AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 10(12): 29342–29379.
DOI: 10.3934/math.20251289
Received: 03 August 2025
Revised: 24 September 2025
Accepted: 25 September 2025
Published: 12 December 2025

Research article

An explainable logic mining framework with multi-objective metaheuristic algorithm for knowledge extraction in discrete Hopfield neural network

Syed Anayet Karim[1], Mohd Shareduwan Mohd Kasihmuddin[2], Sowmitra Das[3], Nur Ezlin Zamri[4,*], Akib Jayed Islam[5], Alyaa Alway[6] and Deepak Kumar Chowdhury[5]

[1] Department of Natural Science, Faculty of Science & Engineering, Port City International University, Chattogram, 4225, Bangladesh

[2] School of Mathematical Sciences, Universiti Sains Malaysia, Penang, 11800, Malaysia

[3] Department of Computer Science & Engineering, Faculty of Science & Engineering, Port City International University, Chattogram, 4225, Bangladesh

[4] Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

[5] Department of Electrical & Electronic Engineering, Faculty of Science & Engineering, Port City International University, Chattogram, 4225, Bangladesh

[6] School of Distance Education, Universiti Sains Malaysia, Gelugor, Penang, 11800 USM, Malaysia

* **Correspondence:** Email: ezlinzamri@upm.edu.my; Tel: +60397696677.

Abstract: A specific field of data extraction termed "logic mining" is important for retrieving insightful information from intricate datasets by generating logical representations. These logical frameworks are explainable and significant for knowledge-driven technologies in computational optimization. However, existing logic mining models suffer from key limitations, including inadequate attribute selection, rigid logical rule structures, inefficient training processes, and storage constraints that often lead to overfitting. To address these challenges, this study proposed an explainable logic mining framework that integrated four key components: At first, a log-linear based attribute selection method to identify significant features; second, a non-systematic higher-order logic structure using random k satisfiability (for $k \leq 3$) to enhance flexibility; after that, a multi-objective hybrid election algorithm for efficient and adaptive training; and, finally, an expanded retrieval phase employing a permutation operator to optimize the synaptic weight space in the discrete Hopfield neural network. The proposed framework was validated through comparative analyses against eight baseline models using real-world multidisciplinary datasets. Performance was rigorously evaluated across four evaluation metrics, where the experimental results demonstrated that the proposed model achieved a maximum accuracy of 97.73%, a precision of 100%, a specificity of 99.17%, and a matthews correlation coefficient (MCC) of 0.95 across 20 real-world datasets. Moreover, the proposed model's efficiency was also statistically validated through Nemenyi's post-hoc test and Cohen's d effect sizes, confirming its superior classification capability, stability, and reliability in logic-based knowledge.

Keywords: random satisfiability; non-systematic logic; hybrid election algorithm; logic mining; discrete Hopfield neural network
Mathematics Subject Classification: 68N17, 68R07, 68T27

1. Introduction

In the ever-evolving landscape of computational design, data analysis and knowledge extraction, the integration of cutting-edge techniques from various fields has become increasingly crucial. Many researchers nowadays use data mining, which involves various mathematical and computational techniques to identify patterns, relationships, and insights from large datasets. The goal is to extract significant information in a structured format for further analysis across fields like telecommunication, business, and healthcare [1–3]. Prediction and classification using data mining techniques [4–6] have attracted significant research attention due to their effectiveness in organizing data into distinct groups or classes. Nowadays, various AI models, such as random forest [7], support vector machine [8], and artificial neural network (ANN), along with metaheuristics [9, 10], are proposed for forecasting, prediction, and classification. Despite their accuracy and reliability, many classification decisions remain opaque, making it difficult to understand their underlying rationale. To address this challenge, data representation can shift toward logical structures instead of relying solely on black-box models. Therefore, it is crucial to explore how these logical structures can effectively represent data within ANNs, given their widespread use and versatility in handling complex patterns.

Discrete Hopfield neural network (DHNN), a form of an ANN introduced by [11], is widely used in data mining for its ability to learn complex patterns and solve optimization problems such as the traveling salesman and linear programming. DHNN features consist of content associative memory (CAM), which stores problem patterns and consists of interconnected neurons without hidden layers, making it suitable for optimization problems and complex dynamic tasks [12, 13]. Due to its capabilities, DHNN has been widely utilized in various applications. DHNN is often criticized as a "black box" due to interpretability issues. To address this, integrating satisfiability (SAT) concepts into DHNN enhances the ability to understand the complexity of neuron management during training and testing phases. The article [14] first time composed SAT as a logical framework into DHNN for user-interpretable outputs. SAT can be systematic, limiting clauses to k variables, or non-systematic, allowing varying variables.

Another new concept of SAT studies is random k satisfiability (RANkSAT) in DHNN, using flexible clause structures for optimized neuron states. RANkSAT can include first, second, and third-order logic for broader rule satisfaction. This RANkSAT concept was utilized and explored by [15], where it proposed major 2 satisfiability with majority second-order logic for better DHNN interpretability. After that, the article [16] further explored major random first and third order logic, balancing negated literals and neuron representation. These findings show non-systematic SAT supports diverse neuron solutions. Later, the work [17] investigated third and second-order clauses, revealing improved performance and broader solution exploration in non-systematic logic. This work used the Wan Abdullah (WA) method, which successfully enumerated various magnitudes of synaptic weights, resulting in the generation of different final neuron states. This is evident from the DHNN's performance in achieving a high diversity of neuron variations. However, prior studies using single-objective DHNN models revealed significant shortcomings in minimizing the cost function, primarily due to the absence of training algorithms and limited exploration of the search space. These issues highlight the need for more robust training strategies. Notably, the training phase of the DHNN–SAT model can benefit substantially from optimized training algorithms [18].

Incorporating multi-objective functions alongside effective algorithms can address these limitations and expand the solution space, thereby enhancing dataset interpretation [19]. In this context, it needs to investigate how the multi-objective concept can be incorporated into the DHNN. Meanwhile, the work by [20] proposed a multi-objective function with a hybrid exhaustive search (HES) algorithm to increase the DHNN's storage capacity, while [21] introduced methods to improve local field searches. Despite these advancements, both studies faced limitations related to search space, algorithm effectiveness, and logical rule structure. Thus, refining logical representation and employing efficient training methods remain essential for real-life problem-solving and data analysis for knowledge extraction.

Identically, extraction of knowledge from the datasets through the 'Logic mining' framework will be a significant approach. In this mechanism, the logical relationship is considered as induced logic, which has the capacity to analyze classifications. By utilizing the generated induced logic from a dataset, the extracted knowledge not only focuses on the final outcomes of the dataset but also identifies significant features within the dataset and uncovers potential logical trends. Moreover, logic mining in the DHNN-SAT model involves constructing logical rules to describe associations between input features and output variables in classification tasks [12]. From the logic mining perspective, where knowledge extraction is the key focus, the feedback network DHNN is ideal since it has an associative memory concept along with an energy minimization process that ensures robust and noise-resistant pattern extraction. Moreover, the symbolic and explainable architecture of DHNN makes it preferable to other networks for structured knowledge extraction. Meanwhile, the most crucial part of logic mining is the data preprocessing phase, particularly attribute selection methods. These methods are categorized into unsupervised and supervised approaches. Unsupervised learning plays a key role in enhancing classification performance by randomly identifying the most relevant attributes. Studies have focused on unsupervised methods like the reverse analysis (RA) method [22], which integrates neural logic to extract rules describing the behavior of a dataset.

Building on the previous logical extraction concept, [23] proposed the 2 satisfiability reverse analysis (2SATRA) method to classify credit assessments across different financial sectors. This work captured systematic second order logical structures where it only limited pairwise constraints. These approaches face limitations due to random attribute selection, resulting in low accuracy values. Despite advancements, challenges persist in effectively arranging attributes within 2SAT rules, thereby limiting accuracy and interpretability. The systematic SAT concept, further updated by the work [24], subsequently introduced permutation based 2SATRA (P2SATRA), integrating a logical permutation operator to refine attribute arrangement and enhance the accuracy of DHNN retrieval phases. The findings underscored that P2SATRA surpassed all current logic mining methods across diverse performance metrics. Lamentably, none of these logic mining models had incorporated a feature selection method prior to embedding data entries as neurons in DHNN, resulting in the training of insignificant attributes and thereby increasing redundancy and decreasing relevance in classification.

Therefore, one of the creative works by [25] introduced a supervised approach (S2SATRA) to address interpretability issues in logic mining. Instead of random attribute selection, correlation analysis is conducted in the preprocessing phase to choose attributes representing the 2SAT logical rule. S2SATRA demonstrated superior classification performance compared to other logic mining methods. This supervised approach selects significant attributes, reducing the need for permutation operators to obtain optimal induced logic. In another study, [26] proposed a log-linear analysis to address issues with randomized attribute selection in 2SAT logic representation (A2SATRA). Attributes with significance levels less than the threshold value are selected as optimal attributes.

From the above literature, the existing logic mining models, though capable of producing optimal induced logic, still face two major limitations. First, their reliance on systematic logic structures often leads to overfitting, reducing generalization across diverse datasets. Second, the training mechanism of DHNN remains a critical bottleneck—most models do not adequately address how to efficiently optimize the training phase to ensure stability and scalability. Moreover, the concept of higher-order non-systematic logic, which could identify a more suitable subset of induced logic to better represent dataset behavior, has not been sufficiently explored. However, the optimal attribute selection method with higher-order logical rules and multi-objective training algorithms in DHNN models remains an unexplored area of research.

To bridge these gaps, this article introduces a novel logic mining framework that departs from purely systematic structures by incorporating higher-order non-systematic logic, enabling more accurate and flexible representation of dataset behavior. Here, the proposed approach further enhances the training phase of DHNN through a multi-objective dynamic hybrid election algorithm (HEA), ensuring efficient convergence and reducing the risk of overfitting. Additionally, the framework integrates an optimal feature selection strategy and a constructive computation of best logic, along with an effective retrieval mechanism to diversify induced logic outcomes. This combination represents a unique advancement in logic mining, offering a more robust, adaptive, and generalizable solution compared to prior approaches. Hence, the contributions of this paper are listed as follows:

(1) To formulate and implement a multi-objective function for higher-order non-systematic logic with a metaheuristic named hybrid election algorithm to optimize the training phase of the discrete Hopfield neural network. This efficient network trains the logic concerning fitness and diversity, which corresponds to increasing the storage capacity of the network. Meanwhile, this network utilizes five (05) best logical structures so that optimal induced logic can be produced during the retrieval phase.

(2) To propose a dynamic and explainable logic mining model that is highly capable of predictions and classifications of real-life datasets. The proposed model inputs the ‘log-linear’ method in the preprocessing phase and embeds the ‘permutation operator’ in the retrieval phase of the network. Notably, the filtering mechanism of the log-linear method ensures optimal attributes in the logical structure. At the same time, the permutation operator expands the search space to find the optimal logic that represents the behavior of the datasets.

(3) To ensure a comprehensive and practically meaningful evaluation, the proposed logic-mining model is validated on diverse real-world datasets and assessed using multiple established

performance metrics. Its results are benchmarked against state-of-the-art logic-mining approaches to highlight improvements. Furthermore, statistical significance is verified through the Nemenyi post-hoc test, and effect sizes are also reported to rigorously confirm the robustness and reliability of the proposed model.

This paper is structured as follows: Section 2 reviews the theoretical underpinnings of multi-objective functional RAN*k*SAT logic within discrete Hopfield neural networks and introduces the hybrid election algorithm. Section 3 details the proposed logic-mining model, while Section 4 describes the experimental design. In Section 5, the empirical results are presented and critically analyzed, and Section 6 offers concluding remarks and outlines directions for future research.

2. Background study

This section presents the theoretical framework underlying all components of the proposed logic mining. The initial subsection addresses the general formulation of higher-order non-systematic logic, specifically RAN*k*SAT (where $k \leq 3$). Following this, the second subsection elucidates the fundamental principles of the neuro-symbolic model of RAN*k*SAT logic, referred to as DHNN-RAN*k*SAT.

2.1. Random k satisfiability representation

One of the notable advancements in the study of SAT is RAN*k*SAT, which is highly favored by ANN researchers due to its independent clause arrangements [17]. This structure consists of random literals, each being a combination of positive and negative literals within a single part. RAN*k*SAT represents a type of non-systematic logical formulation in which every logical clause comprises exactly k variables connected by the OR($\vee$) operator, with each variable negated with a probability of $\frac{1}{2}$. The key features of RAN*k*SAT can be summarized as follows:

a) The RAN*k*SAT model is constructed using a set of x variables where $A_1, A_2, A_3, ...A_x$ with a set of y clauses denoted as $Z_1^{(k)}, Z_2^{(k)}, Z_3^{(k)}, ...Z_y^{(k)}$.

b) A set of definite clauses $C_1, C_2, C_3, ...C_y$ relate to logical AND($\wedge$) operator, and the clauses themselves are linked by the logical OR($\vee$) operator.

c) Each clause contains ‘x’ different literals selected at random, with the probability ratio of positive to negative literals being l:m where l,m $\in [1, 2]$ for (for $k \leq 3$).

d) The logical formulation must consist of non-redundant variables, meaning no single variable is repeated within the entire logical structure.

Each variable can only be assigned a bipolar value of 1 or -1, representing whether the variable is TRUE or FALSE, respectively. The general form of RAN*k*SAT is defined by Eqs (2.1) to (2.3), where $P^{1,3}_{RANkSAT}$ comprises first and third-order logic, $P^{2,3}_{RANkSAT}$ consists of second and third-order logic, and $P^{1,2,3}_{RANkSAT}$ includes first, second, and third-order logic, respectively. The RAN*k*SAT logical rules articulated in Eqs (2.1) to (2.4) are integrated into DHNN.

$$P^{1,3}_{RANkSAT} = \wedge^{u}_{i=1} Z^{(1)}_i \wedge^{w}_{i=1} Z^{(3)}_i, \tag{2.1}$$

$$P^{2,3}_{RANkSAT} = \wedge^{v}_{i=1} Z^{(2)}_i \wedge^{w}_{i=1} Z^{(3)}_i, \tag{2.2}$$

$$P^{1,2,3}_{RANkSAT} = \wedge^{u}_{i=1} Z^{(1)}_i \wedge^{v}_{i=1} Z^{(2)}_i \wedge^{w}_{i=1} Z^{(3)}_i, \tag{2.3}$$

$$where, Z^{(k)}_i = \begin{cases} A_i, & k = 1; \\ (B_i \vee C_i), & k = 2; \\ (D_i \vee E_i \vee F_i), & k = 3, \end{cases} \tag{2.4}$$

where by u, v and w is the total number of first, second, and third-order logic in each clause in $P^k_{RAN3SAT}$, respectively. Note that $u > 0, v > 0, w > 0$, and it consists of clauses with various orders of $P^k_{RAN3SAT}$. From Eq (2.4), the literals show whether it is positive or negative. Conjointly, the selection $Z^{(k)}_i$ is set at random as it investigates the behavior of $P^k_{RAN3SAT}$. Following Eq (2.4), the literals either positive or negative is set at random where $A_i \in \{A_i, \neg A_i\}, B_i \in \{B_i, \neg B_i\}, C_i \in \{C_i, \neg C_i\}$, and $D_i \in \{D_i, \neg D_i\}, E_i \in \{E_i, \neg E_i\}\, F \in \{F_i, \neg F_i\}$. Meanwhile, DHNN outputs are represented as either +1 or -1. Each neuron's inputs are multiplied by synaptic weights, which are stored in the CAM for determining the final neuron states. This study employs the asynchronous updating rule to prevent neuron oscillation. Theorem 2.1 demonstrates the asynchronous operation of DHNN and the updating rule for DHNN with the i-th activation.

