AIMS Mathematics

AIMS Mathematics, 11(6): 18081–18121.
DOI: 10.3934/math.2026736
Received: 06 February 2026
Revised: 16 March 2026
Accepted: 20 March 2026
Published: 18 June 2026

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

Research article

A major random 1,3-satisfiability logic mining model in discrete Hopfield neural networks for adverse event analysis

Gaeithry Manoharam[1], Nur 'Afifah Rusdi[2], Nurshazneem Roslan[2], Nurul Atiqah Romli[3], Mohd Shareduwan Mohd Kasihmuddin[1,*], Nur Ezlin Zamri[4], Suad Abdeen[1], Mohd. Asyraf Mansor[5] and Xiaoyan Liu[1,6]

[1] School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800 USM, Malaysia
[2] Department of Mathematical Sciences, Faculty of Intelligent Computing, Universiti Malaysia Perlis, Kampus Alam UniMAP, Pauh Putra, 02600 Arau, Perlis, Malaysia
[3] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM) Cawangan Perlis, Arau, 02600 Perlis, Malaysia
[4] Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia
[5] School of Distance Education, Universiti Sains Malaysia, Penang 11800 USM, Malaysia
[6] School of Integrated Circuits and New Energy, Guangzhou College of Technology and Business, 510850 Guangzhou, China

***Correspondence:** Email: shareduwan@usm.my.

Abstract: Logic mining in neural networks enables the extracting of interpretable symbolic knowledge from complex datasets. This study proposes a logic mining approach in discrete Hopfield neural networks based on a non-systematic major random 1,3-satisfiability (MR1,3SAT) logical structure for adverse event analysis in Alzheimer's disease datasets. The proposed model incorporates first- and third-order logical clauses to enhance representational flexibility while maintaining computational tractability. A feature selection mechanism based on Jaccard analysis is incorporated to identify relevant attributes, while the intelligent artificial bee colony algorithm is applied during the retrieval phase to refine the final neuron states. Experimental evaluation on Alzheimer's disease neuroimaging initiative datasets demonstrate that the proposed model achieves superior performance compared with existing logic mining approaches, obtaining 90.23% accuracy, 98% sensitivity, 99% specificity, and a Fowlkes-Mallows index of 93.46%. These results indicate that the MR1,3SAT-based logic mining model improves both interpretability and classification reliability for adverse event symptoms analysis in Alzheimer's disease neuroimaging initiative datasets.

Keywords: Alzheimer disease; discrete Hopfield neural network; logic mining; depression detection; adverse event and Fowlkes-Mallows
Mathematics Subject Classification: 68N17, 68R07, 68T27

1. Introduction

Alzheimer's disease (AD) is the leading cause of dementia worldwide, accounting for more than 70% of cases associated with emotional and behavioral complications such as depression [1]. First described by Alois Alzheimer in 1907 [2], AD is a progressive neurodegenerative condition that disrupts cognitive functions including memory, reasoning, and communication. Despite extensive global research and substantial advancements in medical science, no curative treatment currently exists for AD and available therapies remain limited in their ability to prevent further progression or reverse disease progression. Early identification is especially important because the pathological changes of AD begin long before clinical symptoms surface. The disease quietly damages neural structures for years before impairments such as memory decline, language difficulties, and reduced problem-solving abilities become noticeable [3]. As the condition worsens, affected individuals gradually lose their capacity to manage daily activities, make decisions, and maintain personal independence. These limitations often persist for many years, placing considerable emotional and practical burdens on patients and caregivers alike [4]. Among the adverse effects of AD, depression is a critical but often overlooked symptom commonly triggered by diagnosis shock, reduced independence, and social isolation, all of which undermine personal identity and autonomy [5–7]. Therefore, intelligent systems capable of identifying symptom patterns linked to such emotional disturbances are essential for enabling timely interventions and improving clinical decision-making [8].

In recent years, advancements in artificial intelligence (AI) have enabled the design of tools capable of solving large-scale optimization problems with minimal human intervention [9]. Among these, artificial neural networks (ANN) have been widely applied due to their ability to model nonlinear relationships by simulating the behavior of interconnected biological neurons [10]. These results illustrate the optimized solution for the task. An ANN structure is determined by the way neurons are organized and how data is transmitted through the network. In optimization tasks, feedback-based ANN models are often preferred because their interconnected neuron structures can capture complex relationships between inputs and outputs. Among these models, the discrete Hopfield neural network (DHNN) has gained prominence as an effective optimization tool. Originally introduced by Hopfield and Tank [11] to address the traveling salesman problem (TSP), DHNN employs the Lyapunov energy function to guide the system toward stable solutions. The network stores synaptic weights in content addressable memory (CAM), enabling the retrieval of complete patterns from partial or corrupted inputs [12]. These properties have facilitated the use of DHNN in diverse optimization problems including transportation routing, medical image encryption and the analysis of electromagnetic radiation [13]. However, DHNN faces challenges related to transparency. It operates as a black-box model, which is the internal process that influences synaptic weight changes or energy shifts and which remains unclear [14]. Recent advances in explainable artificial intelligence (XAI) have emphasized the importance of developing models that provide transparent and interpretable decision-making processes. One promising direction is neuro-symbolic learning, which integrates neural network learning capabilities with symbolic reasoning mechanisms [15].

