final neuron states. Therefore, the retrieval phase is formulated as a multi-objective optimization problem where the decision variable is the final neuron state $S_i^f(t)$ and the objective is to optimize diversification, global optimality, and similarity minimization simultaneously. In the retrieval phase of the DHNN, the multi-objective functions are defined as indicated in Eq. (12):

$$Obj\left(f_D, f_Z, f_{SI}\right), \tag{12}$$

where $f_D$ is represents the fitness for the diversification of the final neuron state, $f_Z$ is denotes the fitness for the global state and $f_{SI}$ is indicates the fitness for minimizing the similarity indices. The first objective function, $Obj_1(f_D)$ defined as in Eq. (13) aims to maximize the diversification of the final neuron state.

$$Obj_1(f_D) = \max[f_D], \quad 0 \leq f_D \leq F. \tag{13}$$

This diversification of the final neuron state is measured by the variation of negative neuron states within the clauses of CRAN2SAT. Consequently, the fitness evaluation for the diversified neuron states is subject to Eq. (14):

$$Obj_1(f_D) : f_D = \sum_{i=1}^{m} f_{C_i^{(2)}} \tag{14}$$

where $f_{C_i^{(2)}}$ is fitness of second-order clause for $i = 1, 2, \ldots, m$. The fitness evaluation for $f_D$ is only measured based on the second-order clauses of the logical rule $P_{CRAN2SAT}$. The second objective function, $Obj_2(f_Z)$ is defined as in Eq. (15) which aims to achieve a final neuron state with the maximum number of global minima solution

$$Obj_2(f_Z) = \max[f_Z], \quad 0 \leq f_Z \leq F, \tag{15}$$

subject to

$$Obj_2(f_Z) : f_Z = \sum_{i=1}^{m} f_{C_i^{(2)}} + \sum_{i=1}^{n} f_{C_i^{(1)}}. \tag{16}$$

According to Eq. (16), the fitness evaluation for $f_Z$ is evaluated based on the logical rule $P_{CRAN2SAT}$ where $f_{C_i^{(2)}}$ represents the fitness of the second-order clause for $i = 1, 2, \ldots, m$ and $f_{C_i^{(1)}}$ represents the fitness of the first-order clause for $i = 1, 2, \ldots, n$. Based on $Obj_1(f_D)$ and $Obj_2(f_Z)$, $F$ represents the maximum solution obtained from the product of the number of trials and the number of neuron combinations. This setting ensures sufficient exploration during the retrieval phase.

Finally, the objective function, $Obj_3\left(f_{SI}\right)$ defined as in Eq. (17) ensures minimal similarity between the optimized final neuron state and the benchmark state. In this context, the benchmark neuron state represents the initial neuron state that is generated by the logical rule $P_{CRAN2SAT}$.

$$Obj_3\left(f_{SI}\right) = \min\left[f_{SI}\right], \tag{17}$$

where

$$Obj_3\left(f_{SI}\right) : f_{SI} \leq \sigma, \quad \sigma \in [0, 1]. \tag{18}$$

Based on Eq. (18), the fitness evaluation for $f_{SI}$ will be determined using similarity indices where the obtained values are less than $\sigma$ which is generated by a respective formulation of similarity indices. To fulfil all the multi-objective functions outlined in Eq. (12), a Hybrid Binary Whale Optimization Algorithm (HBWOA) is designed to optimize the final neuron state retrieved by DHNN. Fig. 1 shows a schematic diagram illustrating how the retrieved final neuron state is embedded into HBWOA during the retrieval phase of DHNN.

As illustrated in Fig. 1, each whale represents a candidate of neuron state in the DHNN retrieval phase. The binary representation corresponds to the bipolar neuron states, where each element denotes the state of a neuron. The fitness of each candidate is evaluated using the proposed multi-objective functions, and HBWOA iteratively updates these states through its exploration and exploitation mechanisms to obtain an optimal final neuron state.

## 4. The proposed hybrid binary whale optimization

The Whale Optimization Algorithm (WOA) is a nature-inspired metaheuristic that mimics the hunting and social behavior of humpback whales. WOA was introduced by [31] and has demonstrated strong performance across a range of optimization tasks. To adapt WOA for discrete optimization tasks, [32] proposed two binary variants of WOA, namely BWOA-S and BWOA-V. Despite their potential, the application of these variants in DHNN remains unexplored. WOA updates solutions through two mechanisms, namely the exploitation of the best solution and the exploration of random candidates which provides strong potential in optimizing logical representations. However, the effective application of BWOA in logic representations requires a well-balanced exploitation strategy. Uncontrolled exploration may lead to the evaluation of irrelevant solutions that do not satisfy the criteria of multi-objective optimization. Therefore, a well-designed optimization operator is essential for guiding the search toward high-quality final neuron states. To address this issue, the proposed work implements a Hybrid Binary Whale Optimization Algorithm (HBWOA) in the retrieval phase of DHNN—CRAN2SAT. The goal is to optimize the final neuron

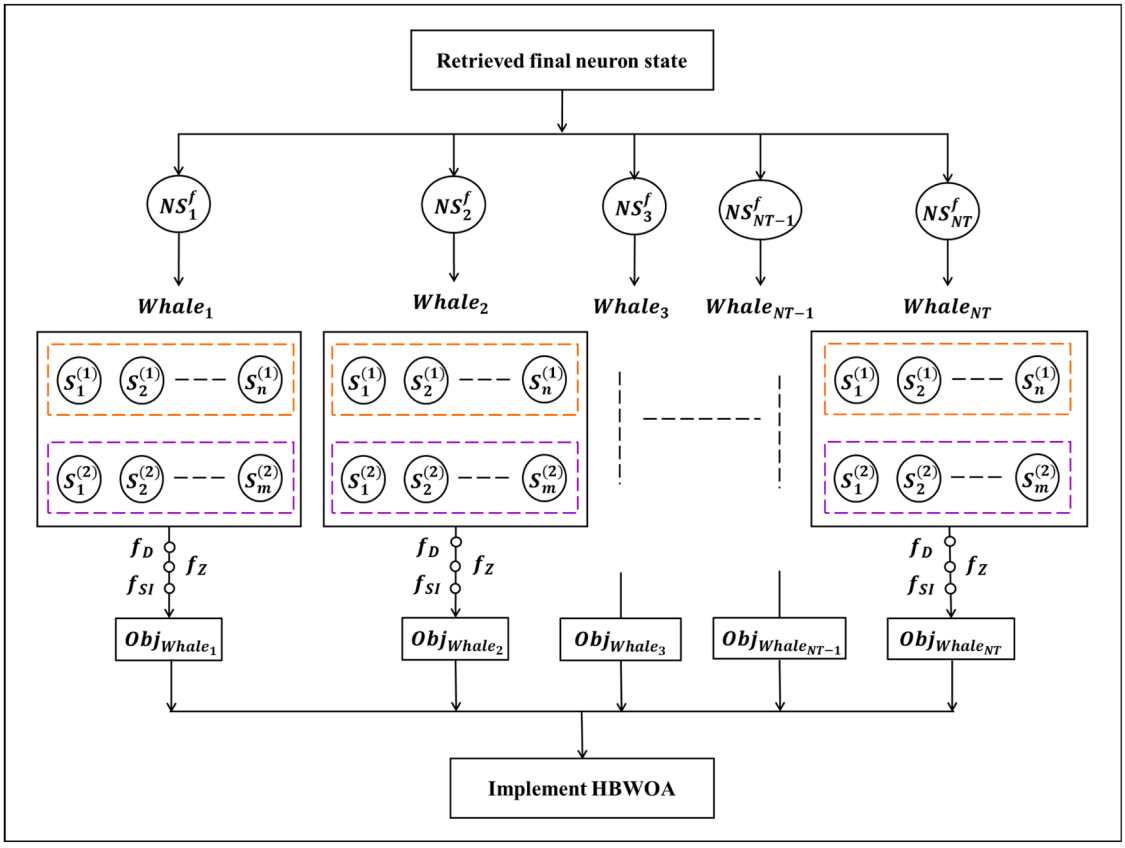


**Fig. 1.** Schematic diagram of optimizing retrieval phase of DHNN.

state that satisfies all multi-objective functions outlined in Eq. (12). The first-order clauses are excluded from the optimization process because changes in their neuron states directly affect the satisfiability of $P_{CRAN2SAT}$. Small changes in these states may immediately influence clause satisfaction and affect the retrieval process. Therefore, first-order clauses are treated as fixed logical constraints during the HBWOA optimization. This allows the search to focus on second-order clauses while maintaining the satisfiable interpretation of $P_{CRAN2SAT}$ and ensuring stable and effective multi-objective optimization [33].

### 4.1. Initialization

The HBWOA process begins with initializing the whale population as shown in Eq. (19) where $g(h_i)$ is generated using the value of local field, $h_i$ which is transformed into a bipolar state using the Smish activation function in Eq. (8).

$$Whale_i\left(S_i^f\right) = \begin{cases} 1, & \text{if} g(h_i) \geq 0 \\ -1, & \text{otherwise} \end{cases}. \tag{19}$$

### 4.2. Fitness evaluation

After initializing the whale population, the fitness of each whale is evaluated based on the three objective functions in Eq. (12). Diversification of the final neuron state is essential to avoid repetitive states. The percentage of negativity denoted as $Q$ is used as an indicator for achieving diversification. Therefore, the fitness evaluation for the diversified neuron state $(f_D)$is formulated in Eq. (20):

$$f_D = Q\left(\sum_{i=1}^{m} f_{C_i^{(2)}}\right), \quad Q = [0,1] \tag{20}$$

$$\left(f_{C_i^{(2)}}\right) = \begin{cases} 0, & \text{if}\left(S_{B_i}^f, S_{D_i}^f\right) = (1,1) \\ 1, & \text{Otherwise} \end{cases}. \tag{21}$$

The $\left(S_{B_i}^f, S_{D_i}^f\right)$ refers to the optimized final neuron state for the second-order clause with $B_i$ and $D_i$ representing the literals within the clauses. If a clause in $Whale_i\left(S_i^f\right)$ contains at least one negative state, the algorithm counts this as achieving diversity denoted by 1. If diversity is not achieved, it is counted as 0. The fitness evaluation for the global state $(f_Z)$ is given by Eq. (22):

$$f_Z = \sum_{i=1}^{m} f_{C_i^{(2)}} + \sum_{i=1}^{n} f_{C_i^{(1)}}, \tag{22}$$

$$\left(f_{C_i^{(2)}}, f_{C_i^{(1)}}\right) = \begin{cases} 1, & \text{if} E_{P_{CRAN2SAT}} = 0 \\ 0, & \text{Otherwise} \end{cases}, \tag{23}$$

where $f_{C_i^{(2)}}$ represents the fitness of the second-order clause for $i = 1, 2, \ldots, m$ and $f_{C_i^{(1)}}$ denotes the fitness of the first-order clause for $i = 1, 2, \ldots, n$. Based on Eq. (23), if the first and second-order clause of $P_{CRAN2SAT}$ achieve a satisfied interpretation, then it will evaluated as a satisfied clause denoted as $E_{P_{CRAN2SAT}} = 0$. Furthermore, the fitness evaluation for minimizing similarity indices $(f_{SI})$ is given by Eq. (24):

$$f_{SI} \leq \sigma, \quad \sigma \in SI \tag{24}$$

The fitness evaluation for $f_{SI}$ is determined using similarity indices $(SI)$ where the obtained values are less than $\sigma$ generated by the respective $SI$ formulation. Each $Whale_i\left(S_i^f\right)$ then undergoes a fitness evaluation to identify the best solution denoted as $Whale_i^*\left(S_i^f\right)$ which achieves the highest fitness as described in Eq. (12).

### 4.3. Primary selection

In the primary selection stage, the $Whale_i^*\left(S_i^f\right)$ with the highest fitness value as defined in Eq. (12) is selected. This mechanism, inspired by the cloning operator in CSA by [17]. The cloning operator is implemented in the HBWOA to enhance exploration by expand the solution space and improve the search for optimal neuron states. $Whale_i^*\left(S_i^f\right)$ is cloned based on a clone rate $CR = [0, 1]$. The number of clones $n\left(Whale_i^*\left(S_i^f\right)\right)$ is determined as shown in Eq. (25):

$$n\left(Whale_i^*\left(S_i^f\right)\right) = CR{\cdot}Pop\left(Whale_i\left(S_i^f\right)\right) \tag{25}$$

The total whale population, $Pop\left(Whale_i\left(S_i^f\right)\right)$ increases with the addition of copies of the best whale solutions. This new population, $Pop^{New}\left(Whale_i\left(S_i^f\right)\right)$ moves to the next stage where each whale updates its position using either the shrinking encircling mechanism or the spiral update.

### 4.4. Update position

The position of each whale, $Whale_i\left(S_i^f\right)$ is updated based on a simultaneous behavior where a random choice with a 50% probability is made between the shrinking encircling mechanism and the spiral update [31]. The random probability $p$lies within $[0, 1]$, while $b$ a constant defining the shape of the logarithmic spiral, is set to $b = 1$[34]. Additionally, $l$ is a random number within $[-1, 1]$. This behavior is mathematically represented in Eq. (26):

$$Whale_i\left(S_i^f(t+1)\right) = \begin{cases} Whale_i\left(S_i^f(t)\right) - A{\cdot}D_i, & \text{if } p < 0.5 \\ D_i{}^*{\cdot}e^{bl}{\cdot}\cos(2\pi l) + Whale_i{}^*\left(S_i^f(t)\right), & \text{otherwise} \end{cases}. \tag{26}$$

When $p \geq 0.5$, the position of each whale is updated using a spiral pattern with the distance $D_i^*$ between the best whale and the current whale calculated using Eq. (27):

$$D_i{}^* = \left|Whale_i{}^*\left(S_i^f(t)\right) - Whale_i\left(S_i^f(t)\right)\right| \tag{27}$$

If $p < 0.5$, $Whale_i\left(S_i^f\right)$ is updated using the shrinking encircling mechanism which depends on the coefficient $A$ in Eq. (32). When $|A| \geq 1$, the mechanism emphasizes exploration, allowing the WOA to perform an extensive search by utilizing a randomly selected whale from the current population, $Whale_{i_{rand}}\left(S_i^f\right)$. The distance $(D_i)$ and update states are also calculated based on $Whale_{i_{rand}}\left(S_i^f\right)$ as shown in Eqs. (28) – (29):

$$D_i = \left|C{\cdot}Whale_{i_{rand}}\left(\left(S_i^f(t)\right)\right) - Whale_i\left(\left(S_i^f(t)\right)\right)\right|, \tag{28}$$

$$Whale\left(\left(S_i^f(t+1)\right)\right) = Whale_{rand}\left(\left(S_i^f(t)\right)\right) - A{\cdot}D_i. \tag{29}$$

For $|A| < 1$, the update states are based on $Whale_i{}^*\left(S_i^f\right)$ as shown in Eqs. (30) – (31):

$$D_i = \left|C{\cdot}Whale_i{}^*\left(S_i^f(t)\right) - Whale_i\left(S_i^f(t)\right)\right|, \tag{30}$$

$$Whale_i\left(S_i^f(t+1)\right) = Whale_i\left(S_i^f(t)\right) - A{\cdot}D_i. \tag{31}$$

Note that, the value of $A$ and $C$ are mathematical coefficients expressed by Eqs. (32) and (33):

$$A = 2{\cdot}a{\cdot}r_1 - a \tag{32}$$

$$C = 2{\cdot}r_2 \tag{33}$$

$$a = 2 - \frac{2t}{\text{maxIter}} \tag{34}$$

As shown in Eqs. (32) and (33), $r_1$ and $r_2$ are random vectors in the range $[0, 1]$. The convergence factor $a$ decreases gradually from 2 to 0 over iterations. The variable $t$ represents the current iteration and maxIteris the maximum number of iterations.