Theorem 2.1. *Starting from an initial state in the search space, all networks represented by Eq (2.5) in randomized asynchronous mode will fall into a network gap with a probability of one [27].*

$$S_i = \begin{cases} 1, \ if \ \sum\limits_{j}^{n} W^{(3)}_{abc} S_b S_c \geq U_p; \\ -1, \ otherwise. \end{cases} \tag{2.5}$$

From Eq (2.5), $W^{(3)}_{abc}$ is the synaptic weight from unit a to c . S_b is the current state of the unit b, and U_p is the predefined threshold. Several studies defined $U_p = 0$ to verify that the DHNN always lead to a decrease in energy monotonically [17].

2.2. RANkSAT in discrete Hopfield neural network

This section illustrates the general overview of RAN*k*SAT logic as represented within DHNN, referred to as DHNN-RAN*k*SAT. The embedding of RAN*k*SAT in DHNN is divided into two phases: the learning or training phase, and the retrieval or testing phase. In this context, logic satisfiability notations are derived from propositional logic formulas integrated with DHNN-RAN*k*SAT. The primary objective of the DHNN-RAN*k*SAT training phase is to determine the appropriate synaptic weights for the RAN*k*SAT clauses. According to the [14] technique, any logic governed by a set of instructions must achieve a consistent interpretation that produces $P^k_{RAN3SAT}$ true. Meanwhile, computing the cost function $C_{P^k_{RAN3SAT}}$ in DHNN is crucial for reducing the logical inconsistencies in $P^k_{RAN3SAT}$ $\left(C_{P^k_{RAN3SAT}} = 0\right)$. The formulation $C_{P^k_{RAN3SAT}}$ of Eqs (2.6) and (2.7), which accommodate all forms of logic combinations $P^k_{RAN3SAT}$, is as follows:

$$C_{P^k_{RAN3SAT}} = \frac{1}{8}\sum_{i=1}^{w}\left(\prod_{j=1}^{3} L_{ij}\right) + \frac{1}{4}\sum_{i=1}^{v}\left(\prod_{j=1}^{2} L_{ij}\right) + \frac{1}{2}\sum_{i=1}^{u}\left(\prod_{j=1}^{1} L_{ij}\right), \tag{2.6}$$

$$L_{ij} = \begin{cases} \frac{1}{2}(1 - S_i), \; if \neg i; \\ \frac{1}{2}(1 + S_i), \; otherwise. \end{cases} \tag{2.7}$$

Here, S_i represents the neuron state where $i \in \{1, -1\}$. DHNN-RANkSAT effectively minimizes the cost function $C_{P^k_{RAN3SAT}}$, facilitating the determination of appropriate synaptic weights and generating a favorable energy profile. Given that $P^k_{RAN3SAT}$ has a zero-cost function, it ensures a satisfied interpretation in which all clauses yield a truth value. The local field of the proposed model is formulated in Eqs (2.8) and (2.9) as follows:

$$h_{P^k_{RAN3SAT}}(t) = \sum_{c=1,c\neq b}^{n} \sum_{b=1,b\neq c}^{n} W^{(3)}_{abc} S_b S_c + \sum_{b=1,b\neq a}^{n} W^{(2)}_{ab} S_b + W^{(1)}_a, \tag{2.8}$$

$$where, S_i(t) = \begin{cases} 1, & \sum\limits_{c=1,c\neq b}^{n} \sum\limits_{c=1,c\neq b}^{n} W^{(3)}_{abc} S_b S_c + \sum\limits_{b=1,b\neq a}^{n} W^{(2)}_{ab} S_b + W^{(1)}_a \geq 0; \\ -1, & \sum\limits_{c=1,c\neq b}^{n} \sum\limits_{c=1,c\neq b}^{n} W^{(3)}_{abc} S_c S_b + \sum\limits_{b=1,b\neq a}^{n} W^{(2)}_{ab} S_b + W^{(1)}_a < 0\,. \end{cases} \tag{2.9}$$

Equation (2.8) provides the overall formulation of the local field for $P^k_{RAN3SAT}$, while Eq (2.9) presents a piecewise function representing the neuron's final state according to Theorem 2.1. During the testing phase, DHNN employs the hyperbolic tangent activation function (HTAF) to ensure the convergence of final neuron states and to prevent neuron oscillation. Additionally, the [14] method is used to compare Eq (2.6) with Eq (2.10), which is referred to as the energy function $H_{P^k_{RAN3SAT}}$.

$$\begin{aligned} lH_{P^k_{RAN3SAT}} = &-\frac{1}{3} \sum_{a=1,a\neq b\neq c}^{n} \sum_{b=1,a\neq b\neq c}^{n} \sum_{c=1,a\neq b\neq c}^{n} W^{(3)}_{abc} S_a S_b S_c \\ &-\frac{1}{2} \sum_{a=1,a\neq}^{n} \sum_{b=1,a\neq b}^{n} W^{(2)}_{ab} S_a S_b - \sum_{a=1}^{n} W^{(1)}_a S_a. \end{aligned} \tag{2.10}$$

Subsequently, the value $H_{P^k_{RAN3SAT}}$ reaches the absolute final energy, and the minimum energy $H^{min}_{P^k_{RAN3SAT}}$ is derived from $P^k_{RAN3SAT}$, which decreases monotonically. Thus, $H^{min}_{P^k_{RAN3SAT}}$ is computed using Eq (2.11),

$$H^{min}_{P^k_{RAN3SAT}} = -\frac{a\left(\psi^3_i\right) + 2\left(b\left(\psi^2_i\right)\right) + 4\left(c\left(\psi^1_i\right)\right)}{8}, \tag{2.11}$$

where, ψ^1_i, ψ^2_i, $\psi^3_i \in J^{(k)}_i$, and c, b, a represent the number of 1-literal, 2-literal, and 3-literal clauses in $P^k_{RAN3SAT}$. Equation (2.12) can ultimately assess the quality of the final neuron state by differentiating between global and local minimum solutions. Specifically, if Eq (2.12) is satisfied, the final neuron states achieve a global minimum solution; otherwise, it is trapped in a local minimum solution.

$$\left| H_{P^k_{RAN3SAT}} - H^{min}_{P^k_{RAN3SAT}} \right| \leq \tau, \tag{2.12}$$

where, $\tau = 0.001$ is a predefined value known as the tolerance value [21].

Furthermore, RANkSAT, as an innovative structure in SAT research, introduces a significant new dimension to the study of DHNN. By integrating multiple objectives, RANkSAT can enhance the motivation for exploring multi-objective concepts. This article explores the multi-objective nature of

RAN*k*SAT, demonstrating its ability to not only maximize fitness values but also to enhance diversity in logical rules while increasing the storage capacity of DHNN which greatly impacts on logic mining perspective. The multi-objective framework effectively integrates all objectives within the RAN*k*SAT paradigm, highlighting its potential and versatility in advancing DHNN research.

2.3. Multi-objective functions for RANkSAT in DHNN

Since RAN*k*SAT is one of the creative structures in SAT study, it opens another significant dimension in the exploration of DHNN. Multiple combination forms of RAN*k*SAT can include the probability of more than one objective. This higher-order RAN*k*SAT can increase the motivation on exercising multi-objective concepts. Several researchers [28–30] explored the multi-objective nature of neural network study and found significant development and scopes for future researchers. In this article, the multi-objective concept integrates three major components, which ensures optimal fitness and diversity that represents an innovative exploration of higher-order RAN*k*SAT representations. Meanwhile, the three objectives are: (i) maximum fitness ($F_{\max}$), (ii) diversity ratio (γ), and (iii) k ideal solution strings ($S^{(i)}_{\max}$). Overall, these multi-objective functions are formally expressed mathematically in Eqs (2.13) and (2.14):

$$f(F_{\max,}, \gamma, S^{(i)}_{\max}), \tag{2.13}$$

subject to

$$\left.\begin{aligned} F_{\max} &= \sum_{i=0}^{p} C_i^{(3)} + \sum_{i=0}^{q} C_i^{(2)} + \sum_{i=0}^{r} C_i^{(1)}, \\ \gamma &= \sum_{i=0}^{p} C_i^{(3)} + \sum_{i=0}^{q} C_i^{(2)}, \\ S^{(i)}_{\max} &= \left[S^{(1)}_{\max}, ...S^{(n)}_{\max}\right], \end{aligned}\right\} \tag{2.14}$$

where, $C_i^{(3)}, C_i^{(2)}, C_i^{(1)}$ are the third-order, second-order, and first-order clauses of RAN*k*SAT, respectively.

2.4. HEA for RANkSAT in DHNN

This research introduced a modified socio-inspired algorithm named HEA for the higher order SAT study in DHNN and disclosed the key prospects of this algorithm. The HEA in DHNN-RAN*k*SAT involves five key steps, from initialization to the final election day. The following steps and flow diagram of HEA are presented below:

(1) *Initialization:* A population of individuals (voters and candidates) is randomly generated. Each individual's state is noted as 1 (TRUE) or -1 (FALSE), aligning with the search space.

(2) *Eligibility assessment:* Random instances are subjected to a fitness function assessment to reward instances that correctly satisfy a RAN*k*SAT clause. The fitness value is utilized to ascertain eligibility, which can be determined using Eqs (2.15) and (2.16),

$$f_{L_j} = \sum_{i=1}^{NC} C_i^{(k)}, \; k = 1, 2, 3, \tag{2.15}$$

$$where, C_i^{(k)} = \begin{cases} 0, \; False; \\ 1, \; True. \end{cases} \tag{2.16}$$

(3) *Forming initial parties:* The solution space is divided into several parties. Each party's candidate is the individual with the highest fitness value, and the rest become voters. The correlation distance between candidates and voters is calculated.

(4) *Advertisement campaign:* In this hybrid algorithm, the key part lies in the advertisement campaign phase which is divided into four domains. The domains are discussed in brief below:

   (a) *Positive advertisement:* In this stage, the voters focus their internal possible candidates among themselves. After that the candidates reveal plans to sway voters, influencing their states and potentially replacing candidates with higher fitness voters.

   (b) *Negative advertisement:* After completing the positive advertisement stage, the technical stage is introduced which is named the negative advertisement. In this stage, the candidates attract voters from other parties by presenting their mission with vision, and expanding the search space.

   (c) *Coalition:* In this stage, more than one parties combine to explore additional search areas, selecting new candidates if voters have higher fitness values. After this traditional stage, the next stage, the Caretaker Party, is introduced to ensure diversity along with fitness.

   (d) *Caretaker Party:* The voters who have the highest fitness values are stored in an 'Elite pool' to maintain diversity and improve the model. The selection process for choosing the highest fitness can be calculated by Eq (2.17),

$$C_p^{best} = \mu f_{L_j}. \tag{2.17}$$

   Here, $\mu = [0.1, 0.4]$ is the ratio of the achieved maximum fitness value. Meanwhile, the mutation insertion concept is also utilized in this caretaker party to further improve diversity, thereby introducing a novel aspect to the proposed HEA model. It is noteworthy that the caretaker party emphasizes exploitation. Therefore, the choice of mutation insertion needs to prioritize methods that allow for random selection with a more localized search approach. In this context, the 'Shift Mutation' concept is utilized for shifting a randomly chosen frontier between two adjacent clauses by one/single step, either to the right or to the left, or vice versa [31]. Notably, shift mutation not only focuses on non-satisfied clauses but also focuses on the inclusion of positive-negative state combinations in each clause. Hence, the condition of Eq (2.17) is fulfilled, then the fittest voters are moved to the next stage for participating in the final round- Election Day.

(5) *The election day:* If the termination criteria for the 'Advertisement Campaign' exist, then the election is conducted to evaluate the final eligibility of all the candidates. The best solution for each party is tested. If it achieves maximum fitness with diversity, then it is declared the optimal solution. Otherwise, iterations continue until all conditions are met.

3. Proposed hybrid election algorithm for random *k* satisfiability based reverse analysis (R-HEARA) method

This article introduces a novel and explainable hybrid method in RA called the hybrid election algorithm for RAN*k*SAT-based reverse analysis (R-HEARA) in order to overcome the shortcomings of previously published research. Notably, this R-HEARA model is composed of several significant layers, which can be exposed as (i) the preprocessing phase, (ii) the training phase, and (iii) the testing phase. The proposed R-HEARA cartels flexible non-systematic logical combinations with a preprocessing phase and an ideal attribute selection approach to represent and map datasets flexibly and straightforwardly. To expand the search space of the dataset, the permutation operator has also been incorporated into the model. Henceforth, the characteristics in the dataset is translated into bipolar form, and then the attributes are chosen using log-linear methods in order to identify the best attribute for developing optimal logic. To get an optimized training mechanism a novel metaheuristics HEA is employed with a multi-objective concept which makes a new dimension in the study of logic mining. The link between the attributes of a data collection and the multi-objective HEA is represented by the optimal logic retrieved from R-HEARA. The general formation of the proposed R-HEARA is shown in Eqs (3.1) and (3.2):

$$P_{R-HEARA}^{k} = \wedge_{i=0}^{n} C_i^{(3)} \wedge_{i=0}^{m} C_i^{(2)} \wedge_{i=0}^{l} C_i^{(1)}, \tag{3.1}$$

where

$$C_i^{(k)} = \begin{cases} (E_i \vee F_i \vee G_i), \text{where} \quad k = 3; \\ (A_i \vee B_i), \text{where} \quad k = 2; \\ D_i, \text{where} \quad k = 1. \end{cases} \tag{3.2}$$

An example of different combinations is given as follows:

$$P_{R-HEARA}^{1,2,3} = (A_1 \vee \neg A_2 \vee A_3) \wedge (A_4 \vee A_5) \wedge (A_6 \vee A_7) \wedge \neg A_8 \wedge \neg A_9 \wedge A_{10}, \tag{3.3}$$

$$P_{R-HEARA}^{2,3} = (A_1 \vee \neg A_2 \vee A_3) \wedge (A_4 \vee A_5 \vee A_6) \wedge (A_7 \vee \neg A_8) \wedge (\neg A_9 \vee A_{10}), \tag{3.4}$$

$$P_{R-HEARA}^{1,3} = (A_1 \vee \neg A_2 \vee A_3) \wedge (A_4 \vee A_5 \vee A_6) \wedge A_7 \wedge \neg A_8 \wedge \neg A_9 \wedge A_{10}. \tag{3.5}$$

As referred from Eqs (3.3)–(3.5), $P_{R-HEARA}^{k}$ is satisfiable when $(A_1, A_{2,} A_3, A_{4,} A_5, A_{6,} A_7, A_{8,} A_9, A_{10})$ = (1,-1,1,1,1,1,1,-1,-1,1) or (-1,1,-1,-1,-1,-1,-1,1,1,-1) meaning, that the outcome result of the dataset shows $P_{R-HEARA}^{k} = 1$ which is noted as true positive (*TP*), and $P_{R-HEARA}^{k} = -1$ which is addressed as true negative (*TN*), respectively. It needs to be mentioned that all the $C_i^{(k)}$ must be satisfied to obtain the outcome of the dataset whether it is $P_{R-HEARA}^{k} = I; I = 1 \text{or} - 1$. Notably, the proposed $P_{R-HEARA}^{k}$ also strictly maintained non-redundant variables, which is another feature of the RAN*k*SAT structure. Moreover, the article [24] pointed out another critical issue in developing the possible solution space by introducing a permutation operator in the logical rule. Changing the arrangement of the variables through permutation can create more possible directions in real-life problems. Hence, in this article, the permutation operator is implied where the generalized rule of the $P_{R-HEARA}^{k}$ is expressed through Eq (3.6):

$$\begin{aligned} &P_{R-HEARA}^{k} = \wedge_{i=1}^{n} C_i^{(k)}, \\ &\text{where,} C_i^{(k)} = \begin{cases} \vee_{j=1}^{m} (A_v^a, B_v^b), k = 2; \\ \vee_{j=1,k=1}^{m} (A_v^a, B_v^b, C_v^c), k = 3. \end{cases} \end{aligned} \tag{3.6}$$