This hybrid paradigm combines the pattern recognition strength of neural models with the interpretability and reasoning capability of symbolic systems, enabling AI models to produce human-understandable explanations while maintaining strong predictive performance [16]. Neuro-symbolic approaches have recently gained increasing attention in domains requiring trustworthy and interpretable AI, such as healthcare, cybersecurity, and scientific discovery. By embedding logical structures within the neural structure, these models enable the extraction of explicit rules and semantic relationships from complex datasets, thereby improving transparency and interpretability compared with conventional black-box neural networks [17]. In other words, in the model, it is difficult to determine exactly how the network arrives at its final output. Given that most ANN models exhibit linearity, understanding how neuron states evolve from initial to final stages is essential. To improve interpretability, an optimal logic representation in DHNN is needed [18]. This would allow the network decision-making process to be explained through logical rules, providing a clearer understanding of how input and output neurons are connected.

Abdullah [19] proposed a symbolic ANN, where logic was implemented directly as neurons in the DHNN. Here, the neurons act as variables of logical rules and operate in bipolar states. A cost function was formulated to minimize inconsistencies in the logical structure, ensuring alignment with the intended network logic. This cost function could be determined by comparing optimal synaptic weights to the network final energy state. This transformation showed that the DHNN no longer relied solely on bipolar patterns, but instead adopted logic-driven patterns influenced by environmental factors. This development helped create different types of logic-based DHNN models [20]. The systematic logical structure of the DHNN was strengthened when Kasihmuddin et al. [21] implemented 2-satisfiability (2SAT) logic in which each clause entailed two neurons. This method improved the network by embedding a well-defined logical structure and resulted in a higher global minima ratio with faster convergence. Building on this, Mansor et al. [22] introduced 3-satisfiability (3SAT) logic into the DHNN learning phase. Their work demonstrated that increasing storage capacity minimized the occurrence of local solutions thereby improving network performance. However, both 2SAT and 3SAT introduced limitations by restricting the number of literals per clause. To overcome these limitations, Sathasivam et al. [23] introduced the random 2-satisfiability (RAN2SAT) model incorporated both first- and second-order logic into the DHNN model. The simulation outcomes demonstrated that RAN2SAT consistently attained optimal final neuron states across a range of datasets. Another similar no-systematic structure introduced by Karim et al. [24] known as random 3-satisfiability (RAN3SAT), extending the logic structure by including first-, second- and third-order components. Their simulation results showed that RAN3SAT was capable of achieving optimal neuron states particularly in scenarios, where the presence of first-order clauses was relatively limited. Moreover, Alway et al. [25] introduced major 2-satisfiability (MAJ2SAT), which is defined by a specific distribution of clauses in the logical formulation. In DHNN, this model emphasized second-order clauses, which enhanced neuron diversity and improved the likelihood of reaching global minima. Similarly, Jiang et al. [18] introduced the J-type random 2,3-satisfiability into the DHNN model by combining second-order and third-order clauses to reduce learning errors and minimize global energy. However, incorporating randomization with higher-order clause combinations may introduce uncertainty in the satisfied interpretations, which in turn can compromise the attainment of global solutions and limit the variability of final neuron states in the DHNN model. Additionally, this study did not investigate the first-order logic, which is very important to the DHNN model that it is able to learn lower-order logic to provide a more diversified solution. Although several logical rules have been developed, their overall influence on the effectiveness of logic mining is still not well understood. Choosing the right logical rule for DHNN has the potential to improve how well meaningful patterns are uncovered in data, leading to clearer and more accurate interpretations.