### 4.5. Transfer into bipolar

After updating the position of $Whale_i\left(S_i^f(t+1)\right)$, the states are converted to bipolar values of 1 or −1 using the sigmoid function as describe in Eq. (35):

$$S\left(Whale_i\left(S_i^f(t)\right)\right) = \frac{1}{1 + e^{-Whale_i\left(S_i^f(t+1)\right)}} \tag{35}$$

The bipolar update is performed based on the value obtained from $S\left(Whale_i\left(S_i^f(t)\right)\right)$. The bipolar update for each whale in the population is shown in Eq. (36):

$$Whale_i\left(S_i^f(t+1)\right) = \begin{cases} 1, & \text{if } S\left(Whale_i\left(S_i^f(t)\right)\right) \geq rand \\ -1, & \text{otherwise} \end{cases} \tag{36}$$

Although Eq. (36) involves a probabilistic element in updating the final neuron state, this equation does not represent a purely random or exhaustive search. Eq. (36) serves as a guided transfer mechanism influenced by the Whale's solutions updates through Eqs. (19) – (35). This mechanism enhances the ability of Whale's solutions to converge toward an optimal solution within the discrete domain. The updating process only involves the second-order clause of the CRAN2SAT logical rule with the first-order clause remains unchanged. The effectiveness of the updated states is evaluated through the fitness function in Eq. (12). If the updated $Whale_i\left(S_i^f(t+1)\right)$ solution does not satisfy the multi-objective functions, the neuron states proceed to the next stage called greedy mutation.

### 4.6. Greedy mutation

The HBWOA introduces a greedy mutation mechanism that applies random changes within $Whale_i\left(S_i^f(t+1)\right)$ to enhance diversity and improving overall performance. By focusing on localized search operations, greedy mutation identifies the desired state through iterative adjustments until the conditions in Eq. (12) are met. This mechanism is applied exclusively to the second-order clauses of CRAN2SAT as first-order clauses are excluded from the optimization process. The highest fitness value of the solution strings for $Whale_i\left(S_i^f(t)\right)$ is carried forward to the next stage, where these neuron states serve as the initial population for the next generation. The Gower and Legendre similarity index $(GLI)$ will be used to measure similarity of negative values between the optimized final neuron state $Whale_i\left(S_i^{fa}\right)$ with the benchmark state $Whale_i(S_i)$ representing the initial neuron state of logical rule $P_{CRAN2SAT}$.

$$a = \begin{cases} 1, & \text{if } \left(S_i, S_i^{fa}\right) = (1, 1) \\ 0, & \text{otherwise} \end{cases} \tag{37}$$

$$b = \begin{cases} 1, & \text{if } \left(S_i, S_i^{fa}\right) = (1, -1) \\ 0, & \text{otherwise} \end{cases} \tag{38}$$

$$c = \begin{cases} 1, & \text{if } \left(S_i, S_i^{fa}\right) = (-1,1) \\ 0, & \text{otherwise} \end{cases} \tag{39}$$

$$d = \begin{cases} 1, & \text{if } \left(S_i, S_i^{fa}\right) = (-1,-1) \\ 0, & \text{otherwise} \end{cases} \tag{40}$$

The range of *GLI* is between $(0,1)$ with the lowest value indicating low similarity in negative values. Eq. (37) – (40) show the computation of similarity between the states while Eq. (41) presents the *GLI* formulation for measuring the similarity of negative values between $Whale_i\left(S_i^{fa}\right)$ and $Whale_i(S_i)$.

$$GLI = \frac{a+d}{a+0.5(b+c)+d} \tag{41}$$

One of the fitness evaluations aims to minimize the similarity indices $(f_{SI})$ as stated in Eq. (17) subject to Eq. (41). In Eqs. (37) – (40), $S_i$ represents the initial neuron state of logical rule $P_{CRAN2SAT}$ while $S_i^{fa}$ denotes the final neuron state optimized using HBWOA. The optimization using HBWOA continues until the stopping criteria are met, either when the maximum number of generations is reached or when the solution for $Whale_i\left(S_i^{fa}\right)$ achieves the fitness specified in Eq. (12). $Whale_i\left(S_i^{fa}\right)$ is considered optimal if it meets the fitness evaluation as represented by:

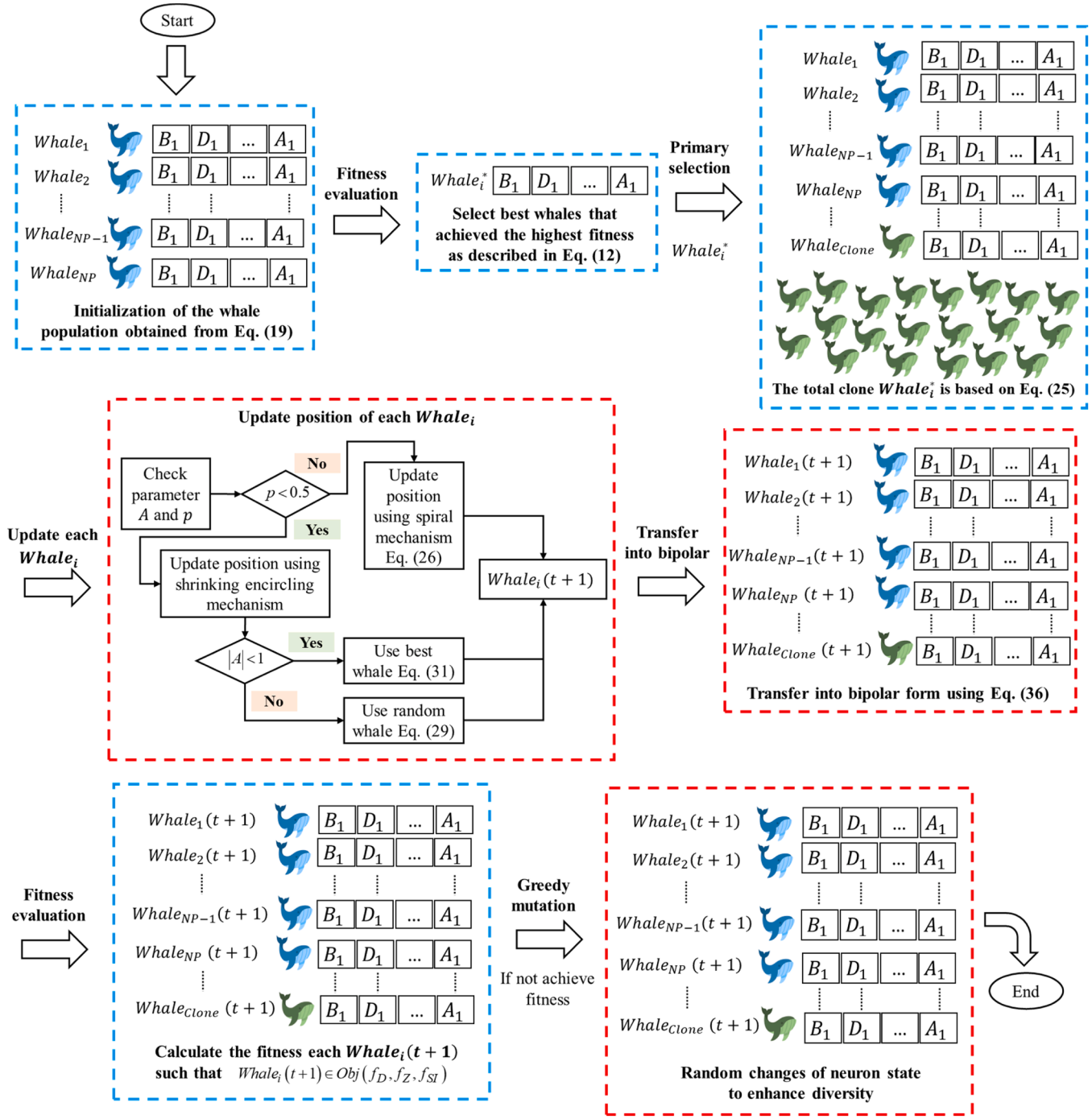


**Fig. 2.** Illustrative example of operator in HBWOA.

$$S_i^{fa} \rightarrow Obj\left(f_D, f_Z, f_{SI}\right). \quad (42)$$

Although HBWOA offers flexibility by updating solutions based on both the best solution (Equation ((26)) and random solution (Eq. (29)), there remains a potential risk of overfitting, especially when HBOWA primarily focuses on exploiting the best solutions during the search process. To address this issue, the final solutions are guided by a minimum *GLI*similarity threshold which promotes diversity and helps reduce the risk of overfitting.

Fig. 2 illustrates the process of HBWOA in optimizing the retrieved final neuron state within the DHNN where whales symbolically represent neurons. The red dotted box highlights the operator used to update the neuron state based on the second-order clause while the first-order clause remains unchanged throughout the process. While Algorithm 1 presents the pseudocode of HBWOA in the retrieval phase of DHNN—CRAN2SAT.

## 5. The proposed logic mining model

This section explains the proposed logic mining model in doing classification tasks. Classification is a fundamental task in machine learning where the goal is to train a computational model on a labelled dataset to accurately classify new unlabelled data. The previous section discussed the implementation of CRAN2SAT into DHNN with the Smish activation function, along with a multi-objective function that optimize the retrieved final neuron state. These mechanisms form the main component of the proposed logic mining model namely Conditional Random 2 Satisfiability Reverse Analysis Method (CR2SATRA). CR2SATRA consists of three main phases: pre-processing phase, learning phase and retrieval phase of DHNN. Fig. 3 illustrates the key components of CR2SATRA.

**Algorithm 1**
Pseudocode of HBWOA in the retrieval phase DHNN—CRAN2SAT.

**Initialization:** $g\left(h_i\right)=\left\{S_1^f\ S_2^f\ S_3^f\ S_4^f\ S_5^f \ldots S_{NN}^f\right\}$
**for** $i=1,2,\ldots,m$ **do**
{Fitness Evaluation}
  **for** $i=1,2,\ldots,m$ **do**
    Evaluate the fitness of second order $f_{C_i^{(2)}}$ by Equation (12);
    Select best whales $Whale_i^*\left(S_i^f\right)$ by Equation (12);
  **End for**
{Primary Selection}
  **for** $i=1,2,\ldots,m$ **do**
    The best whales $Whale_i^*\left(S_i^f\right)$ is obtained by Equation (12);
  **End for**
{Update Solution}
  **For** $i=1,2,\ldots,m$ **do**
    **If** $p<0.5$
      **If** $|A|<1$
        Update position by Equation (31)
      **Else**
        Update position by Equation (29)
      **End If**
    **Else**
      Update position by Equation (26)
    **End if**
      Transfer into bipolar form by Equation (36)
  **End for**
{Greedy Mutation}
  **for** $i=1,2,\ldots,m$ **do**
    Calculate the fitness value to update whale position Equation (12)
    **If** (the whale population reaches its highest level of fitness of diversity)
      Termination Criteria
    **Else**
      Greedy mutation to $Whale_i\left(S_i^f(t)\right)$
    **End if**
  **End for**
**End for**
**Output:** Final neuron state that achieve maximum $N_D$, $N_{DG}$ and minimum $GLI$

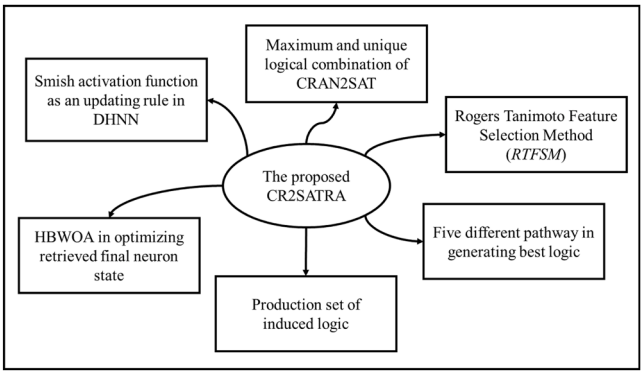


**Fig. 3.** Components of CR2SATRA.

### 5.1. Pre-processing

During the pre-processing phase, the data preparation involves three main steps. First, the attributes of the raw data entries $(e_{ij})$ are transformed into bipolar representation using the *k*-mean clustering method [19] as formulated in Eq. (43):

$$S_i = \begin{cases} 1, & \text{if } e_{ij} \geq \overline{A_i} \\ -1, & \text{otherwise} \end{cases} \tag{43}$$

Based on Eq. (43), $e_{ij}$ represent the values within the attributes while $\overline{A_i}$ referred to the mean value of each attribute within the dataset. After all the $e_{ij}$ have been transformed into bipolar representation, the significant attributes are selected based on the proposed attribute selection method.

Effective attribute selection is crucial especially for datasets with many attributes, as it helps to identify the most significant attributes that best represent the behavior of the dataset. In this paper, the Rogers Tanimoto Feature Selection Method (*RTFSM*) is used to eliminate irrelevant attributes with low similarity to the class of the dataset. *RTFSM* calculates the distance between the positive and negative entries of each attribute to assess their relevance to the outcome of the datasets. This method uses similarity indices as the foundation for attribute selection where a similarity index is a formula used to measure the distance between two objects. In this context, the focus is on both positive and negative states, specifically the states between any attribute and the dependent attribute of the dataset. By calculating this distance, *RTFSM*identifies the attributes with the highest similarity index which indicates a strong relationship with the class of the dataset and dependent attribute. Since *RTFSM* considers both positive and negative entries in bipolar data representation, it provides a more balanced assessment of the relationship between attributes and the dataset outcomes which is particularly important for imbalanced datasets. Moreover, it aligns well with the bipolar representation used in the DHNN framework.

The following equation expresses the formulation of *RTFSM* with $e_i$ representing the raw data entries with respect to the attributes of $E_i$ and the $P_i$ denoting the decision or actual outcome of the dataset.

$$a_i(e_i) = \begin{cases} 1, & \text{if}(E_i, P_i) = (1,1) \\ 0, & \text{otherwise} \end{cases}, \tag{44}$$

$$b_i(e_i) = \begin{cases} 1, & \text{if}(E_i, P_i) = (1,-1) \\ 0, & \text{otherwise} \end{cases}, \tag{45}$$

$$c_i(e_i) = \begin{cases} 1, & \text{if}(E_i, P_i) = (-1,1) \\ 0, & \text{otherwise} \end{cases}, \tag{46}$$

$$d_i(e_i) = \begin{cases} 1, & \text{if}(E_i, P_i) = (-1,-1) \\ 0, & \text{otherwise} \end{cases}, \tag{47}$$

$$RT = \frac{\sum_{i=1}^{n} a_i + \sum_{i=1}^{n} d_i}{\sum_{i=1}^{n} a_i + 2\left(\sum_{i=1}^{n} b_i + \sum_{i=1}^{n} c_i\right) + \sum_{i=1}^{n} d_i}, \tag{48}$$

$$E_i \in \max[RT]. \tag{49}$$

Noting that $a_i,\ b_i,\ c_i,\ d_i$ denote the scoring mechanism for the similarity differences between the *i*th entry of the independent attribute and the *i*th entry of the dependent attribute with *n* representing the total number of entries. Consequently, based on Eq. (49), the maximum value of *RT*will be chosen as the attribute selection. In the final step of the pre-processing phase, the selected attributes$(E_i)$ including the outcome of the dataset $(P_i)$ will be divided into a learning dataset $(P_{learn})$ and a testing dataset $(P_{test})$. Additionally, during the process of splitting the dataset, *k*-fold cross validation is employed to helps the model explore a broader search space [24] and then the average of all *k*-fold will be taken as the final result for CR2SATRA.

### 5.2. Learning phase

After the splitting dataset, the logic mining model proceeds to the learning phase of DHNN as described below. In the learning phase of CR2SATRA, a maximum number of unique logical rule combinations for the proposed model $(Unique_{CR2SATRA})$ is generated as shown in Eq. (50):

$$Unique_{CR2SATRA} = \left(2^k - 1\right)^m \cdot \left(2^k\right)^n. \tag{50}$$

Based on the Eq. (50), generally, the possible combination of clauses is denoted as $c = 2^k$ where *k* is denote the order of the clauses. The computation of $\left(2^k - 1\right)^m$ in Eq. (50) is generalized for the second-order clauses of $P_{CRAN2SAT}$ where *m* represents the number of second-order clauses in CR2SATRA. While $\left(2^k\right)^n$ is generalized for the first-order clauses of $P_{CRAN2SAT}$ where *n* represents the number of first-order clauses in CR2SATRA. Each $Unique_{CR2SATRA}$ is embedded in $P_{learn}$ to compare the actual class with the predicted class to obtain the classification for the confusion matrix. The unique structure of the proposed logical rule $P_{CRAN2SAT}$ enhances the ability of CR2SATRA to capture the majority of the patterns within $P_{learn}$ and expanding the search space while avoiding learn redundant logical rules.