Here, a, b, c are the arrangement of the attributes and $a \neq b \neq c$. Then, A_v^a, B_v^b, C_v^c are the entries corresponding with the attributes a, b, c, respectively. Meanwhile, the k-mean clustering method was also introduced since this method is significant in ensuring that DHNN analyzes the appropriate values of the neurons [25]. Mathematically, the k-mean analysis for the R-HEARA model can be written as below in Eq (3.7),

$$J_{P_{R-HEARA}} = \sum_{J=1}^{n}\sum_{i=1}^{m}(x_i^{(j)} - C^2_{J_{P_{R-HEARA}}}), \tag{3.7}$$

where, $J_{P_{R-HEARA}}$ is an objective function, and is the number of observations and $C_{J_{P_{R-HEARA}}}$ is the centroid for the cluster j in R-HEARA. To set up k-mean clustering analysis in R-HEARA, it needs to select an arbitrary number of clusters for the given dataset and then compute the distance between each data point. Then update the number of clusters and their average. Repeat the process, until there is no change in mean. In order to improve the quality of induced logic, R-HEARA implements another statistical method named- 'Log-linear' to squash non-significant attributes from the datasets. According to [26], this log-linear method can investigate the associations of the attributes independently and also not consider the outcome of the dataset. The association of the attributes is considered acceptable if the p-value is less than 0.05. To measure the K-way and higher order effects of the attributes and calculate the partial association for the significant attributes corresponding with the outcome with Chi-Square, χ^2, and it is reported through Eq (3.8).

$$\begin{aligned}\chi^2 &= \sum_{S_pS_qS_r} fy_{S_pS_qS_r}[\ln(fy_{S_pS_qS_r} - a_{S_pS_qS_r})^2 - \ln(a_{S_pS_qS_r})]\\ &= \sum_{S_pS_qS_r} fy_{S_pS_qS_r}\ln\frac{(fy_{S_pS_qS_r} - a_{S_pS_qS_r})^2}{(a_{S_pS_qS_r})},\end{aligned} \tag{3.8}$$

where, $fy_{S_pS_qS_r}$ represents the observed frequencies, $a_{S_pS_qS_r}$ denotes the expected frequency, and Eq (3.7) is applied to determine the adequacy of sample size for the proposed model based on statistical conclusions. The partial association test assesses the significance of a parameter by comparing χ^2 values with and without that parameter in the model, considering the relevant degrees of freedom (df). In other words, it can be examined using the null hypothesis of no correlation between two or more variables in the RANkSAT strategy. Rejecting the null hypothesis indicates a significant correlation between two or more variables. Thus, the choice of positive/negative neurons are based on the frequency of the specific $C_i^{(k)}$ that appears in the dataset. Consequently, the RANkSAT structures of $C_i^{(k)}$ that leads to $P^{train}_{R-HEA} = I; I = 1\text{or} - 1$ which obtains the best logic that symbolizes as $P^{best}_{R-HEARA}$. To calculate the $P^{best}_{R-HEARA}$ concerning $C_i^{(k)}$ can be expressed in Eq (3.9):

$$P^{best}_{R-HEARA} = \max[n(C_i^{(k)})], P^{best}_{R-HEARA} = I; I = 1\text{or} - 1, \tag{3.9}$$

where, $n(C_i^{(k)})$ is the number of combination of third-second or third-first or third-second-first order clauses that leads to $P^{train}_{R-HEARA} = I$. The obtained $P^{best}_{R-HEARA}$ are trained in DHNN-RANkSATHEA. Then, in the testing phase, the final neuron state S_{D_l} is converted into an induced literal based on the following form of Eq (3.10):

$$S^{induced}_{D_l} = \begin{cases} S_{D_l}, S_{D_l} = 1;\\ \neg S_{D_l}, S_{D_l} = -1.\end{cases} \tag{3.10}$$

To enhance interpretability, the detailed procedural framework of the R-HEARA model is outlined as below:

The process begins with selecting a dataset, which is then partitioned into training and testing subsets using a standard k-fold cross-validation approach. In the dataset, each variable is assigned a corresponding logical representation. Prior to the clustering, a preprocessing phase is conducted wherein the log-linear concept is applied to identify optimal attributes, thereby enabling the generation of efficient induced logic. A confusion matrix (Table 1) is constructed to quantify the performance metrics, including TP, TN, false positives (FP), and false negatives (FN) [32]. Meanwhile, a set of optimal logic rules, denoted as $P^{best}_{R-HEARA}$, is generated and selected. These rules aim to maximize TP and TN, and are considered ideal representations of the original dataset. The associated cost function, $C_{R-HEARA}$, is computed using Eq (2.6).

Table 1. Binary classification error matrix [32].

Predicted class	True classification	
	Class 1	Class 2
Class 1	True Positive(*TP*)	False Positive (*FP*)
Class 2	False Negative (*FN*)	True Negative(*TN*)
Total Number of Objects in the class	*TP*+*FN*= Total number of Objects of Class 1	*FP*+*TN*= Total number of Objects of Class 2

Subsequently, clause satisfaction is evaluated through the hybrid election algorithm. The synaptic weights are computed using the Wan Abdullah method, followed by randomization of neuron states. The optimal logic rules and corresponding synaptic weights are stored in the Hopfield CAM, with each $P^{best}_{R-HEARA}$ capable of generating multiple CAMs. Local fields are calculated via Eq (2.8), and the HTAF is applied to stabilize neuron states. Final energy is computed using Eqrom neuron states. These induced logics are then evaluated using Eq (3.6). Finally, training and testing performances are assessed using standard metrics such as Accuracy, Precision, Specificity, and Matthews Correlation Coefficient.

In the Figure 1, the implementation process of the proposed R-HEARA logic mining model is shown, where the three different phases are shown in three different blocks; also, the pseudocode of the proposed R-HEARA model is given through Algorithm 1.

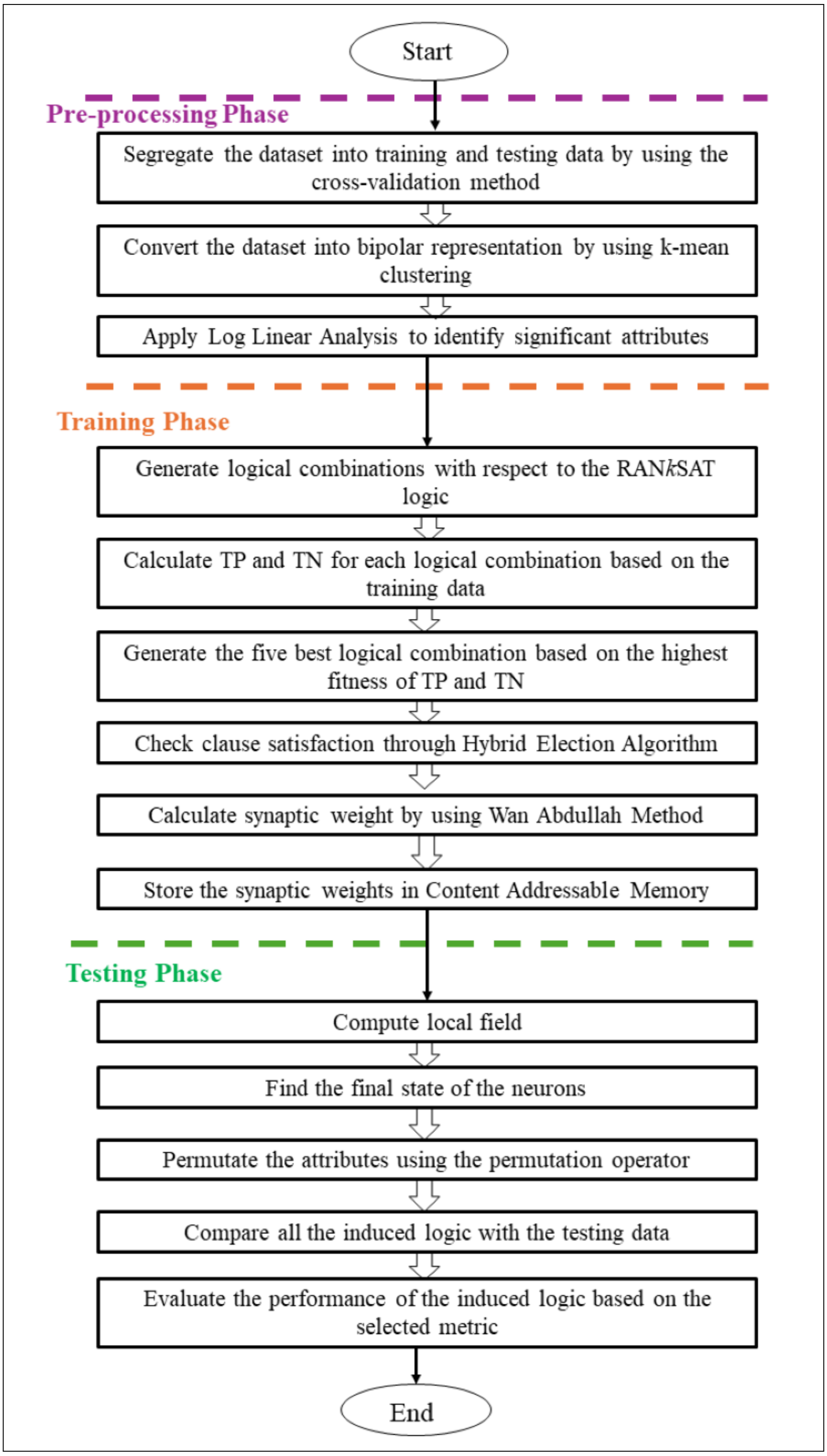
Start
Pre-processing Phase
Segregate the dataset into training and testing data by using the cross-validation method
Convert the dataset into bipolar representation by using k-mean clustering
Apply Log Linear Analysis to identify significant attributes
Training Phase
Generate logical combinations with respect to the RAN*k*SAT logic
Calculate TP and TN for each logical combination based on the training data
Generate the five best logical combination based on the highest fitness of TP and TN
Check clause satisfaction through Hybrid Election Algorithm
Calculate synaptic weight by using Wan Abdullah Method
Store the synaptic weights in Content Addressable Memory
Testing Phase
Compute local field
Find the final state of the neurons
Permutate the attributes using the permutation operator
Compare all the induced logic with the testing data
Evaluate the performance of the induced logic based on the selected metric
End

Figure 1. The implementation process of the proposed R-HEARA logic mining model.

Algorithm 1: Proposed R-HEARA model.

```
Input: Set all attributes $x_1, x_2, x_3, \ldots, x_n$ with respect to $P_{\text{train}}$, $R_{\text{HEARA}}$, $PS$, $trial$
Output: The best induced logic $P^{best}_{R\text{-HEARA}}$
Begin
Initialize all the parameters;
Initialize all the parameters;
Define the attributes for $x_1, x_2, x_3, \ldots, x_n$ with respect to $P^{k}_{R\text{-HEARA}}$;
Find the $p$-value for each attribute;
for All attributes where $p \leq 0.05$ do
    if Equation (3.4) is satisfied then
        Assign $x_i$ as $S_{Dl}$;
        while $i \leq PS$ do
            Find $P^{best}_{R\text{-HEARA}}$ using the attributes;
            Check the clause satisfaction for $P^{best}_{R\text{-HEARA}}$;
            Compute $H^{Pk}_{RAN3SAT}$ using Eq (2.10);
            Compute synaptic weight using WA method;
            Generate five $P^{best}_{R\text{-HEARA}}$ and store in CAM;
            Initialize the final neuron state;
            for $k \leq trial$ do
                Convert $S_{Dl}$ to logical form using Eq (3.10);
                Combine $S_{Dl}$ to generate induced logic $P^{induced}_{R\text{-HEARA}}$;
                Compare $P^{induced}_{R\text{-HEARA}}$ with $P^{test}_{R\text{-HEARA}}$;
End
```

4. Experimental design

This section provides a detailed discussion of the experimental workflow. Initially, the dataset undergoes analysis using International Business Machines Corporation Statistical Package for the Social Sciences (IBM SPSS) version 27, where it is structured in bipolar form. For the case of missing values, the missing values are replaced with the random bipolar state (1 or -1). However, the mentioned procedure can be applied with a minimal rate of missing values. Based on Dou et al. [33], if the dataset has more than 10% missing value rate, data imputation needs to be used to handle the missing values. As shown in Table 3, all 20 datasets used in this experiment have a lower rate of missing values. Subsequently, the program execution takes place in Microsoft Visual Studio with C Sharp due to its user-friendly interface and open-source nature. For efficient processing, the setup includes a device equipped with Windows 11, a Core i7 processor, and 16GB of RAM. Table 2 outlines the relevant parameters and corresponding values used in both proposed and existing models for experimentation and simulation.

Table 2. List of parameters of R-HEARA, including all baseline models that are used for experimentation and simulation.

Parameter	R-HEARA	S2SATRA	L2SATRA	P2SATRA	E2SATRA	3SATRA	A2SATRA	2SATRA	RA
		[25]	[42]	[24]	[43]	[44]	[26]	[23]	[22]
Learning Iter.	100	100	100	100	100	100	100	100	100
Learning Method	HEA	ES	ES	ES	ES	ES	ES	ES	ES
No. of Attributes	10	6	6	6	6	6	9	6	6
Attr. selection	LL*	LL*	LL*	LL*	RN†	RN†	LL*	RN†	RN†
Clause Comb.	$C_i^{(3,2,1)}$	$C_i^{(2)}$	$C_i^{(2)}$	$C_i^{(2)}$	$C_i^{(2)}$	$C_i^{(3)}$	$C_i^{(2)}$	$C_i^{(2)}$	$C_i^{(2,1)}$
No. of Trials	100	100	100	100	100	100	100	100	100
Permutation Size	10	–	–	100	100	–	–	–	–
Neuron Comb.	100	100	100	100	100	100	100	100	100
Learning Diversity	40%	–	–	–	–	–	–	–	–
Max. Comb.	100	100	100	100	100	100	100	100	100
Selection rate	0.1	0.1	0.1	0.1	–	0.1	0.1	0.1	–
Activation	HTAF	HTAF	HTAF	HTAF	HTAF	HTAF	HTAF	HTAF	–
CPU Time	24H	24H	24H	24H	24H	24H	24H	24H	24H
P-value	0.05	0.05	0.05	0.05	–	0.05	0.05	–	–
Energy Tol.	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	–

All the simulations employ the train-test splitting, which follows that 60% of the dataset is in the P-train, and the remaining 40% is in the P-test. In the present work, the authors applied distinct operations such as train-test splitting, clustering, or cross-validation following the practice of prior ANN-based satisfiability studies [24–26]. Here, the k-fold cross-validation concept is addressed to make the classification setup. In our implementation, log-linear attribute selection and k-means analysis are performed inside each fold of the k-fold cross-validation, not globally across the entire dataset. In the k-fold cross-validation, the dataset is partitioned into k-sized folds, where each partition is split into training and testing sets. Moreover, the works by [34] and [35] successfully utilized 5-fold cross-validation for their experiments and showed that it can safeguard against any leakage of label information throughout the preprocessing phase. This ensured that no label information from the test fold was used during feature selection or discretization. Additionally, partial association analysis is conducted on the attributes, involving the calculation of p-values compared against a threshold obtained from the Chi-square table to determine whether to reject the null hypothesis or not.

Metrics for performance evaluation serve as a baseline in the process of defining whether a model is effective or not. The confusion matrix serves as a tool to assess the discriminatory performance of the optimal solution during the training phase of binary classification tasks [36]. This evaluation can be based on the results of the classification training. This proposed study examines the distinctions between the proposed R-HEARA model and eight other existing baseline models. The comparison utilizes various performance metrics such as Accuracy (*Acc*) [37], Precision (*Prec*) [38], Matthews Correlation Coefficient (*MCC*) [39], and Specificity (*Spec*) [40,41]. These metrics are widely accepted for evaluating the optimal network configuration.