Logic mining refers to the process of extracting meaningful knowledge from data by formulating it as logical rules. This approach supports symbolic representation, making the extracted information more transparent and easier to interpret. It enhances artificial intelligence by allowing neural networks to identify the most appropriate logical structures for representing important patterns in data. The origin of the logic mining technique was established by Sathasivam and Abdullah [26], who utilized the reverse analysis (RA) technique to derive logical rules from an education dataset. Their approach was based on the Wan Abdullah (WA) method [19], which determined optimal synaptic weights among neurons. The obtained weights were utilized to generate induced logic that represents the underlying structure of the dataset. Despite achieving over 80% in support and confidence, RA often struggled to generate induced logic that was both optimal and generalizable. To solve these limitations, Kho et al. [27] introduced the 2-satisfiability-based reverse analysis (2SATRA), enhancing the earlier RA method proposed by Sathasivam and Abdullah [26]. To improve efficiency, 2SATRA introduced a logic mining method with improved steps in the preprocessing phase aiming to eliminate unnecessary learning in DHNN. Consequently, the model successfully extracted logic and identified the optimal induced logic in the League of Legends dataset. Expanding this method, Alway et al. [28] applied 2SATRA to analyze palm oil price trends in relation to other commodities. This model results demonstrated that 2SATRA could generate highly accurate induced logic by providing precise interpretations of observed price fluctuations. For more complex, higher-order logic extraction, Zamri et al. [29] proposed the 3-satisfiability-based reverse analysis (3SATRA), which was applied to Amazon employee resource access data. By leveraging the increased storage capacity of 3SAT, this method outperformed conventional data mining techniques in capturing meaningful patterns. In addition, Jamaludin et al. [30] proposed the energy-based 2-satisfiability reverse analysis (E2SATRA) model to extract logical rules from E-recruitment datasets. Unlike traditional k satisfiability reverse analysis (kSATRA) approaches, E2SATRA eliminated neuron states trapped in local minimum energy levels, thereby refining the overall logic extraction process. Despite these improvements, existing kSATRA methods still lack a systematic approach for determining the optimal combination of logical rules that lead to the most effective induced logic. Addressing this gap remains crucial for enhancing the accuracy and efficiency of logic mining in DHNN.

To overcome the limitations of earlier logic mining techniques, Jamaludin et al. [31] introduced permutation-based logic mining (P2SATRA), which embedded a permutation operator into DHNN neurons. This mechanism of logical permutations improved the accuracy of induced logic compared with the standard 2SATRA model, although its reliance on randomized variable placement within clauses restricted its capacity to capture meaningful real-world datasets. Within supervised logic mining, Kasihmuddin et al. [30] developed supervised 2-satisfiablity reverse analysis (S2SATRA), which used correlation analysis of dataset attributes to generate more precise logic rules. The exclusion of irrelevant features before logic integration allowed S2SATRA to outperform conventional logic mining methods in terms of accuracy. Building upon this, Jamaludin et al. [32] further enhanced 2SATRA by introducing a log-linear modeling approach, resulting in the 2SAT reverse analysis method (A2SATRA). Through the application of a log-linear method, A2SATRA emphasized the extraction of essential attributes in advance of logic construction, which contributed to producing higher-quality induced logic. In addition, it adopted the permutation mechanism introduced by

Jamaludin et al. [33] enabling local field optimization and leading to more effective induced logic performance. The findings provided evidence that A2SATRA surpassed the majority of current logic mining approaches. Despite these developments, the nature structured of existing logic mining may still lead to overfitting, restricting their generalizability across different type of datasets. The remaining challenge lies in incorporating non-systematic logic to identify optimal subsets of induced logic, allowing DHNN to better capture and model the complexity of dataset behavior. In response, Alway et al. [34] introduced major 2-satisfiability hybrid exhaustive search reverse analysis (2HESRA), a logic mining model featuring a hybrid exhaustive search algorithm during the learning phase. Similarly, Zamri et al. [35] proposed a non-systematic logic mining approach that utilizes a modified Niched genetic algorithm to construct a "super logic" based on weighted first- and second-order clauses. This method, known as weighted random 2-satisfiability modified reverse-based analysis (r2SATMRA) selects the optimal induced logic by maximizing the combined positive and negative outcome scores. The recent logic mining method employed a hybrid differential evolution algorithm (HDEA) for synaptic weight optimization, a swarm mutation (SM) operator to diversify neuron states, and an enhanced reverse analysis technique for generating effective training logic [36].