Existing logic mining models often identify the best logic based on the highest frequency of clauses associated with a positive outcome, max[*TP*]. However, this approach can be biased with imbalanced datasets where the frequency of clauses might not accurately reflect the actual relationships of the datasets, posing challenge for real-world applications. Additionally, existing logic mining models retrieve a single induced logic for knowledge extraction, limiting their ability to effectively address a wide range of real-world datasets. To overcome these issues, CR2SATRA introduces a new method for identifying the best logic $\left(P^{best}_{CR2SATRA}\right)$ during the learning phase. By using different pathway in generating the best logic by utilizing different confusion matrixes. This approach expands the search space and producing a different set of induced logics that enhance flexibility in doing classification tasks.

The developed CR2SATRA is based on multi-unit DHNN model that employs five confusion matrixes such as Accuracy (*ACC*), Precision (*PRE*), Specificity (*SPE*), Matthews Correlation Coefficient (*MCC*) and Misclassification Rate (*MCR*) to generate multiple best logic for the $P^{best}_{CR2SATRA}$. These metrics were selected because they collectively capture critical aspects of both majority and minority class performance, providing a comprehensive view of classification behavior without bias toward the majority class, which is essential for addressing imbalanced datasets. In the proposed logic mining model, multiple best logics are generated using different confusion matrixes. Each confusion matrixes emphasizes different aspects of classification performance, capturing both positive and negative outcomes. This allows the model to analyze the dataset from multiple perspectives rather than relying on a single

evaluation criterion. By considering multiple best logics, the proposed approach reduces bias toward majority-class and enhances flexibility in representing classification behavior, particularly for imbalanced datasets. The first metric used to generate $P^{best}_{CR2SATRA}$ is *ACC* as shown in Eq. (51) with its formulation in Eq. (52):

$$P^{best}_{CR2SATRA} = \max[ACC] \tag{51}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{52}$$

Generating $P^{best}_{CR2SATRA}$ using *ACC* no longer focuses solely on positive outcomes. But this approach will utilize both *TP* and *TN* classifications enabling to a more comprehensive evaluation of both positive and negative outcomes. Medical datasets often suffer from class imbalance, where the majority class represents negative or healthy patients (*TN*) which consequently results in a limited amount of data for the minority class representing positive or sick patients (*TP*). Therefore, generating best logic based on both *TP* and *TN* classifications as measured by accuracy is crucial for enhancing the performance of logic mining models.

The second metric proposed for generating $P^{best}_{CR2SATRA}$ is *PRE* formulated as in Eq. (53) and the computation of *PRE* as shown in Eq. (54). The $Unique_{CR2SATRA}$ with the highest *PRE* value will be selected as $P^{best}_{CR2SATRA}$. By maximizing *PRE*, CR2SATRA improves knowledge extraction by accurately identifying a higher proportion of positive instances within the dataset.

$$P^{best}_{CR2SATRA} = \max[PRE] \tag{53}$$

$$PRE = \frac{TP}{TP + FP} \tag{54}$$

A high precision value ensures that $P^{best}_{CR2SATRA}$ accurately learns from the data and effectively predicts **positive** outcomes by minimizing the false positive (*FP*) classification. The third metric for generating $P^{best}_{CR2SATRA}$ is *SPE* as shown in Eq. (55) with its formulation in Eq. (56). Selecting $P^{best}_{CR2SATRA}$ based on the maximum *SPE* value highlights the ability of CR2SATRA to effectively learn from the dataset by correctly predicting *TN* while minimizing false positives.

$$P^{best}_{CR2SATRA} = \max[SPE], \tag{55}$$

$$SPE = \frac{TN}{TN + FP} \tag{56}$$

As highlighted by [35], minimizing *FP* is particularly important in the cases where misclassifying a positive instance is more costly than misclassifying a negative one. For example, in medical diagnosis, misclassifying a healthy patient as sick can lead to serious consequences. Similarly in fraud detection, accurately identifying authorized transactions is crucial for minimizing unnecessary investigations.

The fourth metric proposed for generating $P^{best}_{CR2SATRA}$ is *MCC* as presented in Eq. (57) and formulated in Eq. (58). The $Unique_{C2SATRA}$ achieving maximum *MCC* value is selected as $P^{best}_{CR2SATRA}$ during the learning phase. The *MCC* metric which considers all *TP*, *TN*, *FP* and *FN* predictions provide a comprehensive measure of classifier performance. A high *MCC* value indicating better prediction accuracy for both positive and negative instances.

$$P^{best}_{CR2SATRA} = \max[MCC], \tag{57}$$

$$MCC = \frac{TPTN - FPFN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{58}$$

*MCC* is valuable because it evaluates the overall classifier performance across all outcomes [36],[37] describes *MCC* as an effective metric for representing model quality especially with imbalance classes. Additionally, [14] highlight that *MCC* helps determine whether the logic mining model performs better, worse, or similarly to a random classifier.

The final metric proposed for generating $P^{best}_{CR2SATRA}$ is *MCR* presented in Eq. (59) with its formulation in Eq. (60). While generating $P^{best}_{CR2SATRA}$ using *ACC* is important, minimizing *MCR*is particularly crucial in cost-sensitive fields like medical diagnosis and fraud detection. As emphasized by [35], misclassifying a positive instance in medical diagnosis often carries a higher cost than misclassifying a negative one. For example, in cancer prediction, failing to diagnose cancer in a patient who has cancer (a false negative) is far more dangerous than misdiagnosing a healthy individual. This highlights the importance of cost sensitivity in medical decision-making.

$$P^{best}_{CR2SATRA} = \min[MCR], \tag{59}$$

$$MCR = \frac{FP + FN}{TP + TN + FP + FN} \tag{60}$$

Misclassifications can lead to delayed treatment and loss of life, highlighting the importance of accurate diagnostics in saving lives. Therefore, generating the best logic based on *MCR* helps determine if a logic mining model can function as a cost-sensitive classifier. For each performance metric, $\rho$ number of $P^{best}_{CR2SATRA}$ with optimal values are selected as the best logic, where $\rho$ denotes the number of best logics trained by DHNN. These $P^{best}_{CR2SATRA}$ for each confusion matrixes are then trained by EA in DHNN to obtain the synaptic weights. Each confusion matrixes are treated separately, resulting in a multi-unit DHNN where each unit corresponds to a specific confusion matrix. Since $P^{best}_{CR2SATRA}$ is based on five metrics, the DHNN is executed five times, with each unit corresponding to a specific confusion matrix.

### 5.3. Retrieval phase

By utilizing the synaptic weights stored in CAM, the local field $h_i$ for $\rho$ number of $P^{best}_{CR2SATRA}$ corresponding to each confusion matrix is computed by using Eq. (6). The value of $g(h_i)$ is then squashed to the final neuron state $\left(S_i^f\right)$ using the Smish activation function in Eq. (8) resulting in a value of either 1 or $-1$ based on the condition in Eq. (7). Unlike other existing logic mining models, $S_i^f$ retrieved by CR2SATRA is optimized using the proposed multi-objective optimization with HBWOA as shown in Eq. (12). The optimized $S_i^f$ is then transformed into the logical rule, producing the induced logic $\left(P^{induced}_{CR2SATRA}\right)$as shown in Eq. (61):

$$S_i^{induced} = \begin{cases} A_i, & \text{if} S_i^f = 1 \\ \neg A_i, & \text{if } S_i^f = -1 \end{cases}. \tag{61}$$

where $A_i$ and $\neg A_i$ represent positive and negative literal respectively.

Similar to the learning phase, the production of $P^{induced}_{CR2SATRA}$ is embedded into $P_{test}$ to compare the actual and predicted classes generating the classification of the confusion matrix. The $P^{induced}_{CR2SATRA}$with the highest value based on the performance metrics used to generate $P^{best}_{CR2SATRA}$ during the learning phase is selected as the best induced logic $\left(P^{induced}_{best}\right)$ that will be used in knowledge extraction of the datasets. The implementation of CR2SATRA is presented in Fig. 4.

## 6. Experimental setup

The experimental setup for evaluating the proposed CR2SATRA ensures a comprehensive assessment of its performance in doing classification tasks and knowledge extraction. This section outlines the key components of the framework, including datasets, performance metrics, baseline logic mining models for comparison, and parameter settings utilized during experimentation. The setup is designed to ensures reliable and reproducible results while demonstrating the flexibility and effectiveness of the proposed model.

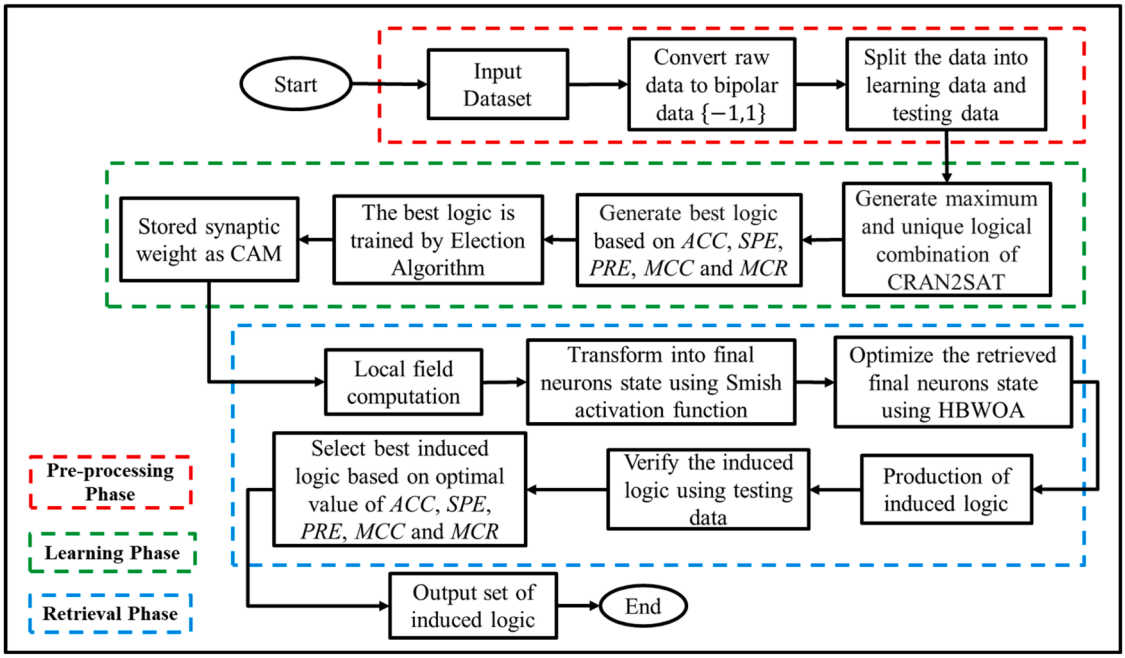


**Fig. 4.** The overall flowchart of CR2SATRA.

### 6.1. Dataset description

To evaluate the proposed logic mining model, this paper uses 20 repository datasets retrieved from two reputable machine learning repositories: the UCI Machine Learning Repository (https://archive.ics.uci.edu/datasets) and Kaggle (https://www.kaggle.com/datasets). These datasets commonly used in classification and prediction tasks.

Table 1 summarizes the datasets used in the experiment and presents the data distribution for each dataset, including the number of positive and negative outcomes. This provides quantitative insight into the class imbalance characteristics of the datasets. Several datasets show different levels of class imbalance allowing the proposed logic mining model to be evaluated under different imbalance conditions. To minimize potential biases common to specific fields, the 20 datasets were chosen from a different range of disciplines including Health, Medicine, Economics, Business, Physics, Chemistry, Social Sciences, Computer Science and Education. Several criteria were considered when in choosing these datasets for logic mining analysis.

First, the dataset must have at least 11 features or variables to ensure optimal attribute selection [20]. Second, the dataset must contain at least 100 instances. According to [20], a minimum of 100 entries is required to preserve the statistical properties of the data distribution.

**Table 1**
The datasets used in the experiment.

| Dataset Code | Name of Dataset | Attributes | Instances | Missing Value | Type of Data | Field | No. of Positive Outcome | No. of Negative Outcome |
|---|---|---|---|---|---|---|---|---|
| D1 | Autistic Disorder | 21 | 292 | No | Mixed | Health | 150 | 142 |
| D2 | Body Signal of Smoking | 26 | 5143 | No | Mixed | Health | 1890 | 3253 |
| D3 | Real Estate Data Set | 14 | 511 | Yes (0.98%) | Mixed | Economics | 208 | 303 |
| D4 | Hungarian Chickenpox | 21 | 522 | No | Quantitative | Health | 192 | 330 |
| D5 | Ionosphere | 35 | 351 | No | Quantitative | Physics and Chemistry | 225 | 126 |
| D6 | Adult | 15 | 32,561 | No | Mixed | Social Sciences | 7841 | 24,720 |
| D7 | Airline Passenger Satisfaction | 21 | 1004 | Yes (0.01%) | Mixed | Social Sciences | 459 | 545 |
| D8 | Divorce Predictor | 55 | 170 | No | Quantitative | Social Sciences | 84 | 86 |
| D9 | Autism Screening | 21 | 704 | Yes (0.28%) | Mixed | Medical | 189 | 515 |
| D10 | Bank Marketing | 21 | 4119 | No | Mixed | Business | 451 | 3668 |
| D11 | Cylinder Bands | 37 | 541 | Yes (0.18%) | Mixed | Physics and Chemistry | 229 | 312 |
| D12 | Leaf | 16 | 340 | No | Quantitative | Computer Science | 169 | 171 |
| D13 | Behaviour of Urban Traffic | 18 | 135 | No | Mixed | Computer Science | 56 | 79 |
| D14 | Asianpaint | 11 | 5306 | Yes (0.19%) | Mixed | Investing | 2825 | 2480 |
| D15 | Brain Stroke | 11 | 4980 | No | Mixed | Medical | 248 | 4732 |
| D16 | Diabetes Classification | 15 | 390 | No | Mixed | Medical | 60 | 330 |
| D17 | Predict Students' Dropout | 37 | 4424 | No | Mixed | Education | 3003 | 1421 |
| D18 | IBM HR Analytics Employee Attrition | 35 | 1470 | No | Mixed | Business | 237 | 1233 |
| D19 | Room Occupancy Estimation | 17 | 10,129 | No | Quantitative | Computer Science | 1901 | 8228 |
| D20 | Image Segmentation | 17 | 210 | No | Mixed | Computer Science | 51 | 159 |

Datasets with small entries may limit the training capability and performance assessment of the logic mining model. This experiment was conducted in a discrete manner, where dataset entries were stored in neurons using bipolar representations $\{-1, 1\}$. The Statistical Package for Social Sciences (SPSS) software was used to convert datasets into bipolar form. For categorical attributes, frequency distribution analysis was applied to transform qualitative data into bipolar representations. For numerical attributes, values were assigned 1 or $-1$ based on average thresholds determined using *k*-means clustering in SPSS, following the approach in [16] and [17]. The missing values were handled by replacing them with 1 or $-1$ based on the lowest frequency in the bipolar representation as described in [22]. This strategy helps reduce bias introduced by incomplete entries, particularly in imbalanced datasets. To minimize the impact of missing values on logic generation, datasets with minimal missing entries were prioritized. For a fair comparison with existing logic mining models, a train-test split method was employed using 60% of the dataset for learning and 40% for testing. Cross-validation was also applied to repository datasets to explore a broader search space [24]. Following [14], five-fold cross-validation was used with the average across all folds representing the result for all metrics. The next section details the performance evaluation of the proposed multi-unit logic mining model.

### 6.2. Performance metrics

In this section, the performance metrics used to assess the effectiveness of the proposed multi-unit logic mining model will be discussed [38] mentioned that a confusion matrix for discrete classification is a two-by-two table that captures the four possible outcomes of a discrete classifier. Therefore, this paper focusses on the confusion matrix as the primary performance evaluation metric. The outcomes of the confusion matrix in term of $TP$, $TN$, $FP$ and $FN$ will be used in both learning phase and retrieval phase. Various performance metrics can be calculated based on the confusion matrix using $TP$, $TN$, $FP$ and $FN$. It is worth noting that the first work that utilized various performance metrics in a logic mining approach is proposed by [20].