4.1. Baseline models

To validate the efficacy of the proposed model, R-HEARA is compared to several existing models. Although there are numerous classification methodologies in the present years, none of the proposed classification algorithms generate induced logical rules that can categorize and extract the dataset pattern properly. Here, the key point of baseline methods is discussed below.

a) RA: The RA method was inspired by [14], which extracted the Horn satisfiability logic from datasets. This work was the first attempt to induce logical rules from empirical data. The number of induced logics generated by the datasets is the primary calibration from the preceding RA. Induced logical rules are entrenched in datasets by using information about the connection strength. This study also revealed that using the RA method can analyze large datasets on a reasonable time scale and identify the hidden patterns of the datasets.

b) 2SATRA: The traditional logic mining model 2SATRA [23] employed P_{2SAT} as a logical rule in DHNN. The main aim of this study is to find a consistent interpretation that makes the P_{2SAT} satisfied. In this method, the attribute was positioned randomly into second-order clauses, which formulate P_{2SAT}^{best}, and the final neuron states were converted into final induced logic. The experimental result showed that this 2SATRA model has decent potential to obtain optimal logic from a trained dataset. Meanwhile, this work had no mechanism for selecting optimal attributes and improving the search space of the model.

c) P2SATRA: P2SATRA [24] introduced the permutation operator, where the attributes were arranged in a second-order clause. In this work, it is shown that logical permutation is a finite arrangement of the attributes that become true. The permutation operator explored all the probable search spaces that were only close to the selected attributes. The effect of logical permutation in P2SATRA integrates successfully in DHNN and showed that it performed well. Though in this work, the authors input different permutations that improve the result in terms of accuracy, there is no effort in introducing more performance indicators. However, this work neither employs any technique to improve the attribute selection nor improves the training mechanism.

d) E2SATRA: E2SATRA proposed by [43] is an energy-based logic mining technique that uses the global minimum energy to determine induced logic. During the testing phase of E2SATRA, the final neuron state that achieves global minimum energy is selected as the best induced logic. E2SATRA may therefore guarantee that the final neuron state is always satisfiable. This work made an alternative approach to extracting the relationships between the attributes that correspond to only positive outcomes of the dataset. Though in this work, two types of representation were introduced to show the efficiency of this model, there are no guidelines or involvement for non-systematic SAT representation, and also no concept to improve the attribute selections.

e) L2SATRA: L2SATRA was inspired by the work of [42], where the log-linear concept is used to do medical dataset analysis. This work introduced another new concept to improve the attribute selection process since earlier studies have drawbacks in selecting optimal attributes. In this work, the log-linear technique is employed to select the attribute and embed it into the traditional reverse analysis concept. This method is preferred when all variables of a dataset are shown in a qualitative measure and are discrete in nature. In this context, L2SATRA is dependent on the log-linear model to select the best attribute for getting P_{2SAT}^{best} . However, in this study, there is no discussion regarding the challenges of systematic and non-systematic SAT presentation and effective training mechanisms, which can play a vital role in analyzing real-life datasets.

f) S2SATRA: S2SATRA, one of the creative attempts by [25], which made a new direction in the study of logic mining. This work integrated supervised learning through association analysis to identify the optimal arrangement with respect to the logical rule. The concept S2SATRA attempted to discover the attribute formation and make a relation between the attribute and the training mechanism. In the testing phase, the S2SATRA was able to generate the best induced logic. Meanwhile, S2SATRA calculates the correlation to decide the actual attribute that is effective only for P_{S2SAT} . Moreover, this work did not approach any point for higher-order systematic and non-systematic SAT representation in the logic mining field, nor did it improve the training mechanism of DHNN.

g) 3SATRA: The work by [44] introduced the first logic mining model that utilized P_{3SAT} as a logical rule in DHNN, which is known as 3SATRA. In this method, the attribute is positioned randomly into third-order clauses, which formulate P_{3SAT}^{best}. In this work, a metaheuristic was introduced to enhance the capability of the training phase and focused on various performance metrics to overcome the previous work's drawbacks. The findings of this work revealed that effective training mechanisms improve the quality of the final neuron states, which leads to qualitative $P_{3SAT}^{induced}$. Though this study gave a breakthrough in the SAT study incorporating metaheuristics in the training phase of DHNN, there was no mechanism for building optimal attribute selection, as well as widening the search space, which made it highly possible to include sub-optimal attributes and generate poor induced logic.

h) A2SATRA: Another significant work named A2SATRA by [26] introduced a new direction of the study of logic mining. This work suggested adding another layer before the training phase, which is named the preprocessing phase. In this phase, the log-linear method is adopted to select optimal attributes as well as permutation operators to overcome the previous work's drawbacks. In this method, the attribute was positioned randomly into second-order clauses, which formulate P_{2SAT}^{best}, and the final neuron states were converted into final induced logic. The experiment result showed that this A2SATRA model has decent potential to obtain optimal logic from a trained dataset. Though this work can create attired induced logic, it only focuses on second-order clauses and also focuses on systematic logic. Moreover, in the training phase, this method applied a conventional ES algorithm, for which an optimal training phase could not be obtained throughout the process.

4.2. Benchmark datasets

In this study, 20 different real-life datasets cover multiple categories such as business, social, financial, medical, education, IT, health sector, etc., that are closely relevant to real life. These datasets are collected from the University of California, Irvine (UCI) machine learning repository (`https://archive.ics.uci.edu/ml/datasets.php`) and the Kaggle dataset (`https://www.kaggle.com/datasets?fileType=csv`) project website. This multivariate data from different fields is considered to analyze the performance of the R-HEARA model. The key task for the proposed

R-HEARA is to extract the possible best logical rule from each of the datasets that represents the dataset's behavior perfectly. In this dataset exploration, the datasets that have more than 13 attributes are considered for analysis. The proposed logical strategy is such that only 10 attributes are selected to describe the behavior of each dataset. The detail of all the 20 datasets is shown in the Table 3.

Table 3. List of benchmark datasets.

SL	dataset	Attr.	Inst.	Missing value	Rate of the missing value	Field	Outcome
D1	Absenteeism at work	21	740	No	0.00000	Business	Absenteeism time (hours)
D2	Adult	15	32561	Yes	0.00935	Social	Income
D3	Airline	21	1004	No	0.00000	Business	Satisfaction
D4	Audiology	24	200	No	0.00000	Life	Audibility
D5	Australian	15	690	No	0.00000	Financial	Class
D6	Bike sharing	17	17378	No	0.00000	Social	Shared
D7	Body signalling	27	55692	No	0.00000	Health	Smoking
D8	Bank marketing	21	4119	No	0.00000	Business	Employed number
D9	Campus recruitment	14	215	No	0.00000	Education	Placement status
D10	Cervical cancer behaviour	20	72	No	0.00000	Health	Cancer prediction
D11	Chemical composition dataset	18	89	No	0.00000	Physical	Part result
D12	Credit approval	15	690	Yes	0.00015	Financial	Approval
D13	Dermatology	34	366	Yes	0.01745	Life	Age
D14	Divorce predictor	54	171	No	0.00000	Life	Class
D15	Dry bean	17	13611	No	0.00000	Business	Class
D16	Facebook data	19	500	Yes	0.01107	IT	Total interactions
D17	Garment worker	16	1197	Yes	0.00063	Business	Productivity
D18	Heart	14	270	No	0.00000	Life	Disease prediction
D19	Obesity	17	2111	No	0.00000	Medical	Obesity level
D20	Lower back pain	13	310	No	0.00000	Medical	Class

In the context of statistical analysis, Table 4 presents the K-way and higher-order effects of model attributes for K=1. The focus on K=1 aims to identify interactions among individual variables. DHNN facilitates the observation of attribute interactions. While Table 4 indicates a significant impact of first-order effects, it does not imply significance for all attributes. Therefore, a detailed analysis of attributes through partial association is necessary. Prior to implementation in the proposed model, attributes undergo analysis based on partial association results. This involves calculating p-values and comparing them with a threshold value from the Chi-square table to determine whether to reject the null hypothesis or not. Rejection occurs when the p-value is below the threshold. In this study, parameter selection relies on p-values, specifically set at $p - value \leq 0.05$.

In neural-based models, computational complexity refers to the minimal neural resources required to complete a given task within an acceptable level of accuracy. Generally, computational complexity is defined as the analysis of how resource consumption scales with the size of the input dataset, particularly running time [45]. The primary focus of this proposed work is to find the ability of the HEA to optimize the multi-objective fitness function in DHNN. Hence, the trade-offs between computational complexity and optimal solutions are not considered in this experiment. For clarity, the computational complexity of each logic mining model is shown in the Table 5. Notably, the computational complexity of the HEA is $O(N^3)$ and traditional ES is $O(2^N)$.

Table 4. 13 K-way and higher order effects components of the model (K=1).

dataset	df	Pearson χ^2	p-Value
D1	2186	20133.681	≤ 0.05
D2	2186	1528391	≤ 0.05
D3	59048	519497.600	≤ 0.05
D4	2186	62566.900	≤ 0.05
D5	2186	3131284.574	≤ 0.05
D6	6560	1996538	≤ 0.05
D7	2186	463418	≤ 0.05
D8	6560	274541.100	≤ 0.05
D9	6560	22031.371	≤ 0.05
D10	2186	9304.137	≤ 0.05
D11	728	3573.568	≤ 0.05
D12	6560	83519.009	≤ 0.05
D13	6560	62240.130	≤ 0.05
D14	6560	382683.600	≤ 0.05
D15	6560	13997802	≤ 0.05
D16	6560	26029.880	≤ 0.05
D17	2186	114586.100	≤ 0.05
D18	6560	23495.400	≤ 0.05
D19	6560	42738.630	≤ 0.05
D20	6560	41095	≤ 0.05

Table 5. The computational complexity of each logic mining model, where g denotes the neuron combinations, N is the number of neurons, and NT is the number of trials.

Logic mining models	Computational complexity
R-HEARA	$O(g + N!)$
A2SATRA	$O(N!)$
S2SATRA	$O(N!)$
P2SATRA	$O(N!)$
L2SATRA	$O(N)$
E2SATRA	$O(N + NT)$
2SATRA	$O(N)$
3SATRA	$O(N)$
RA	$O(N)$

5. Results and discussions

The major objective of this experiment is to evaluate the performance of various logic mining models. In this section, the performance of the proposed R-HEARA model is compared against eight existing models. It is important to note that the bold entries in the tables for each metric indicate the highest performance values. Moreover, in each table, the minimum (Min), Maximum (Max), Standard deviation (STD), and very well-known rank analysis metric- Friedman test result are also depicted. Additionally, (+/=/-) asserts that the proposed logic mining model is better/equal/worst performing compared to the existing models.

5.1. Accuracy (Acc) analysis

From the *Acc* point of view, the proposed R-HEARA model achieved the highest *Acc* value in 15 out of 20 datasets. This demonstrates that R-HEARA produces a large number of positive outcomes in the testing phase. According the author [46], the high number of *TP* and *TN* in the training phase helps to optimize the performance of a model. Point to note that before proceeding to the training phase, the k-fold cross-validation and attribute filtering manuals 'log-linear' are also introduced in the preprocessing phase which perfectly aligned with the proposed HEA in the training phase. Moreover, a permutation operator is also initialized in the preprocessing phase which can enhance the search space of the model.

Moreover, the non-systematic representation, such as RAN*k*SAT, where *K* has different combinations, creates more options to get upper qualitative induced logic, which has the highest *TP* and *TN*. It is worth noting that for the abovementioned datasets, existing S2SATRA, P2SATRA, and A2SATRA also generate close *Acc* values but cannot exceed R-HEARA because of this non-systematic RAN*k*SAT combination. Different order combinations of RAN*k*SAT clauses can keep different accuracy values. Moreover, the combination of the third-order clause and the second-order clause increases the rate of accuracy compared to other existing models. This is because of k=(*2,3*) combinations of R-HEARA, which have a higher probability of getting satisfied interpretations compared to other combinations.

According to the Table 6 and Figure 2, no value for R-HEARA achieved an accuracy value less than 70%. Meanwhile, the *Acc* value is higher when the number of *TP*, *TN* are getting higher and the number of *FP*, *FN* are in the vice versa. Hence, getting *Acc* 70% indicates that the induced logic produced by R-HEARA is able to achieve higher *TP*, *TN* in most of the datasets. One of the reasons for the high *TP* and *TN* is the inclusion of the preprocessing phase, where the best attributes are selected through the log-linear technique [26]. Based on the various attributes in the dataset, log-linear successfully found ten attributes that were associated with the output decision. These ten attributes give optimal p-value, and these can directly affect getting optimal final neuron states, and as a result, the value of *Acc* is raised.

Table 6. Accuracy (*Acc*) for all logic mining models. The bracket indicates the ratio of improvement (a negative ratio implies the method outperformed the proposed method). Higher value and lower rank indicates better result.