Optimization problems in industrial and engineering domains often involve large, nonlinear, and computationally intensive search spaces that are difficult to solve using exact optimization methods. In such situations, heuristic and metaheuristic algorithms provide an effective alternative by enabling efficient exploration of complex solution landscapes [37]. For example, multi-objective metaheuristic approaches have been applied to optimizing the number of train operations, train routing and stopping patterns in railway systems with the aim of reducing operational costs, balancing occupancy rates, minimizing travel time, and improving passenger travel satisfaction [38]. Studies have also shown that heuristic algorithms are capable of navigating large combinatorial solution spaces while maintaining reasonable computational cost, making them highly suitable for solving practical real-world optimization problems [39]. Recent studies have also explored the dynamic behaviors of neural networks using memristive systems and nonlinear circuit implementations. For instance, a universal method for enhancing neural network dynamics using memristor-based architectures has been proposed, demonstrating improved dynamical properties and potential applications in IoT-based robot navigation systems [40]. Additionally, memristive cyclic neural networks capable of generating controllable multiscroll chaotic attractors and multivariable amplitude control have been developed to study complex nonlinear dynamics and circuit implementations [41]. These studies highlight the growing interest in improving the dynamical richness and hardware realizability of neural network models. However, most existing works focus primarily on nonlinear dynamics and circuit-based implementations. In contrast, the present study emphasizes logic mining and interpretable knowledge extraction using DHNN, where logical structures such as major random 1,3-satisfiability (MR1,3SAT) are utilized to discover interpretable patterns in adverse event datasets. Nonetheless, existing models function with only a single induced logic and do not incorporate additional components during the retrieval phase, which may restrict both the interpretability and adaptability of the induced logic. To overcome this limitation, the present study introduces a new logic mining model that combines non-systematic logic, an optimal feature selection method, and a constructive strategy for generating the most effective logic. Furthermore, the model employs an improved retrieval mechanism aimed at enhancing both the diversity and interpretability of the induced logic. The main contributions of this paper can be summarized as follows:

(1) To propose a non-systematic major random 1,3-satisfiability logical structure to represent adverse event data. The proposed logical representation enables the modeling of complex logical relationships among attributes within the dataset.
(2) To incorporate major random 1,3-satisfiability into the discrete Hopfield neural network model to perform logic mining and knowledge extraction. This implementation allows the neural network to learn and optimize logical clauses while identifying meaningful patterns from the dataset.
(3) To implement an intelligent artificial bee colony (IABC) algorithm during the retrieval stage, enabling refinement of the final neuron state through a mutation processes. The mutation operator exchanges neuron states across clause orders, while spot mutation identifies unsatisfied clauses and modifies neuron states without reducing fitness.
(4) To introduce a threefold objective strategy during the retrieval phase of the discrete Hopfield neural network that aims to obtain a high quality final neuron state in terms of global solution, diversity solution, and low similarity. The final neuron state with the highest quality is interpreted as the induced logic for the Alzheimers dataset.
(5) To validate the effectiveness of the proposed logic mining method on Alzheimer's disease datasets using multiple performance metrics, comparing its outcomes with baseline model through the Friedman statistical test.

This paper is organized into the following sections: Section 2 outlines the motivations of the study. Sections 3–6 describe the background, including the logical structure and details of the proposed logic mining method. Section 7 explains the experimental setup and AD dataset. Section 8 presents the results of all models, and Section 9 discusses the main findings.

2. Motivation

Current logic mining models face key limitations: The rigid order of logic application reduces adaptability, the reliance on true positive frequency offers only partial optimization, and the absence of mechanisms to expand the solution space restricts diversity in logical structures. Overcoming these issues is crucial for enhancing the robustness and effectiveness of logic mining.

2.1. Inflexibility of systematic logic

Logical structures serve as the foundation of data representation in DHNN, functioning as symbolic rules that capture the dataset behavior during both training and retrieval. Previously, Kasihmuddin et al. [21] and Mansor et al. [22] successfully applied 2SAT and 3SAT logic in DHNN by achieving the highest global minima ratio. However, these models faced overfitting issues as the network struggled to generate diverse synaptic weights due to a fixed number of neurons assigned to each clause. So far, the most effective *k*-order SAT that encourages non-repetitive neuron patterns has not been identified mainly because current logic structures use a rigid number of literals. Since DHNN assigns equal importance to all solution spaces through uniform synaptic weights, it becomes difficult to retrieve a variety of final neuron states. To address this, study introduces major random 1,3-satisfiability a non-systematic approach that combines first- and third-order clauses. The combination of different orders logic in the major random 1,3-satisfiability enriches neuron variability and enhances retrieval performance.

2.2. Limited search space

The effectiveness of logic mining hinges on identifying the most representative logic structure of a dataset. An optimal logic serves as the foundation for producing high-quality induced logic in DHNN, supporting reliable classification tasks in AI applications. However, many existing studies [28–30] largely depend on true positive outcomes when formulating logic. This single perspective increases the risks overfitting as it fails to capture the dataset full behavior especially in cases where true negative outcomes play an important role. Moreover, the DHNN memory mechanism tends to retain positive-oriented logic, further reducing its ability to generalize. In datasets dominated by negative instances, this imbalance can lead to randomized weight distributions, making the model behave more like a random classifier. To counter this, the present study proposes a balanced evaluation strategy that incorporates both true positive and true negative measures. This dual consideration ensures that induced logic better reflects the dataset's complete structure.