In evaluating the performance of the logic mining model, this paper uses five performance metrics to measure the robustness of the proposed logic mining model compared to existing logic mining models. The effectiveness of the proposed logic mining model in performing classification tasks is assessed through metrics such as$ACC$,$PRE$,$SPE$,$MCC$, and $MCR$. $ACC$ metric is a widely used performance metric and is calculated using Eq. (62) which divides the number of correct predictions (true predictions) by the total number of samples predicted by the model [39], [40] also describe $ACC$ as the ratio of correct predictions to total predictions. The range value of $ACC$ ranges from 0 to 1 with higher values indicating that the model correctly predicts the majority $TP$ and $TN$.$ACC$ is a commonly used across various research fields [22,38,40], representing the overall accuracy of a model by considering both true positives and true negatives.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{62}$$

While the formulation of $ACC$ considers both true outcomes, the formulation of $PRE$ in Eq. (63) focuses solely on one outcome, which is the true positive predictions. $PRE$ also known as Positive Predictive Value which quantifies the accuracy of the positive prediction $(TP)$ obtained by the model [41]. The range value of $PRE$ ranges from 0 to 1, where a higher $PRE$ indicates that the model is effective at making true positive predictions $(TP)$ while minimizing the false positive predictions $(FP)$.

$$PRE = \frac{TP}{TP + FP} \tag{63}$$

In classification problems, $PRE$ and $SPE$ are important metrics for evaluating performance of the model. This is because both $PRE$ and $SPE$ focus on specific types of outcomes. While $PRE$ focuses on the $TP$, $SPE$ focuses on true negative predictions $(TN)$. $SPE$ also known as True Negative Rate, which measures the ability of the model to accurately identify $TN$. The formulation of $SPE$ in Eq. (64) represents the ratio of $TN$ to the total number of $TN$ and $FN$. The range value of $SPE$ ranges from 0 to 1, where a higher $SPE$ indicates that the model effectively predicts true negative predictions $(TN)$ while minimizing the false positive predictions $(FP)$.

$$SPE = \frac{TN}{TN + FP} \tag{64}$$

Another important metrics for analyzing the performance of the logic mining model is $MCC$. By using $MCC$, the performance of the logic mining model is evaluated based on eight key ratios derived from the combination of all components of the confusion matrix as shown in Eq. (65). $MCC$ is considered as a good metric that represents the overall quality of the model and can be applied to classes of different sizes [37]. According to [14], the purpose of $MCC$ analysis is to determine whether the logic mining model performs similarly to, worse than or better than the random classifier model.

$$MCC = \frac{TPTN - FPFN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{65}$$

Lastly, the Misclassification Rate $(MCR)$ measures incorrect predictions through false positive predictions $(FP)$ and false negative predictions $(FN)$. According to the research conducted by [42] on the classification of Diabetic Retinopathy using Convolutional Neural Networks, a lower $MCR$ indicates a better performance in accurately classifying fundus image data. This demonstrates that the model effectively reduces the errors in prediction where true outcomes are incorrectly classified as false positives or false negatives. Additionally, according to [40], $MCR$ is the ratio of incorrect predictions to the total predictions with values ranging from 0 to 1. A lower $MCR$demonstrates the ability of the model to reduce misclassifications of outcomes as $(FP)$ and $(FN)$. The formulation of $MCR$ as shown in Eq. (66):

$$MCR = \frac{FP + FN}{TP + TN + FP + FN} \tag{66}$$

To provide a comparative analysis and highlight the superiority of CR2SATRA, the next section will introduce and discuss a selection of baseline logic mining models. These baseline logic mining models will serve as benchmarks for evaluating the performance of CR2SATRA.

### 6.3. Baseline model

In this paper, the proposed logic mining model CR2SATRA is compared with several baseline logic mining models. To ensure a fair comparison in generating induced logics, baseline models that employ a single Content Addressable Memory (CAM) unit within the DHNN framework are selected. Although [22] and [23] introduced advanced multi-unit logic mining models that select the best logic based on the maximization of $TP + TN$, these approaches primarily focus on this single evaluation criterion. In contrast, CR2SATRA evaluates the best logic using five different confusion-matrix. This allows multiple induced logics to be generated from different evaluation perspectives, providing greater flexibility in representing dataset behavior. It should also be noted that this paper excludes comparisons with r2SATMRA [14] and 2HESRA [24], as these models utilize multiple CAM units within DHNN, whereas the proposed CR2SATRA employs a single CAM unit. This restriction ensures that the comparison focuses on the effectiveness of the proposed CR2SATRA rather than differences in computational capacity. Based on these considerations, the performance of CR2SATRA is evaluated against the following baseline logic mining models.

(a) RA [15]: RA is a logic mining model that used Horn Satisfiability (HornSAT) logical rules to extract logical rules from real-life

datasets. To ensure the comparability of the results, modifications were made to the conventional RA. Instead of assigning a neuron to each instance, as in the original model, this paper assigns neurons to attribute. Consequently, the WA method is employed to determine the synaptic weight of the neuron responsible for the final induced logic. In RA, the induced logic must satisfy the HornSAT condition, requiring each clause to contain both a positive and a negative literal.

(b) 2SATRA [16]: The 2 Satisfiability Reverse Analysis method (2SATRA) is based on the 2SAT logical rule. In 2SATRA, random attribute selection is used to represent the 2SAT logical rule. 2SATRA is the first logic mining model that attempt to obtain the best logic during the learning phase by maximizing the frequency of clauses with an outcome of 1, max[$TP$]. Consequently, WA method is used to determine the optimal value of the synaptic weight for generating the best induced logic.

(c) P2SATRA [19]: P2SATRA is a logic mining model that incorporates a permutation operator. This operator expands the solution space during the DHNN retrieval phase by allowing various arrangements of attributes. This approach increases the probability of obtaining an optimal induced logic with high Accuracy. The model permutes the attribute arrangement while ensuring that the attributes remain non-redundant.

(d) E2SATRA [18]: E2SATRA is a logic mining model that uses an energy-based approach to determine final neuron states that correspond to global minimum energy. During the retrieval phase of DHNN, an energy operator transforms final neuron states into induced logic focusing solely on those achieve global minimum energy. Unlike other models, E2SATRA does not use a permutation operator. Additionally, this model utilizes random attribute selection to choose the attributes representing the logical rule.

(e) 3SATRA [17]: 3SATRA is a higher-order logic mining model that uses 3SAT logical rules to extract patterns from real-life datasets. The logic mining model of 3SATRA does not employ permutation operators and relies on random attribute selection to represent logical rules. In addition, the learning phase of 3SATRA is utilizing Clonal Selection Algorithm (CSA) as a training algorithm and the selection of the best induced logic is obtained by the highest Accuracy value.

(f) S2SATRA [20]: S2SATRA is the first supervised logic mining model that utilizes statistical analysis and correlation to identifying the most relevant attributes for dataset representation. The Pearson chi-square associations between variables are analyzed to ensure that attributes with strong correlations are selected before being processed in DHNN. As an extension of existing logic mining models, S2SATRA which capitalize second-order logical rule employs a permutation operator and focuses solely on induced logic that achieves highest accuracy value.

(g) A2SATRA [21]: A2SATRA is another supervised logic mining model that employs log linear analysis to determine which attributes should be selected for representing second-order logical rules before they can be learned by the network. By using log linear analysis, only attributes that have strong association between the attributes will be selected to represent the logical rules. Additionally, it is also worth noting that A2SATRA also implements permutation operator in its logic mining process.

The next section will present a detailed evaluation of the proposed logic mining model focusing on the key parameters used to assess the performance of CR2SATRA. Additionally, to provide a comparative analysis and highlight the strengths and superiority of CR2SATRA, the next section will discuss the performance analysis as compared to baseline logic mining models.

### 6.4. Parameter assignment

An optimized learning phase increases the probability of getting satisfied interpretation of $P_{CRAN2SAT}$ leading to an optimal value of the synaptic weights. EA has shown a successful result as learning algorithm which can effectively minimize the cost function of the network and consequently lead to an optimal final neuron state. Thus, this paper implements EA as a learning algorithm to secure the optimal retrieved neuron state. The parameter settings for optimizing DHNN—CRAN2SAT are shown in Table 2. The parameters of HBWOA, including the number of generations and clone rate are adopted from previous studies where the algorithm demonstrated stable and effective performance. These settings are retained to ensure consistent and reliable optimization within the DHNN-based logic mining framework.

This section outlines the parameters used to evaluate the performance of CR2SATRA as compared with baseline logic mining models. Table 3 lists the parameters for CR2SARTRA versus the baseline logic mining models. The parameter setup ensure comparability with the baseline logic mining models and various configurations are applied to preserve their originality. During the pre-processing phase, all models follow similar configurations within a 5-fold cross-validation scheme to minimize potential bias. In each fold, the dataset is divided into 60% training data and 40% testing data. Both the training and testing partitions change across folds to ensure that all instances are evaluated during the validation process. The baseline logic mining models use ES as the learning algorithm and do not apply optimization during the retrieval phase of DHNN. In contrast, CR2SATRA uses the Smish activation function as the updating rule and utilizes multi-objective optimization in the retrieval phase. Instead of generating best logic based solely on the highest frequency of clauses with an outcome of 1, denoted as $P_{learn} = 1$ or max[$TP$], CR2SATRA enhances the search space by using different pathway in generating the best logic.

Moreover, while other models focus on training a single best logic, CR2SATRA generates multiple top-performing best logics $\left(P^{best}_{CR2SATRA}\right)$ based on five different confusion matrixes ($\rho = 5$). CR2SATRA also undergoes an optimization process using HBWOA during the retrieval phase, unlike the baseline models. The simulation runs 100 trials to produce a list of induced logics $\left(P^{induced}_{CR2SATRA}\right)$, from which the best induced logic $\left(P^{I}_{best}\right)$is selected based on the highest values of $ACC$, $PRE$, $SPE$, $MCC$ and $MCR$. In contrast, baseline models typically select the best logic based solely on the highest $ACC$ value. The following section presents a comparative performance analysis of CR2SATRA against selected baseline logic mining models. In the next section, the effectiveness of the proposed logic mining model is compared through the analysis of different performance metrics.

## 7. Results and discussion

This section evaluates the performance of the proposed CR2SATRA using benchmark datasets. The results are compared with baseline logic mining models to highlight the superior performance of CR2SATRA in doing classification task and knowledge extraction. Optimal results will demonstrate CR2SATRA as a robust classification model. By employing a new approach in generating the best logic and utilizing $RTFSM$ for

**Table 2**
Parameter settings for optimizing $P_{CRAN2SAT}$ using HBWOA.

| Setting | Parameter Value |
|---|---|
| Learning algorithm | EA [29] |
| Activation function | Smish activation function [43] |
| Updating the neuron state involved | Second-order clause |
| Number of generations in HBWOA | 100 |
| Clone rate ($CR$) | 0.5 [17] |
| Percentage of negativity | 50% |
| Transfer function | Sigmoid function |

**Table 3**
The list of parameters in CR2SATRA versus baseline model.

| Parameter | Logic mining model | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | CR2SATRA | RA | 2SATRA | 3SATRA | E2SATRA | P2SATRA | S2SATRA | A2SATRA |
| Attribute selection method | *RTFSM* | Random | Random | Random | Random | Random | Correlation | Log-linear |
| Types of logical rule | Non-systematic | Systematic | Systematic | Systematic | Systematic | Systematic | Systematic | Systematic |
| Criterion to select $P^{best}$ | max[*ACC*], max[*PRE*], max[*SPE*], max[*MCC*], max[*MCR*] | max[*TP*] | max[*TP*] | max[*TP*] | max[*TP*] | max[*TP*] | max[*TP*] | max[*TP*] |
| Best logics trained by DHNN | 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Number of attributes | 9 | 6 | 6 | 9 | 6 | 6 | 6 | 6 |
| Learning algorithm | EA | ES | ES | ES | ES | ES | ES | ES |
| Retrieval algorithm | HBWOA | NA | NA | NA | NA | NA | NA | NA |
| Number of trial | 100 | 10,000 | 100 | 100 | 10,000 | 100 | 100 | 100 |
| Neuron Combination | 253 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Permutation Operator | 10 | NA | NA | NA | NA | 100 | 100 | 100 |
| Criterion to select best $P_{best}^{induced}$ | *ACC*, *PRE*, *SPE*, *MCC*, *MCR* | *ACC* | *ACC* | *ACC* | *ACC* | *ACC* | *ACC* | *ACC* |

attribute selection method, the performance of CR2SATRA is evaluate across five performance metrics including *ACC*, *PRE*, *SPE*, *MCC* and *MCR*.

### 7.1. Accuracy

The result in Table 4 show the *ACC* values obtained by CR2SATRA through 5-fold cross-validation compared to baseline logic mining models. The (*) in the brackets indicates no ratio of improvement since the respective dataset has no value*ACC*. High *ACC* values imply that CR2SATRA accurately predicts both *TP* and *TN* entries. As shown in Table 4, CR2SATRA achieves a significant advancement with an average *ACC* of 82.75% across 20 datasets outperforming baseline models in 19 out of 20 datasets. The improvement ratio in Table 4 demonstrates how CRAN2SAT enhances the retrieval of true outcomes. The main difference between CR2SATRA and baseline models lies in generating the best logic. CR2SATRA selects the top five highest ACC values to determine the best logic, whereas existing models rely on the highest frequency of TP clauses. This can result in biased solutions in imbalanced datasets. In contrast, the approach in CR2SATRA expands the search space and avoids the limitations of training only a single best logic which improves

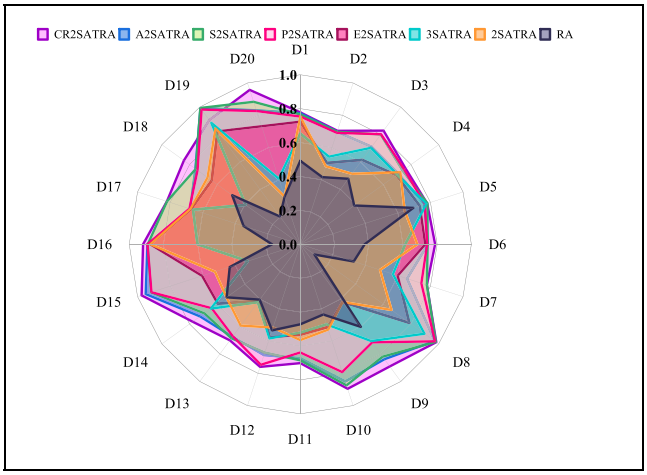


**Fig. 5.** The values of *ACC* for all logic mining models.

**Table 4**
The values of *ACC* for all logic mining models. The brackets indicate the improvement ratio and the negative ratio indicate that baseline model outperformed the proposed model.