List	R-HEARA	A2SATRA	L2SATRA	S2SATRA	P2SATRA	2SATRA	E2SATRA	RA	3SATRA
D1	**0.7861**	0.6371	0.4682	0.5761	0.5891	0.2851	0.4389	0.3950	0.4529
	(*k*=2,3)	(0.062)	(0.231)	(0.123)	(0.110)	(0.414)	(0.261)	(0.305)	(0.247)
D2	**0.7947**	0.7544	0.5233	0.6626	0.7584	0.6842	0.4774	0.4681	0.5874
	(*k*=2,3)	(0.041)	(0.272)	(0.132)	(0.036)	(0.111)	(0.318)	(0.327)	(0.207)
D3	0.7014	0.66119	0.5865	0.7412	**0.7441**	0.5978	0.6107	0.5639	0.6200
	(*k*=2,3)	(0.040)	(0.114)	(–0.039)	(–0.042)	(0.103)	(0.090)	(0.137)	(0.081)
D4	0.7875	**0.8376**	0.6414	0.7175	0.6825	0.5175	0.5000	0.5575	0.5550
	(*k*=2,3)	(–0.050)	(0.140)	(0.071)	(0.105)	(0.270)	(0.287)	(0.230)	(0.232)
D5	**0.7739**	0.6600	0.5275	0.6333	0.6333	0.4760	0.4978	0.5376	0.5666
	(*k*=2,3)	(0.113)	(0.246)	(0.140)	(0.142)	(0.297)	(0.276)	(0.236)	(0.207)
D6	**0.7514**	0.6907	0.5633	0.6135	0.6995	0.5707	0.5381	0.4895	0.6122
	(*k*=2,3)	(0.040)	(0.214)	(0.124)	(0.042)	(0.166)	(0.201)	(0.251)	(0.126)
D7	**0.7500**	0.6766	0.5643	0.6983	0.6791	0.5643	0.6976	0.5477	0.5182
	(*k*=2,3)	(0.0734)	(0.186)	(0.051)	(0.079)	(0.181)	(0.514)	(0.201)	(0.231)
D8	0.7903	0.7191	0.5290	**0.7922**	0.7825	0.5612	0.5627	0.6065	0.5048
	(*k*=2,3)	(0.073)	(0.185)	**(0.048)**	(0.0185)	(0.231)	(0.234)	(0.190)	(0.291)
D9	**0.7697**	0.7052	0.4201	0.6976	0.7052	0.4064	0.4476	0.5380	0.5604
	(*k*=2,3)	(0.0645)	(0.346)	(0.072)	(0.069)	(0.363)	(0.322)	(0.231)	(0.209)
D10	**0.8681**	0.7869	0.7429	0.6137	0.7674	0.6381	0.6124	0.5807	0.6943
	(*k*=2,3)	(0.081)	(0.125)	(0.254)	(0.107)	(0.234)	(0.255)	(0.287)	(0.173)
D11	0.7942	0.8666	0.3290	0.8114	**1.0000**	0.3447	0.6194	0.7445	0.8285
	(*k*=2,3)	(–0.073)	(0.471)	(–0.029)	**(–0.205)**	(0.455)	(0.181)	(0.051)	(–0.034)
D12	0.7992	0.6476	0.5533	**0.8558**	0.6512	0.5337	0.5700	0.5447	0.6471
	(*k*=2,3)	(0.151)	(0.245)	**(–0.056)**	(0.148)	(0.265)	(0.229)	(0.254)	(0.152)
D13	**0.7191**	0.5814	0.5089	0.5863	0.5841	0.5089	0.5101	0.5307	0.5411
	(*k*=2,3)	(0.137)	(0.210)	(0.132)	(0.135)	(0.219)	(0.208)	(0.188)	(0.178)
D14	**0.9773**	0.9617	0.6588	0.9705	0.9705	0.6588	0.9294	0.2117	0.9764
	(*k*=2,3)	(0.015)	(0.318)	(0.006)	(0.006)	(0.318)	(0.047)	(0.760)	(0.0008)
D15	**0.7070**	0.5886	0.3870	0.6991	0.6573	0.3856	0.6858	0.6454	0.5809
	(*k*=2,3)	(0.118)	(0.321)	(0.007)	(0.049)	(0.321)	(0.021)	(0.061)	(0.126)
D16	**0.9150**	0.6826	0.3313	0.7850	0.6456	0.3313	0.4945	0.6144	0.4760
	(*k*=2,3)	(0.212)	(0.561)	(0.110)	(0.243)	(0.560)	(0.400)	(0.280)	(0.419)
D17	**0.7527**	0.7463	0.4257	0.7085	0.7501	0.4324	0.4479	0.5148	0.6137
	(*k*=2,3)	(0.006)	(0.326)	(0.044)	(0.002)	(0.325)	(0.304)	(0.237)	(0.138)
D18	**0.7722**	0.7407	0.6092	0.6981	0.7481	0.6500	0.6190	0.6013	0.6055
	(*k*=2,3)	(0.031)	(0.162)	(0.074)	(0.024)	(0.125)	(0.153)	(0.170)	(0.166)
D19	**0.8345**	0.5691	0.4116	0.8344	0.6127	0.6481	0.7451	0.2798	0.5697
		(0.265)	(0.422)	(0.0001)	(0.221)	(0.186)	(0.089)	(0.554)	(0.264)
D20	**0.7048**	0.5504	0.5342	0.7047	0.5762	0.6451	0.7019	0.4200	0.6146
	(*k*=2,3)	(0.155)	(0.171)	(0.0001)	(0.128)	(0.059)	(0.002)	(0.284)	(0.091)
(+/=/-)	–	18/0/2	20/0/0	15/2/3	18/0/2	20/0/0	20/0/0	20/0/0	19/0/1
Avg.	**0.7893**	0.7032	0.5158	0.7200	0.7118	0.5220	0.5853	0.5196	0.6063
STD	**0.0710**	0.1042	0.1086	0.0991	0.1130	0.1222	0.1228	0.1214	0.1186
Min	0.7000	0.5504	0.3313	0.5761	0.5762	0.2851	0.4389	**0.2798**	0.4529
Max	0.9773	0.8376	0.6414	0.8558	**1.0000**	0.6842	0.9294	0.7445	0.9764
Rank (Avg)	**1.5**	3.53	7.27	3.63	2.98	6.98	5.95	7.38	5.85

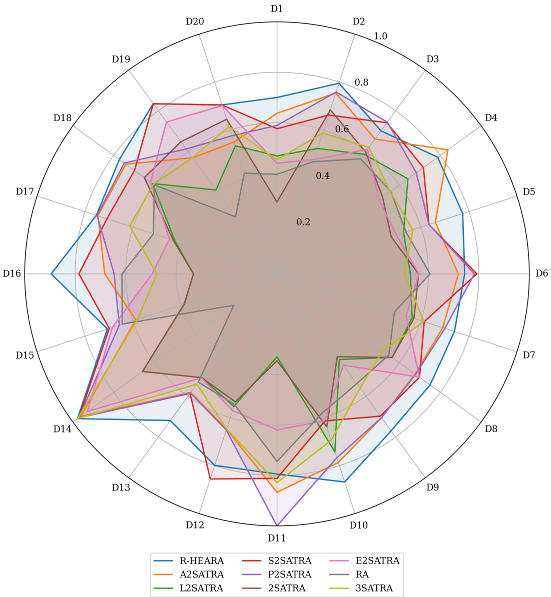

Figure 2. Accuracy for all logic mining models.

Additionally, STD is also calculated for all the logic mining models to understand how the value is spread out in a given dataset. In STD, the lower value represents the better model. Here, it is also seen that the lower STD value occurs for the proposed R-HEARA model, which is 0.07. Meanwhile, the Friedman test is applied for all the datasets with χ^2 and the degree of freedom df. The p-value for *Acc* is $p \leq 0.05$. Hence, the null hypothesis of equal performance for all the logic mining models was rejected. R-HEARA achieves the lowest average rank compared to other existing models.

5.2. *Precision (Prec) analysis*

The ‘Precision’ performance metric measures how accurately the *TP* is predicted and how *FP* can be lesser to achieve high *TP* in the model. In the work by [47], for large datasets classifications and predictions: Precision metric is very impactful for performance analysis of a model. Here, in this article, the Precision (*Prec*) value for R-HEARA and other logic mining models is shown in Table 7 and Figure 3. In this case, the suggested R-HEARA outperforms existing logic mining models in 17 of 20 datasets. This demonstrates that R-HEARA is extremely capable of acquiring a greater quantity of *TP*. This happened due to the inclusion of the RAN*k*SAT logic in the model. The non-systematic behavior of RAN*k*SAT strategy created the opportunity for achieving higher *TP*. The *Prec* value got higher when different combinations of RAN*k*SAT were employed in the model with different order of clauses. The multiple SAT arrangements create the opportunity to reduce the number of *FP*, which impact increasing the rate of the Precision.

It is worth noting that 14 datasets are very near to achieving *Prec*=1, indicating that R-HEARA can correctly forecast 100% of positive instances, one of the possibilities for the inclusion of dynamic metaheuristic in the training phase. Especially, the local search operators such as positive advertisement and caretaker party operators did a dynamic role in this model. Positive advertisement operator prior focus is on getting high *TP* whereas the caretaker party stored the highest fitness value candidate that lead to high *TP*. Importantly, the shift mutation mechanism of the caretaker party operator increases the possibility of *TP* and decreases the number of *FP*.

Hence, the highest fitness candidates preserved by the caretaker party lead to diverse P^{best} logic and generates 100% accurate induced logic. From the dataset D8, the performance of R-HEARA ranges from 37% to 90%, which visualizes the effectiveness of the proposed model. This has happened due to the multi-level processing layers of R-HEARA. In the preprocessing phase, a log-linear method is employed. The log-linear method unbiasedly traces the best attributes from the dataset. Since the D8 dataset has 21 attributes, it is difficult to choose 10 best attributes, and the log-linear method perfectly scaled the best attributes which correctly aligned with the $P^{best}_{R-HEARA}$. Along with robust HEA as well as dynamic RAN*k*SAT logical structure, the *Prec* value range makes such a distance from other logic mining models. The Friedman test results indicate that the average rank for R-HEARA is 1.42, the lowest among the existing logic mining models. This statistically validates the superiority of R-HEARA, demonstrating its strong ability to differentiate between positive and negative outcomes.

Table 7. Precision (*Prec*) for all logic mining models. Numbers in parentheses show improvement ratios (negative means better than proposed method). Higher value and lower rank indicates better result.

List	R-HEARA	A2SATRA	L2SATRA	S2SATRA	P2SATRA	2SATRA	E2SATRA	RA	f3SATRA
D1	**0.799**	0.242 (0.557)	0.489 (0.316)	0.430 (0.373)	0.473 (0.322)	0.345 (0.454)	0.445 (0.354)	0.523 (0.287)	0.592 (0.209)
D2	0.662	0.397 (0.263)	0.419 (0.244)	0.673 (-0.011)	0.383 (0.285)	0.196 (0.465)	0.424 (0.297)	0.533 (0.133)	**0.815** (-0.155)
D3	0.796	0.769 (0.027)	0.614 (0.175)	0.744 (0.051)	0.753 (0.044)	0.632 (0.165)	0.684 (0.115)	0.621 (0.174)	**0.847** (-0.049)
D4	**0.977**	0.646 (0.328)	0.596 (0.385)	0.750 (0.233)	0.729 (0.255)	0.352 (0.621)	0.411 (0.573)	0.573 (0.404)	0.665 (0.315)
D5	**0.847**	0.648 (0.196)	0.384 (0.461)	0.625 (0.222)	0.734 (0.113)	0.778 (0.069)	0.612 (0.235)	0.489 (0.359)	0.831 (0.016)
D6	**1.000**	0.579 (0.419)	0.386 (0.613)	0.537 (0.464)	0.529 (0.435)	0.639 (0.364)	0.545 (0.452)	0.484 (0.514)	0.710 (0.292)
D7	0.938	0.953 (0.015)	**0.965** (0.024)	0.952 (0.017)	0.953 (0.012)	0.965 (0.024)	0.951 (0.016)	0.406 (0.533)	0.828 (0.113)
D8	**1.000**	0.108 (0.897)	0.530 (0.474)	0.101 (0.891)	0.129 (0.861)	0.479 (0.516)	0.475 (0.519)	0.296 (0.698)	0.628 (0.371)
D9	**0.792**	0.640 (0.155)	0.492 (0.292)	0.703 (0.089)	0.640 (0.154)	0.474 (0.316)	0.397 (0.389)	0.497 (0.291)	0.687 (0.115)
D10	**0.975**	0.819 (0.164)	0.778 (0.234)	0.840 (0.132)	0.761 (0.214)	0.560 (0.412)	0.695 (0.291)	0.500 (0.483)	0.800 (0.185)
D11	**1.000**	0.847 (0.152)	0.350 (0.653)	0.857 (0.143)	**1.000** (0)	0.229 (0.773)	0.516 (0.481)	0.735 (0.272)	**1.000** (0)
D12	**0.928**	0.514 (0.409)	0.437 (0.483)	0.917 (0.011)	0.480 (0.446)	0.531 (0.392)	0.352 (0.571)	0.432 (0.491)	0.833 (0.091)
D13	**0.952**	0.454 (0.497)	0.484 (0.447)	0.418 (0.539)	0.570 (0.378)	0.484 (0.441)	0.435 (0.511)	0.478 (0.477)	0.844 (0.111)
D14	**1.000**	0.967 (0.033)	0.479 (0.521)	0.969 (0.038)	0.977 (0.023)	0.479 (0.521)	0.946 (0.053)	0.299 (0.601)	0.962 (0.038)
D15	**0.930**	0.531 (0.391)	0.513 (0.413)	0.420 (0.503)	0.481 (0.434)	0.514 (0.412)	0.207 (0.723)	0.838 (0.091)	0.781 (0.152)
D16	**0.960**	0.880 (0.078)	0.832 (0.126)	0.587 (0.364)	0.285 (0.665)	0.832 (0.114)	0.506 (0.445)	0.196 (0.752)	0.607 (0.343)
D17	**0.960**	0.850 (0.110)	0.218 (0.732)	0.725 (0.234)	0.851 (0.109)	0.329 (0.623)	0.316 (0.634)	0.485 (0.476)	0.807 (0.149)
D18	**0.911**	0.816 (0.098)	0.894 (0.021)	0.897 (0.011)	0.803 (0.113)	0.862 (0.041)	0.891 (0.021)	0.701 (0.199)	0.850 (0.061)
D19	**0.936**	0.749 (0.187)	0.880 (0.056)	0.911 (0.028)	0.863 (0.068)	0.590 (0.347)	0.791 (0.147)	0.154 (0.781)	0.747 (0.181)
D20	**0.914**	0.798 (0.124)	0.786 (0.126)	0.890 (0.021)	0.714 (0.198)	0.659 (0.247)	0.696 (0.218)	0.498 (0.417)	0.722 (0.182)
(+/=/-)		20/0/0	19/0/1	20/0/0	19/1/0	20/0/0	20/0/0	20/0/0	18/1/2
Avg.	**0.914**	0.660	0.576	0.697	0.655	0.546	0.565	0.487	0.778
STD	**0.090**	0.232	0.200	0.227	0.235	0.206	0.212	0.168	0.108
Min	0.662	0.242	0.218	0.410	0.129	0.229	0.207	0.154	0.628
Max	**1.000**	0.967	0.965	0.969	**1.000**	0.965	0.951	0.838	**1.000**
Rank	**1.42**	4.85	5.71	4.55	5.12	6.15	6.71	7.12	3.45

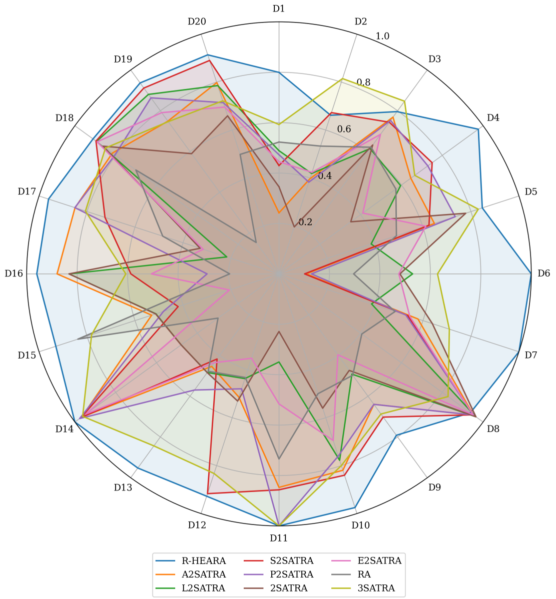

Figure 3. Precision results for all logic mining models.

5.3. *MCC analysis*

The *MCC* assesses the quality of a model by considering the combined performance of all components in the confusion matrix. As a correlation coefficient between observed and predicted binary classifications, *MCC* ranges from -1, indicating a perfect inverse prediction, to 1, indicating a perfect prediction [48]. The *MCC* analysis for each logic mining models are presented in Figure 4 and Table 8.

In this study, R-HEARA obtained the higher *MCC* value in 11 of 20 datasets. Since the value of *MCC* for R-HEARA model is higher, it can be said that the proposed model is better than the random classifier. Moreover, the proposed model illustrates that how RAN*k*SAT is effective compared to systematic *k*SAT. This is a result of the dynamic mechanism of HEA which has a powerful advertising campaign operator that reflects in the training phase and accelerates five distinct P^{best} logics.

The multi-objective functions of HEA, especially the shift mutation of the caretaker party in the advertisement campaign, govern the model to get 100% unique ideal solution strings. These ideal solution strings connect to each P^{best} logic. At the conclusion of the simulation, each P^{best} logic generates several numbers of diverse induced logic. Moreover, the logical diversity of the multi-objective HEA rightly classified the number of *TP* and *TN* of R-HEARA and produced $P^{test}_{R-HEARA}$ that can represent the perfect prediction.

Among the 20 datasets, two datasets (D9 and D16) approach the highest value $MCC \approx 1$. According to [48, 49], if the *MCC* value approaches 1, the prediction generated by the logic mining model is estimated as 'best', while if the *MCC* value approaches to -1 then it predicts the model considered as 'worst'. In this regard it can be said that the proposed R-HEARA is a better-quality model than the random classifier and it is capable of producing near-perfect predictions in the $P^{test}_{R-HEARA}$. Moreover, the RAN*k*SAT combinations properly scale the number of *TP* and *TN* for which the metric *MCC* has the probability in getting a higher value. In this regard, analyzing the *MCC* value of the mentioned datasets ensures the effectiveness of the confusion matrix classified by R-HEARA based on the induced logic.