2.3. Effective feature selection method

Feature selection techniques are an essential aspect in logic mining, as the absence of an effective selection mechanism can result in induced logic that lacks interpretability. Previous studies [27–29] introduced systematic models to improve induced logic, yet the retention of non-significant attributes continued to hinder knowledge extraction. To address this, Jamaludin et al. [33] proposed a log-linear approach for selecting relevant attributes, which improved feature structuring but lacked an intelligent model to capture dataset patterns. Extending this idea, Zamri et al. [35] incorporated Jaccard analysis within a non-systematic model, enabling more precise alignment between induced logic and dataset characteristics. Building on these advancements, this study employs Jaccard analysis in the preprocessing stage before DHNN training, thereby strengthening attribute relevance and improving the overall quality of induced logic.

Recent studies have shown significant progress in neural network architectures for modeling complex data and improving predictive performance. For example, Zhou et al. [42] proposed an attention-based spatio-temporal graph neural network that captures both spatial interactions and temporal motion patterns among pedestrians, enabling more accurate trajectory prediction in dynamic environments. The model leverages graph attention mechanisms to represent interactions between agents and extract meaningful temporal dependencies. Such developments demonstrate the increasing importance of advanced neural architectures and attention mechanisms for modeling complex relationships in data. In addition, recent research has emphasized the importance of effective feature selection and representation learning in machine learning models. Fan et al. [43] introduced an adaptive regularization approach to improve multiclass feature selection by identifying informative attributes while suppressing redundant features. Similarly, Li et al. [44] demonstrated how convolutional neural networks can effectively learn hierarchical representations for complex visual segmentation tasks. Although these deep learning approaches achieve strong predictive performance, many remain difficult to interpret due to their black-box nature. In contrast, the proposed MR1,3SAT-based logic mining approach within a DHNN focuses on generating interpretable logical rules while maintaining effective pattern recognition capability.

2.4. Increase the variation of solution space

Beyond feature selection and evaluation metrics, the retrieval phase strongly influences the reliability and clarity of the induced logic. In particular, the permutation operator has shown great promise in widening the solution space by dynamically rearranging variables within logical clauses. For instance, comparisons between 2SATRA [27] and P2SATRA [31] revealed that fixed attribute arrangements often restricted interaction across clauses, heightening the risk of overfitting. Conversely, the permutation operator in P2SATRA introduced flexibility by generating new attribute combinations, which revealed more diverse neuron states and improved induced logic. This approach has since been extended by several works [32,45], underscoring its effectiveness in overcoming rigid SAT formulations. Building on these insights, this study implements permutation operators within more flexible logic structures, aiming to further expand the search space and enhance the adaptability of induced logic in DHNN.

3. Major random 1,3-satisfiability representation

Major random k satisfiability (MRkSAT) is a non-systematic logical rule that consists of first and major third-order clauses in the logical structure. The concept is inspired by the major satisfiability logic introduced by Alway et al. [25]. In this study, the focus is on MRkSAT logic for k=1 and k=3 (MR1,3SAT), which serves as a symbolic representation of the dataset. The proposed MR1,3SAT model adopts a non-systematic logical structure consisting of first-order and third-order clauses. First-order clauses represent the individual contribution of each attribute that enabling the model to capture the direct influence of individual symptoms. Meanwhile, third-order clauses allow the representation of higher-order interactions among multiple attributes, which is particularly useful for modeling complex symptom relationships in medical datasets. The exclusion of second-order clauses is intended to balance model complexity and interpretability, as incorporating all clause orders would significantly increase the number of possible clause combinations and computational cost. By combining first-order and third-order clauses, the MR1,3SAT model effectively captures both individual attribute effects and multi-variable interactions while maintaining computational efficiency within the DHNN structure. This non-systematic higher-order formulation ensures satisfiability across different SAT combinations, each possessing unique satisfiable properties. The general structure of MR1,3SAT is expressed in Eq (3.1) [46]:

$$T_{MR1,3SAT} = \wedge_{i=1}^{NC} C_i^{(k)}, \text{ where } k = 1,3, \tag{3.1}$$

whereby $C_i^{(k)}$ denotes the i^{th} clause, consisting of k variables, formulated as shown in

$$C_i^{(k)} = \begin{cases} P_i \vee Q_i \vee R_i, & k = 3, \\ S_i, & k = 1. \end{cases} \tag{3.2}$$

In Eq (3.1), NC signifies the total number of k variable in $T_{MR1,3SAT}$ as described by the formulation defined in

$$NC = \sum_{k=1}^{k}\left[n\left(C^{(k)}\right)\right]. \tag{3.3}$$

Accordingly, the fundamental properties of MR1,3SAT can be outlined to highlight its structural formulation and operational characteristics [38]. These properties summarized as below:

(a) A set of z literals $l_1, l_2, l_3, \ldots, l_z$, where $l_i \in \{-1,1\}$ exists in a bipolar state representing True or False outcomes, respectively.