| Dataset | CR2SATRA | 2SATRA | P2SATRA | E2SATRA | RA | 3SATRA | S2SATRA | A2SATRA |
|---|---|---|---|---|---|---|---|---|
| D1 | **0.7778** | 0.7538 (0.0318) | 0.7538 (0.0318) | 0.7231 (0.0756) | 0.4923 (0.5799) | 0.6530 (0.1911) | 0.7658 (0.0157) | 0.7726 (0.0067) |
| D2 | **0.7038** | 0.4837 (0.4550) | 0.6912 (0.0182) | 0.5048 (0.3942) | 0.4165 (0.6898) | 0.5433 (0.2954) | 0.6976 (0.0089) | 0.6976 (0.0089) |
| D3 | **0.8284** | 0.5137 (0.6126) | 0.8029 (0.0318) | 0.6157 (0.3455) | 0.4775 (0.7349) | 0.7029 (0.1785) | 0.8000 (0.0355) | 0.7108 (0.1654) |
| D4 | **0.7608** | 0.7225 (0.0530) | 0.7512 (0.0128) | 0.6938 (0.0966) | 0.3895 (0.9533) | 0.6986 (0.0890) | 0.7397 (0.0285) | 0.7091 (0.0729) |
| D5 | **0.7886** | 0.6414 (0.2295) | 0.7657 (0.0299) | 0.7343 (0.0739) | 0.6946 (0.1353) | 0.7757 (0.0166) | 0.7814 (0.0092) | 0.7614 (0.0357) |
| D6 | **0.7883** | 0.6842 (0.1521) | 0.7418 (0.0627) | 0.7301 (0.0797) | 0.3761 (1.0960) | 0.6135 (0.2849) | 0.7444 (0.0590) | 0.7458 (0.0570) |
| D7 | **0.7781** | 0.4900 (0.5880) | 0.7443 (0.0454) | 0.5990 (0.2990) | 0.3284 (1.3694) | 0.5697 (0.3658) | 0.7781 (0.0000) | 0.6567 (0.1849) |
| D8 | **0.9824** | 0.6588 (0.4912) | 0.9735 (0.0091) | 0.7882 (0.2464) | 0.1029 (8.5471) | 0.8971 (0.0951) | 0.9794 (0.0031) | 0.9706 (0.0122) |
| D9 | **0.8904** | 0.4206 (1.1170) | 0.7149 (0.2455) | 0.4206 (1.1170) | 0.6021 (0.4788) | 0.7050 (0.2630) | 0.8191 (0.0870) | 0.8376 (0.0630) |
| D10 | **0.8960** | 0.5291 (0.6934) | 0.7924 (0.1307) | 0.5143 (0.7422) | 0.4354 (1.0579) | 0.4970 (0.8028) | 0.8748 (0.0242) | 0.8507 (0.0533) |
| D11 | **0.7010** | 0.5648 (0.2411) | 0.6380 (0.0987) | 0.5333 (0.3145) | 0.4722 (0.4845) | 0.5213 (0.3447) | 0.6843 (0.0244) | 0.6722 (0.0428) |
| D12 | **0.7603** | 0.5162 (0.4729) | 0.7471 (0.0177) | 0.5765 (0.3188) | 0.5338 (0.4243) | 0.5838 (0.3023) | 0.6721 (0.1312) | 0.6868 (0.1070) |
| D13 | **0.7000** | 0.5926 (0.1812) | 0.6704 (0.0442) | 0.4111 (0.7027) | 0.4037 (0.734) | 0.4222 (0.6580) | 0.6852 (0.0216) | 0.6741 (0.0384) |
| D14 | **0.7771** | 0.5330 (0.4580) | 0.6406 (0.2131) | 0.5961 (0.3036) | 0.5324 (0.4596) | 0.6463 (0.2024) | 0.6927 (0.1218) | 0.7239 (0.0735) |
| D15 | **0.9755** | 0.5241 (0.8613) | 0.9142 (0.0671) | 0.6032 (0.6172) | 0.4337 (1.2493) | 0.3236 (2.0145) | 0.9175 (0.0632) | 0.9502 (0.0266) |
| D15 | **0.9755** | 0.5241 (0.8613) | 0.9142 (0.0671) | 0.6032 (0.6172) | 0.4337 (1.2493) | 0.3236 (2.0145) | 0.9175 (0.0632) | 0.9502 (0.0266) |
| D16 | **0.9192** | 0.8769 (0.0482) | 0.8936 (0.0286) | 0.8769 (0.0482) | 0.1654 (4.5574) | 0.6013 (0.5287) | 0.8910 (0.0316) | 0.8936 (0.0286) |
| D17 | **0.8195** | 0.6750 (0.2141) | 0.6814 (0.2027) | 0.6714 (0.2206) | 0.3486 (1.3508) | 0.6597 (0.2422) | 0.8189 (0.0007) | 0.6773 (0.2100) |
| D18 | **0.8408** | 0.6701 (0.2547) | 0.7422 (0.1328) | 0.6398 (0.3142) | 0.4922 (0.7082) | 0.3952 (1.1275) | 0.7554 (0.1131) | 0.7578 (0.1095) |
| D19 | 0.9054 | 0.8436 (0.0733) | 0.9816 (−0.0776) | 0.8228 (0.1004) | 0.2033 (3.4535) | 0.8851 (0.0229) | **0.9936** (−0.0888) | 0.9928 (−0.0880) |
| D20 | **0.9571** | 0.2952 (2.2422) | 0.8262 (0.1584) | (*) | 0.2952 (2.2422) | 0.4071 (1.3510) | 0.8833 (0.0836) | 0.8301 (0.1517) |
| (+ / = / −) | | 20/0/0 | 20/0/0 | 20/0/0 | 20/0/0 | 20/0/0 | 18/1/1 | 19/0/1 |
| **Max** | 0.9824 | 0.8769 | 0.9816 | 0.8769 | 0.6946 | 0.8971 | **0.9936** | 0.9928 |
| **Min** | 0.7000 | 0.2952 | 0.6380 | 0.4111 | **0.1029** | 0.3236 | 0.6721 | 0.6567 |
| **Average** | **0.8275** | 0.5995 | 0.7734 | 0.6345 | 0.4098 | 0.6051 | 0.7987 | 0.7786 |

the quality of the induced logic.

Fig. 5 shows high *ACC* values across of imbalanced datasets indicating that generating the best logic based on the top five highest *ACC* values trained by the network is effective. This approach expands the search space during the learning phase of the DHNN and enhances the generalization of optimal synaptic weights. Additionally, CR2SATRA uses EA as a learning algorithm ensuring an effective learning phase led to an optimal synaptic weight. In contrast, the baseline logic mining models use ES as a learning algorithm which limits their ability to retrieve optimal induced logic. By utilizing the optimal synaptic weights from different best logics, CR2SATRA can retrieve the most effective induced logic. Furthermore, the updating rule within DHNN is more effective with the Smish activation function, which improves the mapping of symbolic induced logic to testing data, minimizing *FP* and *FN*. This improvement is due to the capability of Smish activation function to retrieve a diversified final neuron state. In contrast, baseline logic mining models use HTAF which leads to less diversified final neuron states due to the vanishing gradient problem [44].

[45] state that a high *ACC* value reflects the ability of the model to accurately predict true outcomes. CR2SATRA achieved perfect accuracy across 20 datasets outperforming existing models such as E2SATRA, 3SATRA, 2SATRA and RA These models which do not include a permutation operator and rely on random attribute selection, limit the interpretability of neuron connections within the trained logical rules in DHNN. Therefore, their induced logic tends to produce misclassifications, which results in higher *FP* and *FN*. Despite the success of CR2SATRA, baseline logic mining model of A2SATRA, S2SATRA and P2SATRA. achieved higher *ACC* values for dataset D19 due to its high ratio of negative outcomes. Additionally, since CR2SATRA emphasizes the non-systematic logic of second-order and first-order clauses, particularly the presence of the first-order clause increased the chances of false negatives, eventually reducing the *ACC* values. Notably, CR2SATRA achieved the highest *ACC* value of 98.24% indicating its ability to predict the majority of *TP* and *TN* in the testing datasets. This demonstrates the superior ability of CR2SATRA to handle imbalanced datasets when compared to baseline models. However, relying exclusively on *ACC* does not provide a complete performance evaluation of the logic mining model, [46], emphasize that *ACC* is not an ideal metric for evaluating the classification outcomes *TP*, *TN*, *FP* and *FN* in imbalanced datasets.

### 7.2. Precision

The *PRE* results for all logic mining models analyzed across 20 datasets using 5-fold cross-validation are presented in Table 5. The bracket indicates in these tables show the improvement ratio indicating how CRAN2SAT enhances the accurate classification of positive instances compared to baseline models. CR2SATRA demonstrates the highest *PRE* for most datasets with an average value of 91.16%. According to [47]*PRE* values help researchers assess the reliability of a model in classifying positive instances. By generating the best logic based on the top five highest *PRE* values, CR2SATRA outperforms baseline models by achieving $PRE = 1$ for 12 out of 20 datasets. A $PRE = 1$signifies that the model accurately classifies all positive instances, minimizing *FP* and ensuring no negative instances are misclassified. High *PRE*values also correlate with high *TN*values, highlighting the effectiveness of CR2SATRA in correctly identifying negative outcomes.

One of the possible reasons why CR2SATRA achieves high classification of *TN* is caused by the non-systematic structure of $P_{CRAN2SAT}$ which utilizes first-order and second-order clauses. The existence of first-order clauses will disrupt the classification process because the first-order clauses have high tendency of getting negative outcomes. This statement is supported by Zamri et al. (2024), when $PRE = 1$, the first-order logic will influence the retrieved induced logic boosts the interpretability of the logic mining model. These findings support the idea that the non-systematic component of the logical rule provides advantages over those used in other logic mining models. The flexibility of using different orders within each clause allows CR2SATRA to better capture dataset behaviour. Additionally, the combination of different orders influences the values of *TP* and *TN* resulting in low *FP* and high *TN*. This enables CR2SATRA to effectively identify attributes that make significant contributions to the outcome. As shown in Fig. 6, CRAN2SAT

**Table 5**
The values of *PRE* for all logic mining models. The brackets indicate the improvement ratio and the negative ratio indicate that baseline model outperformed the proposed model.

| Dataset | CR2SATRA | 2SATRA | P2SATRA | E2SATRA | RA | 3SATRA | S2SATRA | A2SATRA |
|---|---|---|---|---|---|---|---|---|
| D1 | **1.0000** | 0.8942 (0.1183) | 0.8942 (0.1183) | 0.7535 (0.3271) | 0.6752 (0.4810) | 0.9235 (0.0828) | 0.9135 (0.0947) | 0.9041 (0.1061) |
| D2 | **0.9793** | 0.3978 (1.4618) | 0.9537 (0.0268) | 0.4095 (1.3915) | 0.4198 (1.3328) | 0.7036 (0.3918) | 0.9513 (0.0294) | 0.9513 (0.0294) |
| D3 | **1.0000** | 0.4146 (1.4120) | 0.5656 (0.768) | 0.4997 (1.0012) | 0.4619 (1.1650) | 0.8500 (0.1765) | 0.5906 (0.6932) | 0.4235 (1.3613) |
| D4 | **1.0000** | 0.5852 (0.7088) | 0.6398 (0.5630) | 0.5770 (0.7331) | 0.3428 (1.9172) | 0.5676 (0.7618) | 0.6062 (0.6496) | 0.5786 (0.7283) |
| D5 | **1.0000** | **1.0000** (0.0000) | 0.9057 (0.1041) | 0.8254 (0.2115) | 0.8384 (0.1927) | 0.9183 (0.0890) | 0.8745 (0.1435) | 0.8830 (0.1325) |
| D6 | **0.9242** | 0.1960 (3.7153) | 0.3273 (1.8237) | 0.1497 (5.1737) | 0.4463 (1.0708) | 0.8234 (0.1224) | 0.6424 (0.4387) | 0.1730 (4.3422) |
| D7 | **1.0000** | 0.6494 (0.5399) | 0.7522 (0.3294) | 0.6341 (0.577) | 0.3978 (1.5138) | 0.7904 (0.2652) | 0.8059 (0.2408) | 0.5341 (0.8723) |
| D8 | 0.8000 | 0.4794 (0.6688) | **0.9779** (−0.1819) | 0.7013 (0.1407) | 0.2881 (1.7768) | 0.9463 (−0.1546) | 0.9706 (−0.1758) | 0.9693 (−0.1747) |
| D9 | **1.0000** | 0.9949 (0.0051) | 0.9698 (0.0311) | 0.9949 (0.0051) | 0.6587 (0.5181) | 0.9118 (0.0967) | 0.8855 (0.1293) | 0.7319 (0.3663) |
| D10 | **1.0000** | 0.5297 (0.8879) | 0.0765(12.0719) | 0.5083 (0.9673) | 0.5492 (0.8208) | 0.6410 (0.5601) | 0.1072 (8.3284) | 0.1928 (4.1867) |
| D11 | **1.0000** | 0.4440 (1.2523) | 0.4966 (1.0137) | 0.4019 (1.4882) | 0.5924 (0.6880) | 0.8270 (0.2092) | 0.2365 (3.2283) | 0.300 (2.3333) |
| D12 | **0.8000** | 0.5154 (0.5522) | 0.7128 (0.1223) | 0.4037 (0.9817) | 0.5775 (0.3853) | **0.9186** (−0.1291) | 0.6745 (0.1861) | 0.7559 (0.0583) |
| D13 | **1.0000** | 0.2774 (2.6049) | 0.4290 (1.3310) | 0.4583 (1.1820) | 0.5699 (0.7547) | 0.9826 (0.0177) | 0.4440 (1.2523) | 0.3702 (1.7012) |
| D14 | 0.8588 | **0.9990** (−0.1403) | 0.8134 (0.0558) | 0.6117 (0.4040) | 0.9984 (−0.1398) | 0.8392 (0.0234) | 0.7239 (0.1864) | 0.7750 (0.1081) |
| D15 | 0.4000 | **0.9571** (−0.5821) | 0.1856 (1.1552) | 0.8014 (−0.5009) | 0.4773 (−0.1620) | 0.8233 (−0.5142) | 0.1335 (1.9963) | (*) |
| D16 | **1.0000** | 0.9044 (0.1057) | 0.5444 (0.8369) | 0.9044 (0.1057) | 0.4720 (1.1186) | 0.7078 (0.4128) | 0.4668 (1.1422) | 0.5444 (0.8369) |
| D17 | **1.0000** | 0.9668 (0.0343) | 0.9938 (0.0062) | 0.9106 (0.0982) | 0.2097 (3.7687) | 0.9167 (0.0909) | 0.9205 (0.0864) | 0.9685 (0.0325) |
| D18 | **0.8889** | 0.2169 (3.0982) | 0.1027 (7.6553) | 0.4323 (1.0562) | 0.5986 (0.4850) | 0.7823 (0.1363) | 0.2273 (2.9107) | 0.1792 (3.9604) |
| D19 | **1.0000** | 0.9465 (0.0565) | 0.6443 (0.5521) | 0.7856 (0.2729) | 0.8222 (0.2162) | 0.8973 (0.1145) | 0.9327 (0.0722) | 0.9296 (0.0757) |
| D20 | 0.5812 | 0.9875 (−0.4114) | 0.025 (22.2480) | (*) | 0.9875 (−0.4114) | **0.9938** (−0.4152) | 0.5500 (0.0567) | 0.0250(22.2480) |
| (+/=/−) | | 16/1/3 | 19/0/1 | 19/0/1 | 17/0/3 | 16/0/4 | 19/0/1 | 19/0/1 |
| **Max** | **1.0000** | **1.0000** | 0.9938 | 0.9949 | 0.9984 | 0.9938 | 0.9706 | 0.9693 |
| **Min** | 0.4000 | 0.1960 | **0.0250** | 0.1497 | 0.2097 | 0.5676 | 0.1072 | **0.0250** |
| **Average** | **0.9116** | 0.6678 | 0.6005 | 0.6191 | 0.5692 | 0.8382 | 0.6329 | 0.5889 |

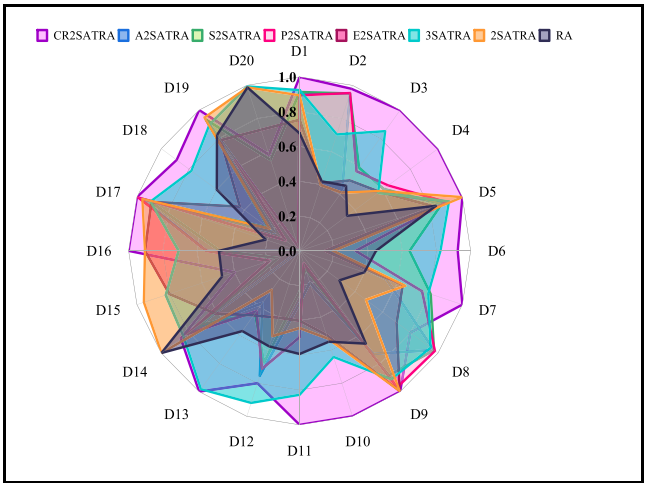


**Fig. 6.** The values of *PRE* for all logic mining models.

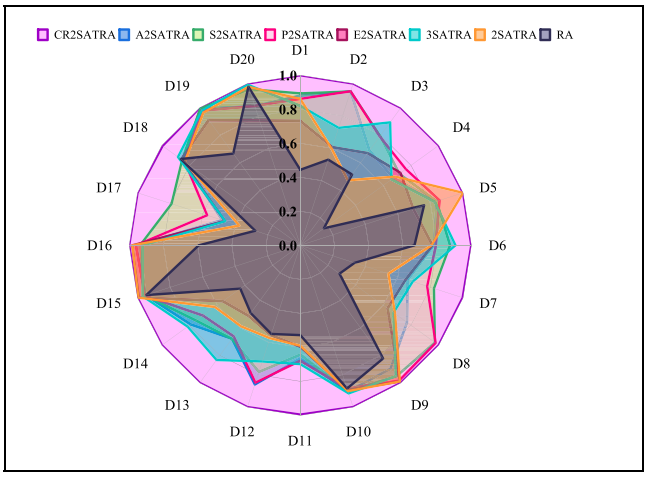


**Fig. 7.** The values of *SPE* for all logic mining models.

minimizes *FP* without misclassifying the negative samples.