Several datasets, such as D1 dataset, achieved low *MCC* values. Even though the *MCC* value is modest, it is also superior to other logic mining models. The multi combinations of the RAN*k*SAT logical rule cause the scales of *TP* and *TN* to be higher, and the *MCC* value is likewise greater. According to the tabulated results, R-HEARA is capable of producing *MCC* values for all 20 datasets, whereas other models cannot create *MCC* values consistently. In this case, it can be stated that the other models are considered to be as random classifier only.

In addition, R-HEARA maintains its place at the top by examining the average value and maximum value for each dataset. The STD number is also the lowest among other models for logic mining. Meanwhile, rank analysis through Friedman test rank is examined for all the datasets with $\alpha = 0.05$ and the degree of freedom, $df = 8$. The p-value for $MCC \leq 0.05(\chi^2 = 46.108)$. Here, the null hypothesis of equal performance for all the logic mining models was rejected. The top rank position is also on behalf of R-HEARA by acquiring the lowest value which is 2.00.

Table 8. *MCC* for all logic mining models. The bracket indicates the ratio of improvement and a negative ratio implies the method outperforms the proposed method. Higher value and lower rank indicates better result.

List	R-HEARA	A2SATRA	L2SATRA	S2SATRA	P2SATRA	2SATRA	E2SATRA	RA	3SATRA
D1	**0.2454**	0.0661 (0.176)	-0.071 (0.312)	0.1498 (0.092)	0.1362 (0.114)	-0.3927 (0.642)	-0.1142 (0.352)	-0.1463 (0.393)	-0.0384 (0.283)
D2	0.3440	**0.3727** (-0.027)	-0.0171 (0.353)	0.4138 (-0.072)	0.4095 (-0.061)	-0.0269 (0.363)	0.14299 (0.192)	-0.0556 (0.392)	0.1654 (0.183)
D3	0.3985	0.369 (0.036)	-0.3713 (0.765)	**0.7654** (-0.367)	0.2768 (0.124)	-0.3713 (0.772)	-0.2095 (0.597)	-0.0727 (0.471)	0.2321 (0.172)
D4	**0.5803**	0.4319 (0.148)	0.0164 (0.557)	0.4648 (0.124)	0.39907 (0.181)	-0.0328 (0.609)	-0.0052 (0.587)	0.1316 (0.451)	0.1304 (0.452)
D5	**0.5439**	0.0791 (0.459)	0.022 (0.521)	0.1223 (0.423)	0.2302 (0.314)	0.022 (0.521)	0.0811 (0.461)	0.1134 (0.431)	0.0462 (0.492)
D6	0.1017	-0.0767 (0.281)	**0.2081** (0.484)	-0.0202 (0.342)	-0.0111 (0.211)	0.0313 (0.451)	0.0306 (0.533)	-0.041 (0.613)	0.0733 (0.352)
D7	**0.5908**	0.3079 (0.289)	0.1058 (0.487)	0.2508 (0.341)	0.3791 (0.221)	0.1404 (0.425)	0.064 (0.614)	-0.0197 (0.743)	0.2383 (0.351)
D8	0.4984	0.4765 (0.025)	0.339 (0.164)	0.502 (-0.003)	0.4795 (0.018)	0.339 (0.165)	**0.5021** (-0.003)	0.0357 (0.455)	0.1775 (0.323)
D9	**0.9125**	0.395 (0.523)	0.888 (0.032)	0.3876 (0.533)	0.395 (0.522)	-0.1839 (1.09)	-0.099 (1.00)	0.0879 (0.833)	0.1301 (0.781)
D10	**0.6056**	-	-	0.05314 (0.543)	-	-	-	-	-
D11	0.6768	0.89138 (-0.216)	0.71076 (-0.034)	**0.91144** (-0.234)	0.8913 (-0.213)	-0.4134 (1.09)	-0.0051 (0.683)	0.16894 (0.513)	0.38574 (0.294)
D12	0.4286	0.17045 (0.257)	0.1134 (0.316)	**0.6413** (-0.208)	0.1423 (0.272)	-0.0135 (0.441)	0.0416 (0.394)	-0.008 (0.434)	0.2809 (0.143)
D13	**0.3096**	0.1509 (0.148)	0.1171 (0.187)	0.1632 (0.147)	0.1685 (0.137)	0.0133 (0.2983)	0.003 (0.251)	0.0612 (0.242)	0.1442 (0.146)
D14	0.2220	0.8937 (-0.667)	-	0.90626 (-0.681)	**0.90718** (-0.685)	0.8689 (-0.642)	0.8441 (-0.623)	-0.804 (1.02)	0.9399 (-0.713)
D15	**0.1718**	0.0306 (0.141)	-	0.0059 (0.172)	0.005 (0.172)	-0.0526 (0.224)	0.02906 (0.144)	-0.0443 (0.213)	0.0572 (0.113)
D16	**0.9325**	0.623 (0.312)	-	-0.0192 (1.03)	0.623 (0.313)	-0.305 (1.06)	0.578 (0.354)	0.432 (0.501)	0.34846 (0.491)
D17	0.42963	0.4642 (-0.031)	-	0.3796 (0.05)	**0.47313** (-0.046)	-0.088 (0.513)	-0.0394 (0.471)	0.0089 (0.423)	0.1237 (0.301)
D18	**0.5395**	0.5016 (0.032)	-	0.4621 (0.073)	0.5098 (0.033)	0.3777 (0.164)	-	-	0.2926 (0.243)
D19	**0.2884**	0.0864 (0.202)	0.046 (0.242)	-	0.07645 (0.247)	0.077 (0.205)	-	-	0.0738 (0.215)
D20	0.3148	0.0856 (0.221)	0.0713 (0.242)	**0.5641** (-0.243)	0.1 (0.214)	-	-	-	0.069 (0.251)
(+/=/-)		15/0/5	19/0/1	14/0/6		19/0/1	18/0/2	20/0/0	19/0/1
Avg.	**0.4277**	0.3326	0.1556	0.3739	0.3469	-0.0006	0.1152	-0.0095	0.2037
Std	**0.17**	0.26	0.3	0.28	0.25	0.3	0.27	0.24	0.2
Min	0.222	-0.0767	-0.071	-0.0192	-0.011	-0.088	-0.0052	-0.0443	-0.0384
Max	0.9325	0.8937	0.7107	0.9114	0.9071	0.86899	0.8441	0.1689	**0.93**
Rank	**2.00**	3.82	5.77	3.00	3.64	7.59	6.82	7.09	5.27

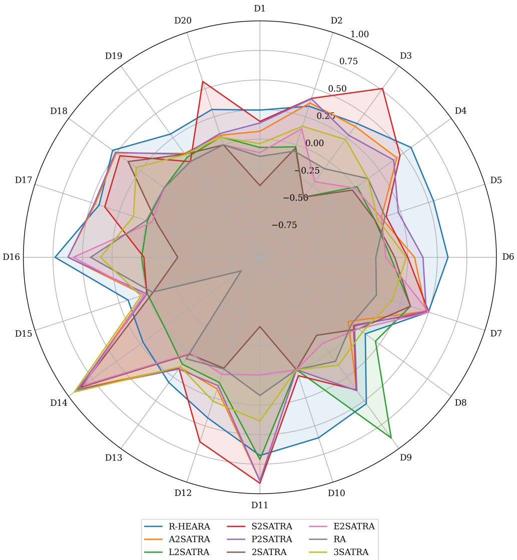

Figure 4. *MCC* results for all logic mining models.

5.4. *Specificity (Spec) analysis*

Figure 5 and Table 9 illustrate the Specificity results for all logic mining models. Specificity measures the proportion of *TN* relative to the total number of *TN* and *FP*. In terms of Specificity (*Spec*), the proposed R-HEARA model outperforms other logic mining models in 16 out of 20 datasets, demonstrating its ability to achieve a high number of *TN*. This visualizes that with this performance metric, the R-HEARA proved the best position compared to other existing models. Notably, 16 datasets were near to achieve the highest specificity value, that is, Spec=1. This validates that R-HEARA undoubtedly produces all the correct *TN* during the testing phase of DHNN. This classy result happened due to the multi-objective concept of HEA in the training mechanism. The proper balance of the exploration and exploitation of the operators correctly identifies the *TP* and *TN*.

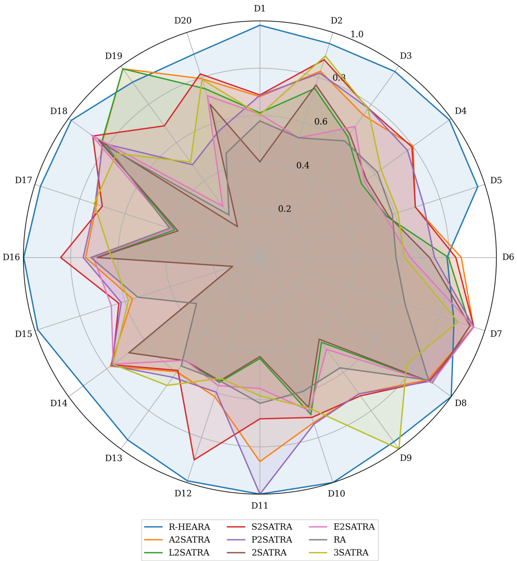

Figure 5. Specificity results for all logic mining models.

Table 9. Specificity value for all logic mining models. The bracket indicates the ratio of improvement and a negative ratio implies the method outperforms the proposed method. Higher value and lower rank indicates better result.

List	R-HEARA	A2SATRA	L2SATRA	S2SATRA	P2SATRA	2SATRA	E2SATRA	RA	3SATRA
D1	**0.9817**	0.6801 (0.301)	0.6113 (0.374)	0.6872 (0.292)	0.6854 (0.294)	0.4046 (0.577)	0.6058 (0.374)	0.5772 (0.404)	0.6031 (0.378)
D2	**0.9508**	0.8281 (0.123)	0.7511 (0.198)	0.8789 (0.068)	0.8195 (0.136)	0.7669 (0.178)	0.5321 (0.416)	0.5319 (0.418)	0.8959 (0.051)
D3	**0.9714**	0.744 (0.232)	0.632 (0.334)	0.7798 (0.192)	0.7796 (0.187)	0.6460 (0.327)	0.6847 (0.287)	0.6091 (0.353)	0.7772 (0.194)
D4	**0.9898**	0.8011 (0.183)	0.5305 (0.459)	0.7949 (0.189)	0.7710 (0.227)	0.5578 (0.432)	0.5461 (0.442)	0.6139 (0.383)	0.6327 (0.367)
D5	**0.9686**	0.6905 (0.284)	0.5639 (0.396)	0.6911 (0.274)	0.7273 (0.244)	0.5712 (0.395)	0.5574 (0.412)	0.5890 (0.379)	0.6136 (0.355)
D6	0.7910	**0.8516** (-0.064)	0.7973 (-0.006)	0.8286 (-0.031)	0.7383 (0.051)	0.7172 (0.071)	0.6355 (0.155)	0.5755 (0.218)	0.6119 (0.185)
D7	0.8630	0.9500 (-0.083)	0.9342 (-0.071)	**0.9511** (-0.088)	0.9502 (-0.087)	0.934 (0.071)	0.9511 (-0.081)	0.646 (0.221)	0.881 (-0.018)
D8	**1.0000**	0.8788 (0.121)	0.9013 (0.094)	0.885 (0.118)	0.8898 (0.119)	0.8996 (0.111)	0.8986 (0.111)	0.8818 (0.114)	0.7667 (0.233)
D9	0.9630	0.7123 (0.251)	0.4429 (0.521)	0.7228 (0.240)	0.7123 (0.250)	0.4267 (0.542)	0.4796 (0.483)	0.5764 (0.391)	**1.0000** (-0.036)
D10	**1.0000**	0.7341 (0.261)	0.6981 (0.295)	0.7107 (0.278)	0.7401 (0.251)	0.6666 (0.333)	0.6809 (0.311)	0.5943 (0.391)	0.6680 (0.332)
D11	**1.0000**	0.8625 (0.137)	0.4267 (0.571)	0.6817 (0.321)	**1.0000** (0)	0.4183 (0.581)	0.5535 (0.442)	0.6162 (0.381)	0.5843 (0.411)
D12	**0.9922**	0.6178 (0.374)	0.5548 (0.444)	0.8991 (0.089)	0.5981 (0.386)	0.5478 (0.438)	0.5694 (0.401)	0.5477 (0.443)	0.5366 (0.451)
D13	**0.9521**	0.5968 (0.363)	0.5389 (0.413)	0.5899 (0.364)	0.6259 (0.327)	0.5389 (0.414)	0.5372 (0.412)	0.5666 (0.379)	0.6680 (0.282)
D14	**0.9230**	0.7818 (0.141)	0.6848 (0.238)	0.7712 (0.151)	0.7765 (0.149)	0.6848 (0.242)	0.7666 (0.155)	0.3312 (0.591)	0.7678 (0.147)
D15	**0.9878**	0.5666 (0.421)	0.1216 (0.867)	0.6269 (0.357)	0.6164 (0.371)	0.1210 (0.858)	0.6605 (0.332)	0.5425 (0.448)	0.5843 (0.401)
D16	**0.9986**	0.7368 (0.261)	0.6878 (0.312)	0.8423 (0.163)	0.7469 (0.251)	0.6878 (0.312)	0.7041 (0.291)	0.7147 (0.282)	0.6273 (0.372)
D17	**0.9755**	0.7266 (0.241)	0.3784 (0.592)	0.7009 (0.271)	0.7324 (0.242)	0.3644 (0.613)	0.3905 (0.582)	0.4013 (0.571)	0.7368 (0.231)
D18	**0.9862**	0.8247 (0.151)	0.8274 (0.151)	0.8737 (0.114)	0.8216 (0.165)	0.8428 (0.138)	0.881 (0.106)	0.8 (0.181)	0.7488 (0.243)
D19	0.9161	**0.9861** (-0.070)	0.9858 (-0.065)	0.6879 (0.235)	0.4846 (0.436)	0.161 (0.763)	0.2674 (0.652)	0.2222 (0.691)	0.4989 (0.421)
D20	**0.9038**	0.7946 (0.101)	0.7502 (0.152)	0.8158 (0.081)	0.5621 (0.337)	0.6808 (0.223)	0.7193 (0.178)	0.4631 (0.437)	0.7921 (0.118)
(+/=/-)		17/0/3	17/0/3	18/0/2	18/1/1	20/0/0	19/0/1	20/0/0	18/0/2
Avg.	**0.9557**	0.7682	0.6409	0.7711	0.7389	0.5819	0.6311	0.5721	0.6997
STD	**0.051**	0.107	0.200	0.091	0.121	0.210	0.161	0.141	0.122
Min	0.7910	0.5666	0.1216	0.5899	0.4846	**0.121**	0.2674	0.2222	0.4989
Max	**1.0000**	0.9861	0.9858	0.9511	**1.0000**	0.934	0.9511	0.881	**1.0000**
Avg. Rank	**1.68**	3.73	6.35	3.05	4.05	7.15	6.00	7.45	5.55

The diversity of the logical rule in HEA with the multi-objective concept, especially the caretaker party of the advertisement campaign, keeps the best combination in fitness and diversity. With this strategy, proposed R-HEARA enhances the negative states of the logical rule, which excelled the ratio of *TN*. Correspondingly, the permutation operator also plays an important role in increasing the value of *TN*, focuses less on *FP*, and helps the total mechanism in finding best quality induced logic [25]. This justifies the superiority of R-HEARA in partitioning *TP* and *TN* cases, which are very crucial in the field of logic mining. From the above discussions, it is observed that the R-HEARA model continues to perform better in analyzing the performance with the 'Specificity' metrics.

From the datasets D15, D16, D17 point of view, the performance of the proposed R-HEARA exceeds 20%–87% compared to other logic mining models. This has happened due to the log-linear concept in the preprocessing phase with k-cross validation as well as non-systematic RAN*k*SAT logical structure. Meanwhile, the three combinations of RAN*k*SAT (k=1,2,3 / 2,3/ 1,3) create more options for getting the best logical structure, which exemplifies the best induced logic at the end. In this regard, the proposed R-HEARA caters the weakness for the work of [25] by embedding optimal attributes in the RAN*k*SAT and as well as a better training model of HEA.