(b) A literal refers to a variable(l_i) or its negated form $(\neg l_i)$.

(c) A collection of n definite clauses, $C_1, C_2, C_3, \ldots, C_n$, where no clause contains redundant variables.

(d) Each C_i is connected with the AND $(\wedge)$ logical operator, while the literals within a clause are connected using the logical OR $(\vee)$ operator.

In accordance with the established properties of MR1,3SAT, an illustrative example of $T_{MR1,3SAT}$ is given in

$$T_{MR1,3SAT} = (P_1 \vee \neg Q_1 \vee \neg R_1) \wedge (P_2 \vee Q_2 \vee R_2) \wedge (P_3 \vee \neg Q_3 \vee R_3) \wedge S_1. \tag{3.4}$$

It is important to note that in the outcome of $T_{MR1,3SAT}$, a clause can only take two possible states: satisfied or unsatisfied. The desired outcome of any $T_{MR1,3SAT}$ is to find the satisfiable interpretation $(T_{MR1,3SAT} = 1)$. Conversely, the expression becomes unsatisfiable if the condition $T_{RAN3SAT} = -1$. The outcome of a clause can either be satisfied or be unsatisfied, and this determination largely depends on the state of the variables, which plays a decisive role in establishing whether the clause is satisfiable for $T_{MR1,3SAT}$ [25]. Referring to Eq (3.4), $T_{MR1,3SAT} = 1$ when $(P_1, Q_1, R_1, P_2, Q_2, R_2, P_3, Q_3, R_3, S_1) = (-1,1,-1,1,1,1,-1,1,1,1)$. However, $T_{MR1,3SAT} = -1$ when the neuron state $(P_1, Q_1, R_1, P_2, Q_2, R_2, P_3, Q_3, R_3, S_1) = (-1,1,-1,1,1,1,-1,1,1,-1)$. Hence, the interconnections among neurons in the DHNN can be formally represented by $T_{MR1,3SAT}$.

4. Major 1,3 random satisfiability in discrete Hopfield neural network

The DHNN is a recurrent model of fully connected neurons without hidden layers. Its learning process comprises training, where satisfied interpretations $T_{MR1,3SAT}$ guide synaptic weight formation, and retrieval, where the stored weights in content addressable memory (CAM) determine the final neuron states. The general neuron activation is expressed in

$$S_i = \begin{cases} 1, & \text{if } \sum_{v}^{N} W_{ij} S_j \geq \tau, \\ -1, & \text{otherwise.} \end{cases} \tag{4.1}$$

In the DHNN model, the synaptic weight between units *i* to *j* is expressed as W_{ij}, where the state of neuron *i* represented as S_i directly influences the excitation of neuron j, contributing to the overall energy minimization process. The threshold τ defines the condition for neuron activation. In DHNN, the weights W_{ij} are symmetric $W_{ij} = W_{ji}$ and there is no self-connection ($W_{ii} = W_{jj} = 0$). The arrangement of these weights critically influences the performance of the content addressable memory (CAM), which functions as a memory system capable of retrieving stored patterns based on neuronal connectivity. These properties highlight DHNN as a reliable model for information storage. In this context, the MR1,3SAT logical rule enhances the representation of neuron interactions through the symbolic formulation $T_{MR1,3SAT}$ in the DHNN-MR1,3SAT model by assigning a neuron to each variable defined in Eq (3.1). Each neuron represents the binary states of False (-1) and True (1) as described by Mansor et al. [22]. The evolution of these neurons is governed by minimizing the cost function that correlated with $T_{MR1,3SAT}$. Equation (4.2) shows the formulation of cost function:

$$\mathbb{C}_{T_{\text{MR1,3SAT}}} = \frac{1}{8}\sum_{i=1}^{x}\prod_{j=1}^{3} S_{ij} + \frac{1}{4}\sum_{i=1}^{y}\prod_{j=1}^{2} S_{ij} + \frac{1}{2}\sum_{i=1}^{z}\prod_{j=1}^{1} S_{ij}. \tag{4.2}$$

S_{ij} is the inconsistency of $T_{MR1,3SAT}$ and is defined as

$$I_{ij} = \begin{cases} \left(1 + S_{A_i}\right), & \text{if } \mathrm{A}_i, \\ \left(1 - S_{A_i}\right), & \text{if } \neg A_{i.} \end{cases} \tag{4.3}$$