The superiority of CR2SATRA in generating best logic using *PRE* is attributed to the logical structure of $P_{CRAN2SAT}$. By incorporating at least one negative literal within the second-order clauses, $P_{CRAN2SAT}$ effectively handles datasets with a high proportion of negative outcomes. In contrast, baseline models use a randomized distribution of positive and negative literals which can impact their performance. Additionally, the use of HBWOA to optimize the final neuron states enhances the capability of CR2SATRA in knowledge extraction. This results in induced logic with highly diversified neuron states which improve the ability of CR2SATRA to accurately classify negative outcome. As supported by Zamri et al. (2024), diversifying final neuron states is crucial for understanding all dataset entries. CR2SATRA uses the Smish activation function to improve the mapping between the symbolic induced logic and testing data which in turns lead to a minimizing both *FP* and *FN*.While 3SATRA achieves the second-highest average of *PRE* at 83.82%. The main difference between 3SATRA and CR2SATRA lies in its logical structure and the absence of a permutation operator. Although the structure of 3SATRA increases the likelihood of *TP*, the absence of a permutation operator limits connectivity between attributes and reduce the quality of the induced logic. This limitation affects the ability of 3SATRA to minimize *FP* and *FN* which impacting its overall performance. The results show that CRAN2SAT competes with existing logic mining model E2SATRA, 3SATRA, 2SATRA and RA on datasets D14, D15 and D20. These baseline models share a systematic logical rule which provides optimal neuron connections for positive classifications that lead to high *TP* and low *FP*. Despite this competition, CR2SATRA outperforms these models, E2SATRA, 3SATRA, 2SATRA and RA on the remaining datasets. By generating the best logic based on the top five highest *PRE* values, CR2SATRA effectively visualizes the reliability of its positive classifications.

### 7.3. Specificity

Fig. 7 illustrate the *SPE* values of the induced logic obtained by CR2SATRA compared to other models. CR2SATRA outperforms the baseline models with the highest average specificity of 99.94%. According to a [48], *SPE*values reflect the accuracy of the model in identifying *TN*. CR2SATRA achieves $SPE = 1$ for 16 out of 20 datasets which highlighting its superior performance. One reason for this high *TN* classification is because of the non-systematic structure of $P_{CRAN2SAT}$ in CR2SATRA model which utilizes first-order and second-order clauses. The existence of first-order clauses will disturb the classification process because the first-order clauses have high tendency of getting negative outcomes. This is supported by the nature of the benchmark datasets used in this paper, with a majority of negative outcomes $(-1)$and a minority of positive outcomes $(1)$. Even though not all datasets are highly imbalanced, CR2SATRA demonstrates a strong ability to correctly classify *TN* rather than misclassifying them as positive, *FP*.

To maximize *SPE*, a logic mining model must accurately classify a high *TN* while minimizing *FP*. Despite of the non-systematic logical structure of $P_{CRAN2SAT}$, the superiority of CR2SATRA in *SPE* is also due to the effectiveness of HBWOA in generating diversified final neuron state, with at least 50% of the clauses $P_{CRAN2SAT}$ containing at least one negative literal. The inclusion of more negative literals enables CR2SATRA to identify more *TN* leading to a higher *SPE*value. As shown in Fig. 7, CR2SATRA outperforms baseline models in *SPE* achieving overall wins across all datasets with a minimum value of $SPE = 0.9951$. The optimization of the final neuron state through HBWOA results in induced logic with a higher distribution of negative literals improving the ability of CR2SATRA to correctly classify *TN*.

The interaction between diversified neuron states and the predominance of negative outcomes in the data, enhances the logical representation in classified *TN*. Compared to baseline models, A2SATRA and S2SATRA are the closed competitor of CR2SATRA, which has differences in their attribute selection methods. Although both A2SATRA and S2SATRA use supervised attribute selection, CR2SATRA outperforms them in *SPE* results. This demonstrates the effectiveness of unsupervised *RTFSM* during the pre-processing which enhance the performance of CR2SATRA in generating high *SPE* values. Additionally, it emphasizes the importance of using *SPE* metrics for generating the best logic. As shown in Fig. 7, CR2SATRA outperforms all baseline models where the use of *SPE*metrics for generating best logic contributes to the optimal induce logic in retrieving *TN*.

### 7.4. MCC

Table 6 present the *MCC* values achieved by CR2SATRA in comparison with the baseline logic mining models across 20 datasets. Note that (*) indicates no ratio of improvement since no *MCC* values for the respective datasets. Bold values represent the highest values, while the bracket with positive ratios indicates that CR2SATRA outperforms the baseline models and the negative ratio show the opposite. While other performance metrics focus on specific component of confusion matrix such as *TP* and *TN*, *MCC* accounts for all four components, making it a more reliable measure of performance. Using *MCC* values to generate the best logic provides a more balanced view of *TP*, *TN*, *FP* and *FN* which is essential to avoid a random classifier. According to [36], an *MCC*value of 0.14 or lower indicates performance similar to or worse than a random classifier. Table 6 highlights the significant advancements of CR2SATRA which outperforms all baseline models with an

**Table 6**
The values of *MCC* for all logic mining models. The brackets indicate the improvement ratio and the negative ratio indicate that baseline model outperformed the proposed model.

| Dataset | CR2SATRA | 2SATRA | P2SATRA | E2SATRA | RA | 3SATRA | S2SATRA | A2SATRA |
|---|---|---|---|---|---|---|---|---|
| D1 | **0.6146** | 0.5233 (0.1745) | 0.5233 (0.1745) | 0.4608 (0.3338) | 0.042 (13.6333) | 0.3493 (0.7595) | 0.5528 (0.1118) | 0.5591 (0.0993) |
| D2 | 0.4753 | 0.0617 (6.7034) | 0.4952 (−0.0402) | 0.0292(15.2774) | 0.1643 (1.8929) | 0.1543 (2.0804) | **0.5021** (−0.0534) | 0.5021 (−0.0534) |
| D3 | **0.5804** | 0.0855 (5.7883) | 0.5020 (0.1562) | 0.2426 (1.3924) | 0.0953 (5.0902) | 0.4313 (0.3457) | 0.5083(0.1418) | 0.3283 (0.7679) |
| D4 | **0.4834** | 0.4406 (0.0971) | 0.4829 (0.0010) | 0.3706 (0.3044) | 0.0906 (4.3355) | 0.421 (0.1482) | 0.4686 (0.0316) | 0.4358 (0.1092) |
| D5 | **0.5542** | 0.3005 (0.8443) | 0.4990 (0.1106) | 0.4401 (0.2593) | 0.4006 (0.3834) | 0.5167 (0.0726) | 0.5357 (0.0345) | 0.4691 (0.1814) |
| D6 | 0.3612 | 0.0400 (8.0300) | 0.2424 (0.4901) | 0.0905 (2.9912) | 0.1748 (1.0664) | 0.3178 (0.1366) | **0.3876** (−0.0681) | 0.1448 (1.4945) |
| D7 | **0.5624** | 0.0043 (129.79) | 0.4891 (0.1499) | 0.2022 (1.7814) | 0.3384 (0.6619) | 0.1905 (1.9522) | 0.5588 (0.0064) | 0.3033 (0.8543) |
| D8 | **0.9903** | 0.8690 (0.1396) | 0.9058 (0.0933) | 0.8040 (0.2317) | 0.8595 (0.1522) | 0.5506 (0.7986) | 0.9264 (0.0690) | 0.9063 (0.0927) |
| D9 | **0.7167** | 0.2494 (1.8737) | 0.5249 (0.3654) | 0.2494 (1.8737) | 0.2112 (2.3935) | 0.4777 (0.5003) | 0.6213 (0.1535) | 0.5956 (0.2033) |
| D10 | **0.3008** | 0.0364 (7.2637) | 0.0376 (7.0000) | 0.0149(19.1879) | 0.0178 (15.8989) | 0.0749 (3.016) | 0.2102 (0.4310) | 0.1159 (1.5953) |
| D11 | **0.2473** | 0.1376 (0.7972) | 0.1182 (1.0922) | 0.0541 (3.5712) | 0.0554 (3.4639) | 0.1498 (0.6509) | 0.2054 (0.204) | 0.0864 (1.8623) |
| D12 | **0.4397** | 0.0657 (5.6925) | 0.3551 (0.2382) | 0.0011 (398.72) | 0.004 (108.925) | 0.1121 (2.9224) | 0.2952 (0.4895) | 0.2937 (0.4971) |
| D13 | **0.4060** | 0.0372 (9.9140) | 0.2597 (0.5633) | 0.1486 (1.7322) | 0.1285 (2.1595) | 0.0606 (5.6997) | 0.3331 (0.2189) | 0.2913 (0.3938) |
| D14 | **0.3243** | 0.014(21.5208) | 0.0123(25.3659) | 0.1735 (0.8692) | 0.0018 (179.16) | 0.1869 (0.7352) | 0.1888 (0.7177) | 0.3044 (0.0654) |
| D15 | 0.1779 | **0.2406** (−0.2606) | 0.1544 (0.1522) | 0.1928 (−0.0773) | 0.0199 (7.9397) | 0.0938 (0.8966) | 0.1221 (0.4570) | (*) |
| D16 | 0.6174 | 0.6020 (0.0256) | 0.6612 (−0.0662) | 0.6020 (0.0256) | 0.5443 (0.1343) | 0.1923 (2.2106) | 0.5052 (0.2221) | **0.6612** (−0.0662) |
| D17 | **0.5774** | 0.0305(17.9311) | 0.0532 (9.8534) | 0.1075 (4.3712) | 0.1560 (2.7013) | 0.0611 (8.4501) | 0.5684 (0.0158) | 0.0533 (9.833) |
| D18 | **0.2982** | 0.0261(10.4253) | 0.0358 (7.3296) | 0.0899 (2.3170) | 0.0530 (4.6264) | 0.082 (2.6366) | 0.0727 (3.1018) | 0.0609 (3.8966) |
| D19 | **0.9774** | 0.5405 (0.8083) | 0.8728 (0.1198) | 0.4639 (1.1069) | 0.0240 (39.7250) | 0.6099 (0.6026) | 0.9419 (0.0377) | 0.9397 (0.0401) |
| D20 | **0.7379** | 0.1483 (3.9757) | 0.0138 (52.471) | (*) | 0.1483 (3.9757) | 0.2459 (2.0008) | 0.4094 (0.8024) | 0.0123(58.9919) |
| (+/=/−) | | 19/0/1 | 19/0/1 | 19/0/1 | 20/0/0 | 20/0/0 | 18/0/2 | 18/0/2 |
| **Max** | **0.9903** | 0.8690 | 0.9058 | 0.8040 | 0.8595 | 0.6099 | 0.9419 | 0.9397 |
| **Min** | 0.1779 | 0.0043 | 0.0123 | **0.0011** | 0.0018 | 0.0606 | 0.0727 | 0.0123 |
| **Average** | **0.5221** | 0.2227 | 0.3619 | 0.2494 | 0.1765 | 0.2639 | 0.4457 | 0.3718 |

average *MCC* value of 0.5221 across all datasets. CR2SATRA achieved the highest *MCC* in 16 out of 20 datasets. Importantly, CR2SATRA stands out by not having any datasets with *MCC* values of 0.14 or below, unlike S2SATRA has two datasets, A2SATRA has five datasets, and both 2SATRA and RA have eleven datasets.

According to [14], *MCC* quantifies the balance between *TP* and *TN*. To achieve this, the induced logic must effectively represent both positive and negative outcomes by accurately capturing entries that lead to *TP* and *TN*. The proposed CR2SATRA model, with its combination of first-order and second-order clauses, benefits from the implementation of HBWOA, which enhances diversification by ensuring 50% negated states within clauses. Furthermore, the uses of the Smish activation function force the local field to generate induced logic with a higher distribution of negative literals. This enables CR2SATRA to effectively capture entries that lead to negative outcomes. With more negative literals, CRAN2SAT improves its ability to detect negative outcomes, resulting in higher *TN* values. This allows CR2SATRA to consistently generate induced logic that demonstrates high *TP* and *TN*. In contrast, the baseline logic mining models rely on systematic logical structures that limit the number of literals in each clause, reducing their flexibility in capturing overall trends of the dataset. While these structures have a high probability of achieving satisfied interpretations, leading to increased *TP*, they lack the ability to balance outcomes effectively. This is evident from the *PRE* results, where 3SATRA achieved the second-highest average *PRE* value indicating its strong tendency to produce positive outcomes. However, the higher-order systematic structure of 3SATRA prioritizes positive outcomes over preserving negative ones, resulting in difficulty maintaining high *TN* values. Consequently, 3SATRA struggles to achieve a balance between *TP* and *TN* leading to decreased *MCC* values across all repository datasets. Although CR2SATRA achieves the highest average *MCC* across datasets, certain datasets show higher *MCC* values for 2SATRA (D15), S2SATRA (D2 and D6), and A2SATRA (D16). These models employ a systematic second-order logical structure which may provide stronger consistency in representing attribute relationships for specific dataset. In such cases, the systematic second-order logical structure can lead to higher *MCC* values. Despite these variations, CR2SATRA achieves the highest average *MCC* and outperforms other models in 16 datasets demonstrating stable performance across different datasets.

RA recorded the lowest average *MCC* at 17.65%, followed by 2SATRA at 22.27% and E2SATRA at 24.94%. Table 6 highlights the performance differences between CR2SATRA, 2SATRA and RA. The lower performance of 2SATRA and RA is due to their reliance on true outcomes, leading to biased classification and overlooking minority classes with insufficient data for learning [22]. This bias contributes to the decline in *MCC* values. Furthermore, the attribute selection method in RA, 2SATRA, and E2SATRA is random and these models do not utilize a permutation operator which limits their ability to balance the distributions of *TP*, *TN*, *FP* and *FN*. Consequently, these models behave like random classifiers. In contrast, CR2SATRA employs a permutation operator that enhances connectivity among optimally selected attributes using unsupervised *RTFSM*.This ensures that only the most relevant attributes are included in the logical rule representation of the model. As shown in Table 6, CR2SATRA achieved the highest *MCC* values of 0.9903 for dataset D8 and 0.9774 for dataset D19, demonstrating its ability to produce optimal induced logic with balanced classification of *TP*, *TN*, *FP* and *FN*. These results demonstrate the effectiveness of unsupervised *RTFSM* during pre-processing in improving performance of CR2SATRA and highlight the importance of using *MCC* as a metric for generating the best logic.

### 7.5. MCR

The results of *MCR* for all logic mining models analyzing 20 datasets with 5-fold cross-validation are reported in Fig. 8. Based on the findings in Fig. 8, CRAN2SAT has reduced the incorrectly classified *FP* and *FN* compared to the baseline logic mining models. It is shown that CR2SATRA achieved the lowest *MCR* for the majority of the datasets with an average value 17.58%. The lowest value of *MCR* indicates that the model is capable of making accurate classifications with minimal misclassification. According to a study by [49], if there is no misclassified between the predicted and actual outcome of the data entries, then the *MCR* will be centered at zero. CR2SATRA outperforms baseline logic mining models by achieving a low *MCR* in 16 out of 20 datasets. This demonstrates the effectiveness of CR2SATRA in minimizing misclassifications of outcomes as *FP* and *FN*.