Furthermore, the average value (Avg.) of specificity in R-HEARA for all datasets extremely exceeds the other logic mining models. Even the STD value and the maximum (Max) value also keep top position, which clearly justifies that R-HEARA outperforms the existing models by its effective mechanism. In addition to the analysis of the rank through the Friedman test, the top position is also kept by R-HEARA with the rank 1.68. Overall, the proposed model R-HEARA clearly acknowledges the work by the work of [50], which shows that scoring the number of TN can be very useful if the objectives of the classification of a model are focused on obtaining excellent results.

5.5. Statistical significance and ablation study

To evaluate whether the observed differences in performance metrics are statistically significant or not, this article adds a Nemenyi post-hoc test and a critical difference diagram for R-HEARA with all baseline models based on the accuracy results of this experiment, since the accuracy results indicate the overall performance scenario of the models. The results of the Nemenyi post-hoc test (Table 10) further reinforce the superiority of the proposed R-HEARA model. Specifically, R-HEARA demonstrates statistically significant improvements, with average rank differences exceeding the critical difference (CD) threshold. The threshold of CD is mathematically calculated by the performance of two models, whose corresponding average ranks differ by at least the CD [51]. By Eq (5.1), the CD threshold can be calculated,

$$CD = q_{\alpha} \cdot \sqrt{\frac{X(X+1)}{6n}}. \tag{5.1}$$

Here, X is the number of algorithms, n is the number of datasets, and q_α is the critical range distribution value based on the classifiers ($\alpha \leq 0.05$). In this experiment, the CD threshold value is 1.96. Additionally, the CD diagram shown in Figure 6 as: R-HEARA is statistically significant than almost all other baseline models. Its rank is clearly separated from the others by more than the rank CD threshold (RCD) and is 3.41, which is calculated as: RCD = Lowest Rank + CD = 1.45 + 1.96 = 3.41. Notably, the Nemenyi post-hoc test shows that S2SATRA and A2SATRA are not statistically significant compared to R-HEARA within the RCD threshold, likely due to the incorporation of the preprocessing phase in these base models. Together, the Nemenyi test table and the CD diagram provide complementary perspectives: the table confirms precise pairwise comparisons, while the diagram offers an intuitive visualization of algorithm performance across datasets. These additional analyses confirm that the superior performance of R-HEARA is both statistically significant and practically meaningful. In the CD diagram, the symbol $\updownarrow$ indicates the rank position of the individual model, and the blue lines indicate the threshold line of the analysis.

Table 10. Nemenyi post-hoc test result for accuracy when $\alpha = 0.05$, where ✓ = significant difference, ns = not significant.

Algorithms ↓ vs. →	R-HEARA	P2SATRA [25]	S2SATRA [42]	A2SATRA [24]	3SATRA [43]	E2SATRA [44]	2SATRA [26]	L2SATRA [23]	RA [22]
R-HEARA	–	✓	ns	ns	✓	✓	✓	✓	✓
P2SATRA	✓	–	ns	ns	ns	ns	ns	ns	ns
S2SATRA	ns	ns	–	ns	ns	ns	ns	ns	ns
A2SATRA	ns	ns	ns	–	ns	ns	ns	ns	ns
3SATRA	✓	ns	ns	ns	–	ns	ns	ns	ns
E2SATRA	✓	ns	ns	ns	ns	–	ns	ns	ns
2SATRA	✓	ns	ns	ns	ns	ns	–	ns	ns
L2SATRA	✓	ns	ns	ns	ns	ns	ns	–	ns
RA	✓	ns	ns	ns	ns	ns	ns	ns	–

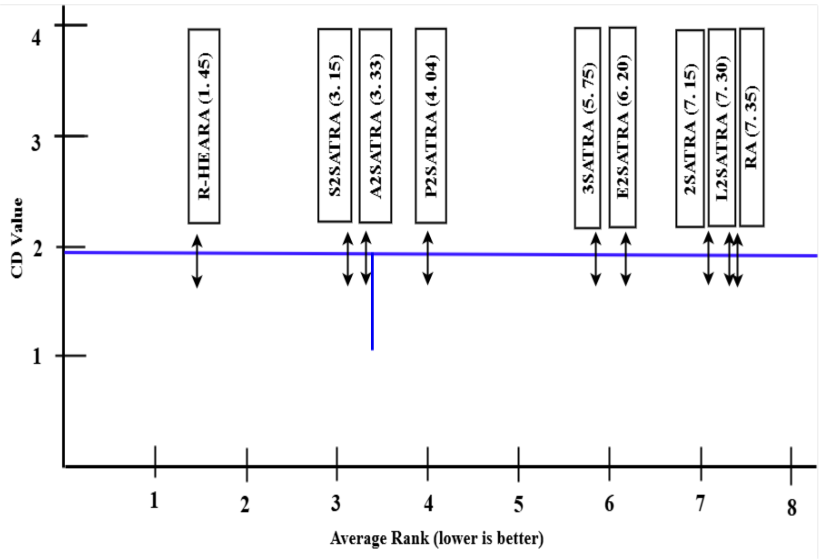

Figure 6. Critical difference diagram for the logic mining models. The symbol ↕ indicates the rank position of the model, and the blue lines indicate the threshold line of the analysis.

Figure 7 presents the ablation study of the R-HEARA, where the stacked contribution of the individual phases is depicted. Starting from the base RA model (0.5196), the addition of log-linear attribute selection improves the accuracy to 0.6063. The integration of the hybrid election algorithm yields the highest average accuracy of 0.7118. Incorporating the permutation operator further raises performance to 0.7300, while RAN*k*SAT ordering provides an additional boost to achieve the final highest average accuracy. Hence, the full R-HEARA model with the highest average accuracy of 0.7893. This progression highlights that while each component contributes incrementally, it is their synergistic integration that enables R-HEARA to vanquish other models.

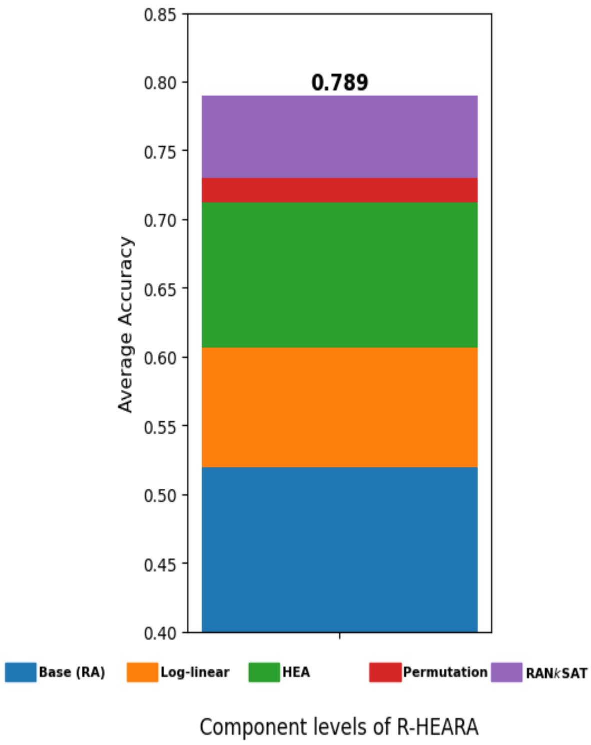

Figure 7. Stacked contributions of components in R-HEARA.

In general, R-HEARA (0.7893) and A2SATRA (0.7032) show that log-linear selection contributes significantly to performance improvement, but A2SATRA only focused on systematic structures, while R-HEARA employs non-systematic structures (Non-systematic VS Systematic). Besides this, another type of attribute selection method (Correlation analysis) input by S2SATRA performs reasonably well on some datasets, but log-linear ensures more consistent generalization (log-linear vs correlation). Additionally, utilizing permutation operator-P2SATRA achieves competitive accuracy (0.7118), occasionally outperforming R-HEARA on datasets where pairwise literal interactions dominate. However, R-HEARA surpasses it overall, showing that permutation alone is not sufficient without hybrid integration. At the end, HEA brings substantial gains compared to baseline models A2SATRA, P2SATRA, S2SATRA, etc., only employing ES (HEA vs ES). This confirms that global optimization via HEA is critical to achieving state-of-the-art results. In a nutshell, it can be said that R-HEARA's improvements do not come from architecture alone, but from the coaction of all four contributions.

This article also reported the effect sizes in addition to significance testing. Exactly, a computation using Cohen's d [52] method for R-HEARA against baseline algorithms, as this method is well accepted for analyzing effect sizes. These measures confirm that R-HEARA's improvements are not only statistically significant (as shown by the Friedman test, Nemenyi post-hoc, and CD diagram) but also of substantial practical magnitude, with effect sizes ranging from medium to very large. According to Cohen (2013) [53], the effect size standard is: Small: 0.2, Medium: 0.5, Large: 0.8, Very large: > 1.3. Here, the calculation process of the effect size is represented by Eq (5.2),

$$d = \frac{(A_1 - A_2)}{SD}, \tag{5.2}$$

where, A_1 and A_2 are the mean accuracies for two algorithms across datasets and $SD = \sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$. Here, s_1, s_2 are the STDs of the two algorithms, and n_1, n_2 are the number of datasets. For example, R-HEARA vs. L2SATRA yielded Cohen's $d = 2.98$ (very large), indicating strong dominance. By combining the CD diagram (significance) with the report of the effect size (magnitude), this experimentation concluded a rigorous and comprehensive statistical validation of the performance advantage of R-HEARA. Here, Table 11 represented Cohen's d analysis of the baseline models.

Table 11. Comparison table of Cohen's d analysis of all logic mining models.

Comparison (vs R-HEARA)	Cohen's d	Effect size level	Interpretation
A2SATRA	0.95	Large	Substantial improvement
L2SATRA	2.98	Very Large	R-HEARA dominates strongly
E2SATRA	3.10	Very Large	R-HEARA dominates strongly
S2SATRA	0.79	Medium–Large	Consistent, meaningful gain
P2SATRA	0.81	Large	Robust advantage
2SATRA	2.67	Very Large	R-HEARA outperforms
3SATRA	2.22	Very Large	R-HEARA outperforms

5.6. *Overall discussions and impact analysis*

From the above discussions, it is observed that all the existing models utilize the systematic satisfiability combination, such as 2SAT or 3SAT. From each dataset experimentation, it is clearly visible that a systematic and bounded logical structure cannot provide the desired outcome. In satisfiability representation, the logical flexibility can purely enhance the performance of a logic mining model. The different and unique combinations of this higher-order non-systematic RAN*k*SAT also outperform systematic combinations for the simulated data. Hence, in the case of real datasets (D1–D20), the RAN*k*SAT logic combination, that is, R-HEARA, performs better than the other logic mining models.

Furthermore, the effectiveness of R-HEARA can be classified into three folds. In the beginning, the proposed model selected proper attributes through a well-known statistical method. In the second step, it gained multiple numbers of unique $P^{best}_{R-HEARA}$, which at the end generates a high-quality solution (final neuron states). Finally, the robustness of the proposed model was examined with different types of existing logic mining models by using several performance metrics. In this regard, the logic mining model required to add of another layer or preprocessing phases, which proves very beneficial in retrieving optimal P^{best}.

Notably, this preprocessing phase is set up before the training phase of DHNN. For this importance, the proposed R-HEARA model successfully implied a statistical method named 'log-linear' in the preprocessing phase. This log-linear model makes the association and interaction patterns among the categorical data. It needs to be mentioned that this log-linear concept works in two major different instincts. First, the K-way effect defines which attribute can be removed from the dataset. Second, the partial association defines the significant effect of the attributes. With these characteristics, log-linear achieved the best attributes which lead five (05) unique optimal $P^{best}_{R-HEARA}$ before entering the training phase. It is noteworthy that the proposed R-HEARA leverages the permutation approach from the previous P2SATRA concept and the log-linear technique from the A2SATRA concept to achieve optimal logical performance. This integration of the permutation operator and log-linear analysis allows R-HEARA to achieve high-quality induced logic, as evidenced by its performance metrics.

More importantly, the proposed R-HEARA model can generate better-induced logic which reflects in the performance of the training phase and successfully interprets the real-life datasets to detect the factors that are more prominent than other existing models. Additionally, the reason for HEA being able to fulfill such a factor is due to the optimization operators (advertisement campaign) in it. These dynamic operators explore a larger search space and variations of solutions that contribute to an effective training stage. The primary exploitative and exploratory operators of HEA in the hybrid RA select meaningful and optimal attributes to achieve the highest classification accuracy. The results of these analyses indicate that R-HEARA demonstrates a remarkable ability to identify significant and optimal features.

While R-HEARA demonstrates overall superiority across most datasets and evaluation metrics, it is noticeable that in a few cases, certain baselines such as P2SATRA and S2SATRA, we achieve competitive or slightly better results. These outcomes likely stem from dataset-specific characteristics that align with the design biases of the baseline models. For example, P2SATRA and S2SATRA emphasize pairwise satisfiability rules, which are particularly effective at capturing local attribute dependencies. In datasets where such local dependencies dominate, these models can occasionally yield higher accuracy despite their limited generalizability. Furthermore, P2SATRA and S2SATRA place stronger emphasis on the frequency of true positives for the most influential logic rules, which may improve their advantage on datasets where attributes align closely in this regard. Another plausible factor is that the attributes selected by R-HEARA's log-linear preprocessing may exhibit slightly weaker correlation strength in these specific datasets. Finally, the larger number permutation used in P2SATRA and S2SATRA can sometimes enhance their accuracy, albeit at the cost of efficiency. Overall, these cases highlight situations where baseline models can exploit narrow structural patterns, whereas R-HEARA maintains broader adaptability and superior performance across diverse datasets.

Beyond benchmark datasets, the design of R-HEARA makes it adaptable to a wide range of real-world applications. Its logical rule integration and hybrid population-based optimization could, for instance, be extended to support structured prediction in computer vision tasks such as semantic segmentation, where spatial consistency is critical. Similarly, the systematic attribute selection and rule-based modeling approach can be leveraged in financial decision-making systems, such as stock selection, or adapted to dynamic optimization settings like time-varying quadratic programming, and also for traffic forecasting with a self-supervised approach [45, 54, 55]. While these directions are beyond the scope of the present study, these works represent promising opportunities for future research and highlight the broader potential of the proposed framework.

6. Conclusions and future direction

This study introduced an explainable hybrid R-HEARA logic mining model grounded in the reverse analysis paradigm, designed to capture the underlying structure of datasets with greater precision. By incorporating a dedicated preprocessing stage with log-linear attribute selection and contingency table construction, the model is able to represent complex relationships more effectively. The integration of the permutation operator and the HEA training mechanism further enhanced the solution search process and classification performance, leading to consistent gains across diverse datasets. Experimental evaluation on twenty benchmark datasets confirmed that R-HEARA outperforms a range of established models, achieving superior accuracy, specificity, MCC, and precision. Furthermore, the ablation study of R-HEARA clearly demonstrates the individual contributions of each component, highlighting how the proposed hybridization is not only beneficial but also essential for effective optimization and knowledge extraction. Additionally, the combination of pairwise statistical significance testing and effect size analysis using Cohen's d provides strong evidence of the robustness and superiority of the proposed explainable logic mining model.

At the same time, some limitations need to be acknowledged. The attribute selection step relied solely on the log-linear method, leaving open the question of how alternative approaches, such as correlation analysis or Jaccard similarity, might affect performance. Moreover, the present work focused exclusively on non-systematic logic within DHNN, without extending the framework to other neural architectures such as memristive inertial networks, radial basis function networks, or fuzzy logic systems, which may require additional optimization layers.