The primary objective of DHNN is to minimize $\mathbb{C}_{T_{MR1,3SAT}}$, ensuring that the derived synaptic weights are accurate. In this context, $\mathbb{C}_{T_{MR1,3SAT}} = 0$ corresponds to the experimental setup, where all clauses are fully satisfied, whereas $\mathbb{C}_{T_{MR1,3SAT}} \neq 0$ indicates that at least one clause remains unsatisfied. Synaptic weights, which define the strength of interconnections between neurons, are computed using the WA method [19]. In this method, $\mathbb{C}_{T_{MR1,3SAT}}$ in Eq (4.3) is compared with the Lyapunov energy function ($L_{T_{MR1,3SAT}}$). The formulation of $L_{T_{MR1,3SAT}}$ is

$$\begin{aligned} L_{T_{MR1,3SAT}} = &-\frac{1}{3}\sum_{i=1,i\neq j\neq k}^{N}\sum_{j=1,i\neq j\neq k}^{N}\sum_{k=1,i\neq j\neq k}^{N} W_{ijk}^{(3)} S_i S_j S_k \\ &- \frac{1}{2}\sum_{i=1,i\neq j}^{N}\sum_{j=1,i\neq j}^{N} W_{ij}^{(2)} S_i S_j - \sum_{i=1}^{N} W_i^{(1)} S_i. \end{aligned} \tag{4.4}$$

The synaptic weight can usually be estimated in advance using the WA method. However, there are a few assumptions that need to be taken into account before determining the synaptic weight of the MR1,3SAT. All variables in $T_{MR1,3SAT}$ are non-redundant, and when each clause is independent of the others $i \in \{1,2,3,\ldots,N\}$, no self-connections exist among the neurons in $C^{(k)}$, where $W_{ii} = 0$, $\forall i \in \{1,2,3,\ldots,N\}$. Furthermore, the symmetry of DHNN ensures that the $W_{ij} = W_{ji}$ and W_{ijk} is equivalent across all possible permutations of *ijk*.

The learning phase in DHNN is guided by the WA method, as conventional Hebbian learning becomes less effective when dealing with networks containing a large number of neurons. Hebbian learning, as noted by Foldiak [47], was initially introduced to determine synaptic weights in DHNN. However, its reliance on local correlation between neurons often limits scalability and interpretability. Unlike Hebbian learning, the WA method offers greater reliability by reducing inaccuracies in the synaptic weights stored within CAM [48]. These optimized weights are maintained in CAM, and later applied during the retrieval phase [49], where the final state of each neuron is computed based on the local field equation as defined in

$$L_i(t) = \sum_{k=1,k\neq j}^{N}\sum_{j=1,j\neq k}^{N} W_{ijk}^{(3)} S_k S_j + \sum_{j=1,j\neq i}^{N} W_{ij}^{(2)} S_j + W_i^{(1)}. \tag{4.5}$$

The coefficient $W_{ijk}^{(3)}$ refers to the interaction weight among neurons *i*, *j*, and *k*, within the third-order clause. In contrast, $W_{ij}^{(2)}$ specifies the weight assigned to the pairwise connection between neurons *i* and *j*, while $W_i^{(1)}$ indicates the weight linked to a single neuron *i* in a first-order clause. Properly structuring these weights is vital to preserve both stability and convergence of the DHNN-MR1,3SAT model. The computed local field, L_i, is subsequently passed through the activation function, as formulated in Eq (4.5). Specifically, the model employs the hyperbolic tangent activation function, which transforms the calculated values of L_i into a bipolar state, as formulated in Eq (4.6) [50]:

$$S_i^f(t) = \begin{cases} -1, & tanh(L_i) < 0, \\ 1, & tanh(L_i) \geq 0. \end{cases} \tag{4.6}$$

The subsequent step assesses the quality of the network's output state by applying the Lyapunov energy function, $L_{T_{MR1,3SAT}}$, and the absolute minimum energy, $L_{T_{MR1,3SAT}}^{min}$, as defined in Eqs (4.4) and (4.7), respectively:

$$L_{T_{MR1,3SAT}}^{min\frac{m_i n_i}{8\ \ 2}}, \tag{4.7}$$

where m_i and n_i represent the number of third-order and first-order clauses in the MR1,3SAT, respectively. The resulting neuron state can converge to the global minimum energy, as specified in

$$\left| L_{T_{MR1,3SAT}} - L_{T_{MR1,3SAT}}^{min} \right| \leq Tol, \tag{4.8}$$

whereby $Tol = 0.001$ is a predefined tolerance value used to distinguish between global and local minimum energy levels. The threshold value has a significant impact on the quality of solutions produced by the DHNN, resulting in better performance compared to other values [20]. If the condition in Eq (4.8) is not met, the final neuron state becomes trapped in a local minimum energy state.