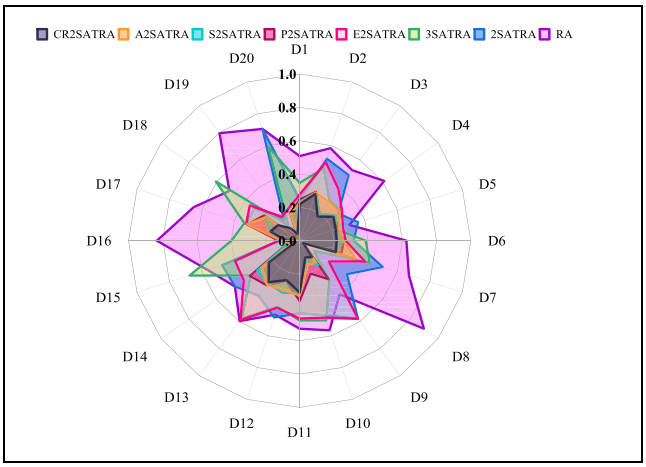


**Fig. 8.** The values of *MCR* for all logic mining models.

Although *MCR* and *ACC* are generally complementary measures of classification performance, the generation of the best logic using different metrics does not consistently lead to a complementary result. The *ACC* values demonstrates the capability of the model to accurately classify outcomes as *TP* and *TN*. While the *MCR* demonstrates the effectiveness of the model in minimizing misclassifications of outcomes as *FP* and *FN*. When generating best logic based on *MCR*, the proposed CR2SATRA demonstrates the ability to produce induced logic with low *FP* and *FN*.This is evident from the comparison of CR2SATRA with baseline logic mining models as shown in Fig. 8. The results indicate that CR2SATRA outperforms E2SATRA, 3SATRA, 2SATRA and RA across all 20 datasets. Although E2SATRA, 3SATRA, 2SATRA and RA employ a systematic logic, these logic mining models generate induced logic with high value of *FP* and *FN* leading to high value of *MCR*.This is attributed to the absence of neuron connectivity across the attributes. These models do not employ a permutation operator and depend on a random attribute selection method, which limits the interpretability of the neuron connections in the trained logical rule within DHNN. According to [19], utilizing neuron permutation is crucial for obtaining accurate induced logic. Furthermore, incorrect attribute selection leads the DHNN to learn suboptimal synaptic weights and thus resulting in suboptimal induced logic [22].

Unlike the *JFSM* by [14], which considers only the distance of positive entries of the attributes to the outcome, the attribute selection method in CR2SATRA based on *RTFSM*. *RTFSM* takes into account the distances of both positive and negative attribute entries to the outcome. This allows *RTFSM* to provide a more comprehensive understanding of the relationship between the attributes and the outcomes. Consequently, the implementation of the logical permutation along with the use of unsupervised *RTFSM* for attribute selection enhances the capability of CR2SATRA to produce induced logic that can effectively minimizes *FP* and *FN*. Despite achieving a good result, CR2SATRA consistently loses for dataset D19 as compared to A2SATRA, S2SATRA and P2SATRA. As mentioned in the previous discussion, the dataset D19 contains a high ratio of negative outcomes. Therefore, the induced logic generated by CR2SATRA tends to misclassify these outcomes as *FP* and *FN*leading to a high *MCR*values. In comparison with the baseline logic mining models A2SATRA, S2SATRA and P2SATRA, one of the reasons for CR2SATRA generates high *FP* and *FN* is influenced by the non-systematic logic of second-order clauses and first-order clauses. Unlike the A2SATRA, S2SATRA and P2SATRA which is a systematic logic of second-order clauses.

The flexible structure of $P_{CRAN2SAT}$ results in higher *MCR* for datasets with a high ratio of negative outcomes making it difficult for CR2SATRA to minimize *FP* and *FN*. However, CR2SATRA generates induced logic with a high distribution of negative literals due to uses of the Smish activation function and the implementation of HBWOA. As a results, the induced logic of CR2SATRA enhances its ability to minimize misclassifications leading to more accurate classification particularly in challenging cases of distinguishing between positive and negative instances, [50] conducted research on cost sensitive model for IVF dataset to check the total cost associated with the misclassification of *FP* and *FN*. IVF decisions made by clinicians are prone to wrong decision if *FP* are not addressed. This highlights the importance of minimizing both *FP* and *FN* for more accurate predictions. A low value of *MCR* indicates a better performing classifier which can significantly improve accuracy.

The improvement ratio is computed as the relative difference between the performance of the proposed model and the baseline model. Positive values indicate that the proposed model outperforms the baseline, while negative values indicate performance degradation relative to the baseline.

### 7.6. *Pareto analysis* of multi-objective optimization

The retrieval phase has been enhanced by implementing multi-objective optimization using HBWOA which focuses on achieving diversified and global states while minimizing similarity indices. Multi-objective optimization is typically associated with Pareto optimality, where a Pareto front represents a set of non-dominated solutions in which improving one objective may reduce the performance of another. According to [51], the points on the Pareto front illustrate optimal trade-offs between competing objectives. In the proposed framework, the objectives are designed to be complementary, aiming to improve diversification, global optimality, and similarity minimization simultaneously during the retrieval process. Therefore, Pareto analysis is used to examine the improvement of these objectives across simulations. This is clearly illustrated in Fig. 9, where all objectives show consistent improvement across simulations. This behavior occurs because the objectives are designed to be complementary within the HBWOA retrieval framework. Consequently, the optimization process seeks solutions that improve diversification, global optimality, and similarity minimization simultaneously.

### 7.7. *Ablation study*

Furthermore, an ablation study was conducted on the proposed logic mining model, CR2SATRA to assess the capability of CR2SATRA in doing knowledge extraction with and without implementing HBWOA. This ablation study evaluates the contribution of HBWOA in achieving a multi-objective function which is expected to improve the logic mining model in doing classification task. The ablation study aims to investigate whether the implementation of HBWOA enables CR2SATRA to produce optimal induced logic with a $N_D$ and $N_G$. Table 7 illustrates the qualitative analysis of CR2SATRA with and without HBWOA. The findings indicate that utilizing HBWOA enhances the optimization of the final neuron state, significantly contributing to the achievement of both $N_D$ and $N_G$. As a result, this contributes to the production of optimal induced

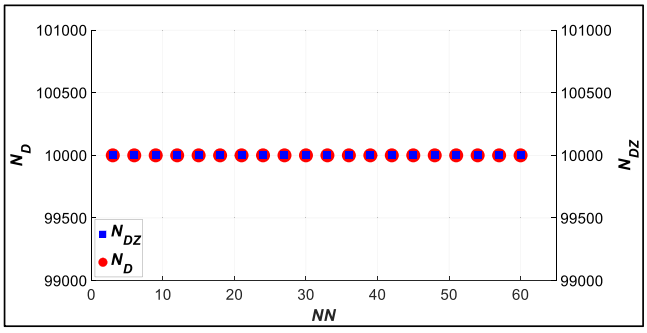


**Fig. 9.** Pareto superiority in multi-objective improvement.

**Table 7**
Ablation study on CR2SATRA.

| $P^{induced}_{CR2SATRA}$ | With HBWOA | Without HBWOA |
|---|---|---|
| Diversified states ($N_D$) | ✔ | x |
| Global states ($N_G$) | ✔ | x |

logic retrieved by CR2SATRA ($P^{induced}_{CR2SATRA}$). This aligns with recent findings by [12]. In their study, multi-objective optimization was implemented during the retrieval phase of DHNN to enhance the logic mining process particularly in producing optimal induced logic for classification tasks. Their approach aimed to generate diversified and globally optimal solutions with low similarity indices which consequently **improved the overall effectiveness of the model**. In this qualitative analysis, a tick mark (✔) indicates that the implementation of HBWOA enables CR2SATRA retrieved induced logic that achieve $N_D$ and $N_G$ leading to a more effective logic mining model. While a cross mark (x) indicates that without implementation of HBWOA, CR2SATRA is unable to retrieved induced logic that achieve both $N_D$ and $N_G$.However, without HBWOA, the induced logic that is retrieved by CR2SATRA fails to achieve both $N_D$ and $N_G$. Thus, this ablation study verifies that HBWOA significantly improves the capabilities of the proposed logic mining model.

### 7.8. Friedman test

Statistical tests were performed to assess the impact of different metrics in generating the best logic, in comparison to baseline logic mining models. The Friedman Test was performed with a 95% confidence interval (α=0.05) and a degree of freedom $df = 7$. Table 8 illustrates the null hypothesis versus the alternative hypothesis for all metrics used to generate the best logic. As mentioned by [52], the Friedman Test evaluates whether there are significant differences in the performance of different methods. A $p$-value$< \alpha$ results in rejecting the null hypothesis ($H_0$) otherwise, $H_0$ is not rejected. The results were analyzed using the Statistical Package for the Social Sciences (SPSS) and the average rank demonstrates the superiority of specific metrics in CR2SATRA over other baseline logic mining models. For metrics with a maximum value of 1, the highest average rank indicates the dominance of the logic mining model in terms of those specific performance metrics. Conversely, for metrics like misclassification rate where best value of 0, the lowest average rank indicates the dominance of the logic mining model in terms of this metrics.

Table 9 highlights the detailed findings of each respective Friedman test in terms of various performance metrics. The capabilities and effectiveness of the proposed logic mining model, which emphasizes different metrics in generating the best logic will be discussed based on the rank analysis of the Friedman Test for each metric. From Table 9, CR2SATRA displays the highest average rank of 7.76 for *ACC*. Additionally, the $p$-value for *ACC* value is $5.1676 \times 10^{-21}$ which is $<0.05$. As a result, the null hypothesis indicating no significant difference in performance among all logic mining model in obtaining *ACC* is rejected. This indicates that the logic mining models perform differently in retrieving induced logic with high *TP* and *TN*. Thus, this highlighting the superiority of CR2SATRA in terms of *ACC* metric

For the metric *PRE* with a $p$-value of $1.6237 \times 10^{-7}$, the $H_0$ is rejected. This indicates that there is a significant difference in performance among all logic mining models in obtaining *PRE*. The proposed logic mining model, CR2SATRA displays the highest average rank of 7.39 which highlight the dominance of CR2SATRA in terms *PRE*. As compared to the baseline logic mining models, CR2SATRA is superior in accurately classifying positive instances which acknowledge the effectiveness of the proposed *PRE* metrics in generating the best logic. This consequently leads to high *PRE* value. The rank analysis from the Friedman Test in Table 9 shows that CR2SATRA obtained the highest average rank for *SPE* which is 7.97. The $p$-value is $3.8881 \times 10^{-14}$ which is $<0.05$ leading to the rejection of $H_0$. The statistical significance of the *SPE* values in Table 9 for each logic mining model shows that they do not perform equally in maximizing *TN* values, which impacts the resulting *SPE*. In terms of *SPE*, CR2SATRA is superior to the baseline logic mining models.

Furthermore, the $p$-value for *MCC* is $9.7719 \times 10^{-14}$ which is $<0.05$. Therefore, the decision is to reject $H_0$. This indicates that the performance of all logic mining models in terms of *MCC* show a significant difference. With an average *MCC* rank of 7.67, CR2SATRA stands out for its ability to generate induced logic that maintains a balanced proportion of *TP* and *TN*, surpassing the baseline models. Based on the ranks analysis from Friedman Test in terms of *MCR* in Table 9, CR2SATRA obtained the lowest average rank of *MCR* which is 1.32. The $p$-value is $7.1418 \times 10^{-21}$ which is also $<0.05$ leading to the rejection of $H_0$. The *MCR* values in Table 9 are statistically significant for each logic mining model, indicating that the models perform differently in minimizing false classifications *FP* and *FN* which directly impacts the*MCR* values. As a result, CR2SATRA stands out for its ability to achieve lower *MCR* values, surpassing the performance of the baseline logic mining models.

**Table 8**
The null hypothesis vs alternative hypothesis for Friedman Test.

| | |
|---|---|
| Null hypothesis, ($H_0$) | There is no significant difference among all logic mining models for metrics *ACC*, *PRE*, *SPE*, *MCR* and *MCR*. |
| Alternative hypothesis, ($H_1$) | There is a significant difference among all logic mining models for metrics *ACC*, *PRE*, *SPE*, *MCR* and *MCR*. |

**Table 9**
The average ranks and $p$-value for Friedman Test.

| Logic mining model | Average Rank based on Performance metric | | | | |
|---|---|---|---|---|---|
| | *ACC* | *PRE* | *SPE* | *MCC* | *MCR* |
| CR2SATRA | **7.76** | **7.39** | **7.97** | **7.67** | **1.32** |
| A2SATRA | 6.00 | 3.47 | 4.63 | 5.17 | 3.00 |
| S2SATRA | 6.47 | 4.18 | 5.18 | 6.47 | 2.45 |
| P2SATRA | 5.53 | 4.37 | 4.95 | 4.89 | 3.47 |
| E2SATRA | 2.95 | 3.21 | 3.42 | 2.83 | 6.05 |
| 3SATRA | 3.05 | 5.68 | 4.79 | 3.83 | 5.95 |
| 2SATRA | 2.92 | 4.37 | 3.68 | 2.58 | 6.08 |
| RA | 1.32 | 3.32 | 1.37 | 2.56 | 7.68 |
| $p$-value | $5.1676 \times 10^{-21}$ | $1.6237 \times 10^{-7}$ | $3.8881 \times 10^{-14}$ | $9.7719 \times 10^{-14}$ | $7.1418 \times 10^{-21}$ |

### 7.9. Computational complexity analysis

The computational complexity of logic mining models primarily depends on the number of initialized solution strings, where the number of neurons denotes the number of initializations. The proposed CR2SATRA introduces key enhancements that influence both the learning and retrieval phases. During the learning phase, CR2SATRA selects $\rho$ best logics and trains each $\rho$ using DHNN which leads to increased computational complexity compared to baseline models that rely on a single logic. In the retrieval phase, CR2SATRA applies HBWOA to optimize the retrieved neuron state. This iterative optimization process further increases the complexity but significantly improves diversification resulting in a more effective logic mining model for classification tasks. In contrast, baseline logic mining models do not incorporate iterative optimization during the retrieval phase and rely on the single best logic in the learning phase. Consequently, their overall computational complexity is $O(N)$ which is significantly lower compared to CR2SATRA. The computational complexity of each logic mining model is analyzed with respect to the number of generated best logics ($\rho$), the number of neurons ($N$) and the number of trials ($NT$) which is shown in Table 10.

**Table 10**
The computational complexity of each logic mining model.

| Logic mining model | Computational complexity |
|---|---|
| CR2SATRA | $O(\rho + 3N!)$ |
| RA | $O(N)$ |
| 2SATRA | $O(N)$ |
| 3SATRA | $O(N)$ |
| E2SATRA | $O(N \times NT)$ |
| P2SATRA | $O(N!)$ |
| S2SATRA | $O(N!)$ |
| A2SATRA | $O(N!)$ |

## 8. Conclusion

This paper contributes to the field of logic mining by presenting a new approach which forms the foundation of CR2SATRA, including the pre-processing, learning, and retrieval phases of DHNN. CR2SATRA generates a set of induced logics that offer flexibility to interpret the behaviour of the dataset based on induced logic from different confusion matrix. CR2SATRA addresses the challenge of imbalanced datasets by expanding the search space for optimal best logic that generates based on different pathways. In this context, CR2SATRA outperforms baseline logic mining models in retrieving both negative and positive classifications. CR2SATRA demonstrating higher average $ACC = 0.8275$, $PRE = 0.9116$, $PRE = 0.9116$, $SPE = 0.9994$, $MCC = 0.5221$and $MCR = 0.1758$. The flexibility of CR2SATRA in interpreting the behavior of the dataset using set of induced logics from different confusion matrix is a key advantage over baseline logic mining models, which rely on a single best logic. Furthermore, the use of *RTFSM* for attribute selection is particularly beneficial for imbalanced datasets, as it considers the distribution of positive and negative entries between the attributes with the outcomes. Despite the promising results, CR2SATRA has certain limitations. As discussed in Section 7.9, the computational complexity of the proposed model is higher than that of the baseline models. The paper also suggests future research directions to further improve logic mining models. Instead of relying solely on a single metric in generating the best logic [14,24], the logic mining model can be improved by introducing alternative approach in generating the best logic by having a combination of confusion matrix such as $ACC + PRE$, $PRE + SPE$ or $ACC + MCR$. Additionally, a new method for selecting the induced logic could involve the process of minimizing the difference between the *ACC* of the best logic with the *ACC* of the induced logic. In the context of attribute selection methods, the integration of supervised and unsupervised approaches can be explored to enhance the quality of selected attributes and improve overall model performance. These approaches will demonstrate consistently high performance across both learning and testing datasets and thus lead to the development of model that can efficiently handle different computational challenges. Additionally, future work will explore the applicability of the CR2SATRA model for real-time or large-scale analysis of online customer reviews, particularly in customer behavior analysis using the Booking.com dataset.