Future research can address these limitations by exploring diverse attribute selection techniques such as Jaccard analysis, incorporating systematic logic representations, and extending the model to other neural paradigms [56–58]. Such directions would not only strengthen the theoretical foundation of logic mining but also broaden the applicability of R-HEARA in complex real-world domains.

Author contributions

Syed Anayet Karim: Contributed to the writing and preparation of the original draft; Mohd Shareduwan Mohd Kasihmuddin: Performed the formal analysis; Sowmitra Das: Conducted the writing review; Nur Ezlin Zamri: Developed the methodology; Akib Jayed Islam: Involved in validation and visualization; Alyaa Alway: Responsible for the conceptualization and provided resources; Deepak Kumar Chowdhury: managed project administration. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

All of the authors acknowledged Universiti Putra Malaysia for the given IPM Putra Grant with Project Code: GP-IPM/2024/9806600.

Conflict of interest

The authors declare that there are no conflicts of interest related to this work.

References

1. A. Mansour, F. Harahsheh, K. W. Wazani, M. khasawneh, B. B. AlTaher, The influence of social media, big data, and data mining on the evolution of organizational behavior: Empirical study in jordanian telecommunication sector, *Int. J. Data Netw. Sci.*, **8** (2024), 1929–1940. https://dx.doi.org/10.5267/j.ijdns.2024.1.020
2. K. Dhanushkodi, A. Bala, N. Kodipyaka, V. Shreyas, Customer behaviour analysis and predictive modelling in supermarket retail: A comprehensive data mining approach, *IEEE Access*, **13** (2024), 2945–2957. https://dx.doi.org/10.1109/ACCESS.2024.3407151
3. Q. Ge, X. Lu, R. Jiang, Y. Zhang, X. Zhuang, Data mining and machine learning in HIV infection risk research: An overview and recommendations, *Artif. Intell. Med.*, **153** (2024), 102887. https://dx.doi.org/10.1016/j.artmed.2024.102887
4. W. Bao, Y. Cao, Y. Yang, H. Che, J. Huang, S. Wen, Data-driven stock forecasting models based on neural networks: A review, *Inf. Fusion*, **113** (2025), 102616. https://dx.doi.org/10.1016/j.inffus.2024.102616
5. V. K. Gugulothu, S. Balaji, Retraction Note: An early prediction and classification of lung nodule diagnosis on CT images based on hybrid deep learning techniques, *Multimed. Tools Appl.*, **83** (2024), 88555. https://dx.doi.org/10.1007/s11042-024-20019-y
6. X. Shu, Y. Ye, Knowledge discovery: Methods from data mining and machine learning, *Soc. Sci. Res.*, **110** (2023), 102817. https://dx.doi.org/10.1016/j.ssresearch.2022.102817
7. B. Guan, D. Wang, D. Shu, S. Zhu, X. Ji, B. Sun, Data-driven casting defect prediction model for sand casting based on random forest classification algorithm, *China Foundry*, **21** (2024), 137–146. https://dx.doi.org/10.1007/s41230-024-3090-1
8. E. I. Elsedimy, S. M. M. AboHashish, F. Algarni, New cardiovascular disease prediction approach using support vector machine and quantum-behaved particle swarm optimization, *Multimed. Tools Appl.*, **83** (2024), 23901–23928. https://dx.doi.org/10.1007/s11042-023-16194-z
9. M. A. Bülbül, Optimization of artificial neural network structure and hyperparameters in hybrid model by genetic algorithm: IOS–android application for breast cancer diagnosis/prediction, *J. Supercomput.*, **80** (2024), 4533–4553. https://dx.doi.org/10.1007/s11227-023-05635-z
10. T. Bezdan, M. Zivkovic, N. Bacanin, I. Strumberger, E. Tuba, M. Tuba, Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm, *J. Intell. Fuzzy Syst.*, **42** (2021), 411–423. https://dx.doi.org/10.3233/JIFS-219200
11. J. J. Hopfield, D. W. Tank, "Neural" computation of decisions in optimization problems, *Biol. Cybern.*, **52** (1985), 141–152. https://dx.doi.org/10.1007/BF00339943
12. 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. Comput. Inf. Sci.*, **35** (2023), 101554. https://dx.doi.org/10.1016/j.jksuci.2023.101554
13. C. Wang, Y. Li, Q. Deng, Discrete-time fractional-order local active memristor-based Hopfield neural network and its FPGA implementation, *Chaos Solitons Fract.*, **193** (2025), 116053. https://dx.doi.org/10.1016/j.chaos.2025.116053

14. W. A. T. W. Abdullah, Logic programming on a neural network, *Int. J. Intell. Syst.*, **7** (1992), 513–519. https://dx.doi.org/10.1002/int.4550070604

15. A. Alway, N. E. Zamri, S. A. Karim, M. A. Mansor, M. S. M. Kasihmuddin, M. M. Bazuhair, Major 2 satisfiability logic in discrete Hopfield neural network, *Int. J. Comput. Math.*, **99** (2022), 924–948. https://dx.doi.org/10.1080/00207160.2021.1939870

16. G. Manoharam, A. M. Kassim, S. Abdeen, M. S. M. Kasihmuddin, N. A. Rusdi, N. A. Romli, et al., Special major 1, 3 satisfiability logic in discrete Hopfield neural networks, *AIMS Mathematics*, **9** (2024), 12090–12127. https://dx.doi.org/10.3934/math.2024591

17. S. A. Karim, N. E. Zamri, A. Alway, M. S. M. Kasihmuddin, A. I. M. Ismail, M. A. Mansor, et al., Random satisfiability: A higher-order logical approach in discrete Hopfield neural network, *IEEE Access*, **9** (2021) 50831–50845. https://dx.doi.org/10.1109/ACCESS.2021.3068998

18. X. Jiang, M. S. M. Kasihmuddin, Y. Guo, Y. Gao, M. A. Mansor, N. E. Zamri, et al., J-type random 2, 3 satisfiability: A higher-order logical rule in discrete Hopfield neural network, *Evol. Intel.*, **17** (2024), 3317–3336. https://dx.doi.org/10.1007/s12065-024-00936-5

19. Y. Gao, M. S. M. Kasihmuddin, J. Chen, C. Zheng, N. A. Romli, M. A. Mansor, et al., Binary ant colony optimization algorithm in learning random satisfiability logic for discrete Hopfield neural network, *Appl. Soft Comput.*, **166** (2024), 112192. https://dx.doi.org/10.1016/j.asoc.2024.112192

20. A. Alway, N. E. Zamri, M. 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, *Decis. Anal. J.*, **9** (2023), 100354. https://dx.doi.org/10.1016/j.dajour.2023.100354

21. N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, S. S. Sidik, A. Alway, N. A. Romli, et al., 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://dx.doi.org/10.1016/j.eswa.2023.122307

22. S. Sathasivam, W. A. T. W. Abdullah, Logic mining in neural network: Reverse analysis method, *Computing*, **91** (2011), 119–133. https://dx.doi.org/10.1007/s00607-010-0117-9

23. S. Z. M. Jamaludin, N. S. Sa'ari, M. S. M. Kasihmuddin, M. F. Marsani, N. E. Zamri, S. A. Azhar, et al., Artificial bee colony for logic mining in credit scoring, *Malays. J. Fundam. Appl. Sci.*, **18** (2022), 654–673. https://dx.doi.org/10.11113/mjfas.v18n6.2661

24. 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, *Comput. Mater. Contin.*, **74** (2023), 2853–2870. https://dx.doi.org/10.32604/cmc.2023.032654

25. 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://dx.doi.org/10.3390/math10060915

26. 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. Comput. Inf. Sci.*, **34** (2022), 9011–9027. https://dx.doi.org/10.1016/j.jksuci.2022.08.026

27. J. Ma, The stability of the generalized Hopfield networks in randomly asynchronous mode, *Neural Netw.*, **10** (1997), 1109–1116. https://dx.doi.org/10.1016/S0893-6080(97)00026-9

28. S. Lopez-Ruiz, C. I. Hernández-Castellanos, K. Rodriguez-Vazquez, Multi-objective optimization of neural network with stochastic directed search, *Expert Syst. Appl.*, **237** (2024), 121535. https://dx.doi.org/10.1016/j.eswa.2023.121535

29. S. A. Karim, M. S. M. Kasihmuddin, S. Sathasivam, M. A. Mansor, S. Z. M. Jamaludin, M. R. Amin, A novel multi-objective hybrid election algorithm for higher-order random satisfiability in discrete Hopfield neural network, *Mathematics*, **10** (2022), 1963. https://dx.doi.org/10.3390/math10121963

30. S. M. E. Saryazdi, A. Etemad, A. Shafaat, A. M. Bahman, Data-driven performance analysis of a residential building applying artificial neural network (ANN) and multi-objective genetic algorithm (GA), *Building and Environment*, **225** (2022), 109633. https://dx.doi.org/10.1016/j.buildenv.2022.109633

31. O. Ramos-Figueroa, M. Quiroz-Castellanos, E. Mezura-Montes, R. Kharel, Variation operators for grouping genetic algorithms: A review, *Swarm Evol. Comput.*, **60** (2021), 100796. https://dx.doi.org/10.1016/j.swevo.2020.100796

32. V. V. Starovoitov, Yu. I. Golub, Comparative study of quality estimation of binary classification, *Informatics*, **17** (2020), 87–101. https://dx.doi.org/10.37661/1816-0301-2020-17-1-87-101

33. J. Dou, Y. Song, G. Wei, Y. Zhang, Fuzzy information decomposition incorporated and weighted Relief-F feature selection: When imbalanced data meet incompletion, *Inform. Sci.*, **584** (2022), 417–432. https://dx.doi.org/10.1016/j.ins.2021.10.057

34. K. Jha, S. Saha, Incorporation of multimodal multiobjective optimization in designing a filter-based feature selection technique, *Appl. Soft Comput.*, **98** (2021), 106823. https://dx.doi.org/10.1016/j.asoc.2020.106823

35. I. Maruotto, F. K. Ciliberti, P. Gargiulo, M. Recenti, Feature selection in healthcare datasets: Towards a generalizable solution, *Comput. Biol. Med.*, **196** (2025), 110812. https://dx.doi.org/10.1016/j.compbiomed.2025.110812

36. M. Hossin, M. N. Sulaiman, A review on evaluation metrics for data classification evaluations, *Int. J. Data Min. Knowl. Manag. Process*, **5** (2015), 1. https://dx.doi.org/10.5121/ijdkp.2015.5201

37. J. B. Brown, Classifiers and their metrics quantified, *Mol. Inform.*, **37** (2018), 1700127. https://dx.doi.org/10.1002/minf.201700127

38. D. Brzezinski, J. Stefanowski, R. Susmaga, I. Szczech, On the dynamics of classification measures for imbalanced and streaming data, *IEEE Trans. Neural Netw. Learn. Syst.*, **31** (2019), 2868–2878. https://dx.doi.org/10.1109/TNNLS.2019.2899061

39. Q. Zhu, On the performance of Matthews correlation coefficient (MCC) for imbalanced dataset, *Pattern Recognit. Lett.*, **136** (2020), 71–80. https://dx.doi.org/10.1016/j.patrec.2020.03.030

40. N. Singh, P. Singh, A hybrid ensemble-filter wrapper feature selection approach for medical data classification, *Chemom. Intell. Lab. Syst.*, **217** (2021), 104396. https://dx.doi.org/10.1016/j.chemolab.2021.104396

41. A. Luque, A. Carrasco, A. Martín, A. de las Heras, The impact of class imbalance in classification performance metrics based on the binary confusion matrix, *Pattern Recogn.*, **91** (2019), 216–231. https://dx.doi.org/10.1016/j.patcog.2019.02.023

42. S. Z. M. Jamaludin, M. T. Ismail, M. S. M. Kasihmuddin, M. A. Mansor, S. N. F. M. A. Antony, A. A. Makhul, Modelling benign ovarian cyst risk factors and symptoms via log-linear model, *Pertanika J. Sci. Technol.*, **29** (2021), 2199–2216. https://dx.doi.org/10.47836/pjst.29.3.26

43. 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://dx.doi.org/10.3390/e23010040

44. N. E. Zamri, M. A. Mansor, M. S. M. 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://dx.doi.org/10.3390/e22060596

45. Q. Lai, P. Chen, Unveiling node relationships for traffic forecasting: A self-supervised approach with MixGT, *Inf. Fusion*, **120** (2025), 103070. https://dx.doi.org/10.1016/j.inffus.2025.103070

46. A. Tharwat, Classification assessment methods, *Appl. Comput. Inform.*, **17** (2021), 168–192. https://dx.doi.org/10.1016/j.aci.2018.08.003

47. M. Dorrich, M. Fan, A. M. Kist, Impact of mixed precision techniques on training and inference efficiency of deep neural networks, *IEEE Access*, **11** (2023), 57627–57634. https://dx.doi.org/10.1109/ACCESS.2023.3284388

48. P. Stoica, P. Babu, Pearson–Matthews correlation coefficients for binary and multinary classification, *Signal Process.*, **222** (2024), 109511. https://dx.doi.org/10.1016/j.sigpro.2024.109511

49. D. Chicco, G. Jurman, A statistical comparison between Matthews correlation coefficient (MCC), prevalence threshold, and Fowlkes–Mallows index, *J. Biomed. Inform.*, **144** (2023), 104426. https://dx.doi.org/10.1016/j.jbi.2023.104426

50. P. Thölke, Y. J. Mantilla-Ramos, H. Abdelhedi, C. Maschke, A. Dehgan, Y. Harel, et al., Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data, *NeuroImage*, **277** (2023), 120253. https://dx.doi.org/10.1016/j.neuroimage.2023.120253

51. J. Demšar, Statistical comparisons of classifiers over multiple datasets, *J. Mach. Learn. Res.*, **7** (2006), 1–30.

52. M. Georgiou, G. Morison, N. Smith, Z. Tieges, S. Chastin, Mechanisms of impact of blue spaces on human health: A systematic literature review and meta-analysis, *Int. J. Environ. Res. Public Health*, **18** (2021), 2486. https://dx.doi.org/10.3390/ijerph18052486

53. J. Cohen, *Statistical power analysis for the behavioral sciences*, 2 Eds., New York: Routledge, 2013. https://dx.doi.org/10.4324/9780203771587

54. M. Ma, S. Chen, L. Zheng, Novel adaptive parameter fractional-order gradient descent learning for stock selection decision support systems, *Eur. J. Oper. Res.*, **324** (2025), 276–289. https://dx.doi.org/10.1016/j.ejor.2025.01.013

55. M. Ma, L. Zheng, J. Yang, A novel improved trigonometric neural network algorithm for solving price-dividend functions of continuous-time one-dimensional asset-pricing models, *Neurocomputing*, **435** (2021), 151–161. https://dx.doi.org/10.1016/j.neucom.2021.01.012

56. Y. Jiang, S. Zhu, S. Wen, C. Mu, Reachable set estimation of memristive inertial neural networks, *IEEE Trans. Circuits Syst. II*, **72** (2025), 903–907. https://dx.doi.org/10.1109/TCSII.2025.3572855

57. N. A. Rusdi, N. E. Zamri, M. S. M. Kasihmuddin, N. A. Romli, G. Manoharam, S. Abdeen, et al., Synergizing intelligence and knowledge discovery: Hybrid black hole algorithm for optimizing discrete Hopfield neural network with negative-based systematic satisfiability, *AIMS Mathematics*, **9** (2024), 29820–29882. https://dx.doi.org/10.3934/math.20241444

58. N. A. Romli, N. F. S. Zulkepli, M. S. M. Kasihmuddin, S. A. Karim, S. Z. M. Jamaludin, N. A. Rusdi, et al., An optimized logic mining method for data processing through higher-order satisfiability representation in discrete Hopfield neural network, *Appl. Soft Comput.*, **184** (2025), 113759. https://dx.doi.org/10.1016/j.asoc.2025.113759

AIMS Press