## CRediT authorship contribution statement

**Nurshazneem Roslan:** Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Saratha Sathasivam:** Writing – review & editing, Validation, Conceptualization. **Nur Ezlin Zamri:** Writing – review & editing, Methodology, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The author is deeply thankful to the editor and the reviewers for their valuable suggestions to improve the quality of this manuscript.

## Data Availability

All data used in the manuscript was retrieved from reputable data repository sites.

## References

[1] I. Świetlicka, W. Kuniszyk-Jóźkowiak, M. Świetlicki, Artificial neural networks combined with the principal component analysis for non-fluent speech recognition, Sensors 22 (2022) 321, https://doi.org/10.3390/s22010321.
[2] B. Pandey, D.Kumar Pandey, B. Pratap Mishra, W. Rhmann, A comprehensive survey of deep learning in the field of medical imaging and medical natural language processing: challenges and research directions, J. King Saud Univ. Comp. Inform. Sci. 34 (2022) 5083–5099, https://doi.org/10.1016/j.jksuci.2021.01.007.
[3] M. Kim, J. Yun, Y. Cho, K. Shin, R. Jang, H.J. Bae, N. Kim, Deep learning in medical imaging, Neurospine 16 (2019) 657–668, https://doi.org/10.14245/ns.1938396.198.
[4] D. Wu, X. Ma, D.L. Olson, Financial distress prediction using integrated Z-score and multilayer perceptron neural networks, Decis. Support. Syst. 159 (2022) 113814, https://doi.org/10.1016/j.dss.2022.113814.
[5] J.J. Hopfield, D.W. Tank, Neural" computation of decisions in optimization problems, Biol. Cybern. 52 (1985) 141–152, https://doi.org/10.1007/BF00339943.
[6] W.A.T.W. Abdullah, Logic programming on a neural network, Int. J. Intell. Syst. 7 (1992) 513–519, https://doi.org/10.1002/int.4550070604.
[7] M.S.M. Kasihmuddin, M.A. Mansor, S. Sathasivam, Hybrid genetic algorithm in the Hopfield network for logic satisfiability problem, Pertanika J. Sci. Technol. 25 (2017) 139–152.
[8] M.A. Mansor, M.S.M. Kasihmuddin, S. Sathasivam, Artificial immune system paradigm in the Hopfield network for 3-satisfiability problem, Pertanika J. Sci. Technol. 25 (2017) 1173–1188.
[9] S. Sathasivam, M.A. Mansor, A.I.M. Ismail, S.Z.M. Jamaludin, M.S. M. Kasihmuddin, M. Mamat, Novel random k satisfiability for $k \leq 2$ in Hopfield Neural Network, Sains Malays 49 (2020) 2847–2857, https://doi.org/10.17576/jsm-2020-4911-23.
[10] Y. Guo, M.S.M. Kasihmuddin, Y. Gao, M.A. Mansor, H.A. Wahab, N.E. Zamri, J. Chen, YRAN2SAT: a novel flexible random satisfiability logical rule in discrete Hopfield neural network, Adv. Eng. Softw. 171 (2022) 103169, https://doi.org/10.1016/j.advengsoft.2022.103169.
[11] N.E. Zamri, S.A. Azhar, M.A. Mansor, A. Alway, M.S.M. Kasihmuddin, Weighted random k satisfiability for k=1,2 (r2SAT) in discrete Hopfield Neural network, Appl. Soft. Comput. 126 (2022) 109312, https://doi.org/10.1016/j.asoc.2022.109312.
[12] N.A. Romli, N.F.S. Zulkepli, M.S.M. Kasihmuddin, N.E. Zamri, N. 'Afifah Rusdi, G. Manoharam, Mohd.A. Mansor, S.Z.M. Jamaludin, A.A. Malik, Unsupervised logic mining with a binary clonal selection algorithm in multi-unit discrete Hopfield neural networks via weighted systematic 2 satisfiability, AIMS Math. 9 (2024) 22321–22365, https://doi.org/10.3934/math.20241087.
[13] N. 'Afifah Rusdi, N.E. Zamri, M.S.M. Kasihmuddin, N.A. Romli, G. Manoharam, S. Abdeen, M.A. Mansor, Synergizing intelligence and knowledge discovery: hybrid black hole algorithm for optimizing discrete Hopfield neural network with negative based systematic satisfiability, AIMS Math. 9 (2024) 29820–29882, https://doi.org/10.3934/math.20241444.
[14] N.E. Zamri, Mohd.A. Mansor, M.S.M. Kasihmuddin, S.S. Sidik, A. Alway, N. A. Romli, Y. Guo, S.Z.M. Jamaludin, A modified reverse-based analysis logic mining model with weighted random 2 satisfiability logic in discrete Hopfield neural network and multi-objective training of modified niched genetic algorithm, Expert. Syst. Appl. 240 (2024) 122307, https://doi.org/10.1016/j.eswa.2023.122307.
[15] S. Sathasivam, W.A.T.W. Abdullah, Logic mining in neural network: reverse analysis method, Computing 91 (2011) 119–133, https://doi.org/10.1007/s00607-010-0117-9.
[16] L.C. Kho, M.S.M. Kasihmuddin, M.A. Mansor, S. Sathasivam, Logic mining in league of legends, Pertanika J. Sci. Technol. 28 (2020) 211–225.
[17] N.E. Zamri, M.A. Mansor, M.S. Mohd Kasihmuddin, A. Alway, S.Z.M. Jamaludin, S. A. Alzaeemi, Amazon employees resources access data extraction via clonal selection algorithm and logic mining approach, Entropy 22 (2020) 596, https://doi.org/10.3390/E22060596.
[18] S.Z.M. Jamaludin, M.S.M. Kasihmuddin, A.I.M. Ismail, M.A. Mansor, M.F.M. Basir, Energy based logic mining analysis with Hopfield Neural Network for recruitment evaluation, Entropy 23 (2021) 40, https://doi.org/10.3390/e23010040.
[19] S.Z.M. Jamaludin, M.A. Mansor, A. Baharum, M.S.M. Kasihmuddin, H.A. Wahab, M.F. Marsani, Modified 2 satisfiability reverse analysis method via logical permutation operator, Comp. Mater. Contin. 74 (2023) 2853–2870, https://doi.org/10.32604/cmc.2023.032654.

[20] M.S.M. Kasihmuddin, S.Z.M. Jamaludin, M.A. Mansor, H.A. Wahab, S.M.S. Ghadzi, Supervised learning perspective in logic mining, Mathematics 10 (2022) 915, https://doi.org/10.3390/math10060915.
[21] S.Z.M. Jamaludin, N.A. Romli, M.S.M. Kasihmuddin, A. Baharum, M.A. Mansor, M. F. Marsani, Novel logic mining incorporating log linear approach, J. King Saud Univ. Comp. Inform. Sci. 34 (2022) 9011–9027, https://doi.org/10.1016/j.jksuci.2022.08.026.
[22] N.A. Rusdi, M.S.M. Kasihmuddin, N.A. Romli, G. Manoharam, M.A. Mansor, Multi-unit Discrete Hopfield Neural network for higher order supervised learning through logic mining: optimal performance design and attribute selection, J. King Saud Univ. Comp. Inform. Sci. 35 (2023) 101554, https://doi.org/10.1016/j.jksuci.2023.101554.
[23] G. Manoharam, M.S.M. Kasihmuddin, S.N.F.M.A. Antony, N.A. Romli, N.A. Rusdi, S. Abdeen, M.A. Mansor, Log-linear-based logic mining with multi-discrete Hopfield neural network, Mathematics 11 (2023) 2121, https://doi.org/10.3390/math11092121.
[24] A. Alway, N.E. Zamri, Mohd.A. Mansor, M.S.M. Kasihmuddin, S.Z.M. Jamaludin, M.F. Marsani, A novel Hybrid exhaustive search and data preparation technique with multi-objective discrete Hopfield Neural Network, Deci. Anal. J. 9 (2023) 100354, https://doi.org/10.1016/j.dajour.2023.100354.
[25] Y. Guo, M.S.M. Kasihmuddin, N.E. Zamri, J. Li, N.A. Romli, M.A. Mansor, W.N. A. Ruzai, Logic mining method via hybrid discrete Hopfield neural network, Comput. Ind. Eng. 206 (2025) 111200, https://doi.org/10.1016/j.cie.2025.111200.
[26] S. Abdeen, M.S.M. Kasihmuddin, N.E. Zamri, G. Manoharam, M.A. Mansor, N. Alshehri, S-type random k satisfiability logic in discrete Hopfield neural network using probability distribution: performance optimization and analysis, Mathematics 11 (2023) 984, https://doi.org/10.3390/math11040984.
[27] M.S.M. Kasihmuddin, M.A. Mansor, M.F.M. Basir, S. Sathasivam, Discrete mutation Hopfield neural network in propositional satisfiability, Mathematics 7 (2019) 1133, https://doi.org/10.3390/MATH7111133.
[28] N. Roslan, S. Sathasivam, F.L. Azizan, Conditional random k satisfiability modeling for k =1,2 (CRAN2SAT) with non-monotonic Smish activation function in discrete Hopfield neural network, AIMS Math. 9 (2024) 3911–3956, https://doi.org/10.3934/math.2024193.
[29] S. Sathasivam, M.A. Mansor, M.S.M. Kasihmuddin, H. Abubakar, Election algorithm for random k satisfiability in the Hopfield Neural network, processes 8 (2020) 568. https://doi.org/10.3390/PR8050568.
[30] F.L. Azizan, S. Sathasivam, N. Roslan, A.D. Ibrahim, Logic mining with hybridized 3-satisfiability fuzzy logic and harmony search algorithm in Hopfield neural network for Covid-19 death cases, AIMS Math. 9 (2024) 3150–3173, https://doi.org/10.3934/math.2024153.
[31] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67, https://doi.org/10.1016/j.advengsoft.2016.01.008.
[32] A.G. Hussien, A.E. Hassanien, E.H. Houssein, M. Amin, A.T. Azar, New binary whale optimization algorithm for discrete optimization problems, Eng. Optimiz. 52 (2020) 945–959, https://doi.org/10.1080/0305215X.2019.1624740.
[33] N. Roslan, S. Sathasivam, A multi-objective whale optimization for solving combinatorial optimization problems via Hopfield neural networks, Neur. Comput. Appl. 38 (2026) 158, https://doi.org/10.1007/s00521-025-11759-5.
[34] M.H. Nadimi-Shahraki, H. Zamani, S. Mirjalili, Enhanced whale optimization algorithm for medical feature selection: a COVID-19 case study, Comput. Biol. Med. 148 (2022) 105858, https://doi.org/10.1016/j.compbiomed.2022.105858.
[35] I.D. Mienye, Y. Sun, Performance analysis of cost-sensitive learning methods with application to imbalanced medical data, Inform. Med. Unlock. 25 (2021) 100690, https://doi.org/10.1016/j.imu.2021.100690.
[36] D. Chicco, G. Jurman, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, BMC Genom. 21 (2020) 1–13, https://doi.org/10.1186/s12864-019-6413-7.
[37] Q. Zhu, On the performance of Matthews correlation coefficient (MCC) for imbalanced dataset, Pattern. Recognit. Lett. 136 (2020) 71–80, https://doi.org/10.1016/j.patrec.2020.03.030.
[38] S. Sen, A.V. Deokar, Toward understanding variations in price and billing in US healthcare services: a predictive analytics approach, Expert. Syst. Appl. 209 (2022) 118241, https://doi.org/10.1016/j.eswa.2022.118241.
[39] T. Suresh, T.A. Assegie, S. Rajkumar, N.K. Kumar, A hybrid approach to medical decision-making: diagnosis of heart disease with machine-learning model, Int. J. Elect. Comp. Eng. 12 (2022) 1831, https://doi.org/10.11591/ijece.v12i2.pp1831-1838.
[40] S. Srivastava, P.P. Dash, ML-based reconfigurable symbol decoder: an alternative for next-generation communication systems, Eng. Appl. Artif. Intell. 114 (2022) 105123, https://doi.org/10.1016/j.engappai.2022.105123.
[41] C.S. Shieh, T.T. Nguyen, M.F. Horng, Detection of unknown DDoS attack using convolutional neural networks featuring geometrical metric, Mathematics 11 (2023) 2145, https://doi.org/10.3390/math11092145.
[42] F.J.M. Shamrat, R. Shakil, B. Akter, M.Z. Ahmed, K. Ahmed, F.M. Bui, M.A. Moni, F.M. Bui, An advanced deep neural network for fundus image analysis and enhancing diabetic retinopathy detection, Healthc. Anal. 5 (2024) 100303, https://doi.org/10.1016/j.health.2024.100303.
[43] X. Wang, H. Ren, A. Wang, Smish: a novel activation function for deep learning methods, Electronics (Switzerland) 11 (2022) 540, https://doi.org/10.3390/electronics11040540.
[44] S.R. Dubey, S.K. Singh, B.B. Chaudhuri, Activation functions in deep learning: a comprehensive survey and benchmark, Neurocomputing 503 (2022) 92–108, https://doi.org/10.1016/j.neucom.2022.06.111.
[45] C. Iwendi, K. Mahboob, Z. Khalid, A.R. Javed, M. Rizwan, U. Ghosh, Classification of COVID-19 individuals using adaptive neuro-fuzzy inference system, in: Multimed. Syst.., 2022: pp. 1223–1237. https://doi.org/10.1007/s00530-021-00774-w.
[46] I. Domingues, J.P. Amorim, P.H. Abreu, H. Duarte, J. Santos, Evaluation of oversampling data balancing techniques in the context of ordinal classification, in: 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8, https://doi.org/10.1109/IJCNN.2018.8489599.
[47] S.O. Folorunso, J.B. Awotunde, N.O. Adeboye, O.E. Matiluko, Data classification model for COVID-19 Pandemic. Advances in Data Science and Intelligent Data Communication Technologies for COVID-19: Innovative Solutions Against COVID-19, 2022, pp. 93–118, https://doi.org/10.1007/978-3-030-77302-1_6.
[48] M. Noureddin, E. Truong, J.A. Gornbein, R. Saouaf, M. Guindi, T. Todo, N. Noureddin, J.D. Yang, S.A. Harrison, N. Alkhouri, MRI-based (MAST) score accurately identifies patients with NASH and significant fibrosis, J. Hepatol. 76 (2022) 781–787, https://doi.org/10.1016/j.jhep.2021.11.012.
[49] Y. Dong, W.D. Pan, D. Wu, Impact of misclassification rates on compression efficiency of red blood cell images of malaria infection using deep learning, Entropy 21 (2019) 1062, https://doi.org/10.3390/e21111062.
[50] A. Kumaravel, T. Vijayan, Comparing cost sensitive classifiers by the false-positive to false- negative ratio in diagnostic studies, Expert. Syst. Appl. 227 (2023) 120303, https://doi.org/10.1016/j.eswa.2023.120303.
[51] J. Shao, Y. Lu, Y. Sun, L. Zhao, An improved multi-objective particle swarm optimization algorithm for the design of foundation pit of rail transit upper cover project, Sci. Rep. 15 (2025), https://doi.org/10.1038/s41598-025-87350-8.
[52] J. Carrasco, S. García, M.M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: practical guidelines and a critical review, Swarm. Evol. Comput. 54 (2020) 100665, https://doi.org/10.1016/j.swevo.2020.100665.