In this study, the retrieval phase multi objective function is formulated to achieve the global minimum, while also considering diversity fitness and the similarity index ratio of the final neuron state, S_i^f produced before and after applying the IABC method. Accordingly, the retrieval phase in DHNN is framed as a multi-objective optimization problem, as outlined in Eqs (4.9)–(4.11):

$$Obj\left(M_{gs}, M_{ds}, Min_{GLI}\right), \tag{4.9}$$

such that

$$M_{gs} = S_i^f, \text{ where } S_i^f \rightarrow \text{Eq (4.8)}, \tag{4.10}$$

$$M_{ds} \geq r, \tag{4.11}$$

$$Min_{GLI} \leq \rho. \tag{4.12}$$

The multi-objective function is guided by three key requirements.

First, the generated solutions S_i^f must be able to attain the global minimum. Any solution fulfilling the requirement in Eq (4.8) is denoted as in maximum global solution, M_{gs} and is obtained through the process described in Eqs (4.9)–(4.12).

Second, the solution S_i^f must preserve the diversity fitness of the neuron states within a defined maximum diversity solution, M_{ds}. This condition is essential to guaranteeing that the obtained solutions S_i^f maintain adequate variability, represented by the inclusion of negative neuron states. To preserve this variability, each solution S_i^f is required to meet a minimum threshold of r. The evaluation of diversity fitness is conducted according to the procedures in Eqs (4.13) and (4.14):

$$M_{ds} = r\left[\sum_{i=1}^{v} ds\left(C_i^*\right)\right] \text{ under condition} \tag{4.13}$$

$$M_{ds_i}(C_i^*) = \begin{cases} 1, & (S_i^{af}, S_j^{af}, S_k^{af}) \neq 1, \\ 0, & otherwise, \end{cases} \tag{4.14}$$

where M_{ds} measures the diversity fitness based on the frequency of ds_i for each clause C_i^*. As indicated in Eq (4.13), the diversity score becomes zero when all neuron states associated with C_i^* take the value of 1 (positive). Therefore, the formulation of r is shown in

$$r = \frac{M_{ds}}{tc},$$

where *tc* is the total number of clauses in $T_{MR1,3SAT}$.

Third, MR1,3SAT must ensure that the similarity between the solutions generated before and after the application of IABC remains within a predefined similarity ratio, ρ. The solutions S_i^f prior to and following the implementation of IABC are denoted as before S_i^{bf} and after S_i^{af}. Maintaining a level of dissimilarity among all generated solutions is essential to support solution diversity. To evaluate this similarity of S_i^f, the Gower Legendre similarity index (GLI) index is employed. The formulation of the minimum similarity index Min_{GLI} within the MR1,3SAT context is detailed in Eq (4.15) through (4.19):

$$GLI = \frac{a+d}{a+0.5(b+c)+d}, \tag{4.15}$$

$$a = \begin{cases} 1, & \text{if } (S_i^{bf}, S_i^{af}) = (1,1), \\ 0, & otherwise, \end{cases} \tag{4.16}$$

$$b = \begin{cases} 1, & \text{if } (S_i^{bf}, A_i^{af}) = (1,-1), \\ 0, & otherwise, \end{cases} \tag{4.17}$$

$$c = \begin{cases} 1, & \text{if } (S_i^{bf}, S_i^{af}) = (-1,1), \\ 0, & otherwise, \end{cases} \tag{4.18}$$

$$d = \begin{cases} 1, & \text{if } (S_i^{bf}, S_i^{af}) = (-1,-1), \\ 0, & otherwise. \end{cases} \tag{4.19}$$

Therefore, the final solutions S_i^f of the MR1,3SAT to be adopted as the new final solution S_i^{new} in MR1,3SAT must satisfy the criterion. Specifically, the overall objective function is formulated as a weighted sum of the three components, where each objective contributes to the evaluation of candidate solutions during the search process. The combined objective function can be expressed as

$$F_{objective} = \alpha f_{global} + \beta f_{divesity} + \gamma\left(1 - f_{similarity}\right), \tag{4.20}$$

where f_{global} represents the global solution quality, $f_{divesity}$ measures the diversity among candidate solutions, and $f_{similarity}$ quantifies the similarity between solution patterns. The coefficients α, β, and γ denote the weighting parameters controlling the contribution of each objective. The overall training using and retrieval process of the DHNN-MR1,3SAT model is outlined in the pseudocode presented in Algorithm 1 of the optimization method that using IABC algorithm. Meanwhile, Figure 1 presents the schematic diagram of the DHNN-MR1,3SAT process. Each major block represents either first- or third-order logic producing clauses of different orders. The blue lines highlight the implementing of first- and third-order clauses.

Algorithm 1. The pseudocode of DHNN-MR1,3SAT

Input: Set the initial parameters such as number of trials (*NT*) and maximum combination (*combmax*).
Begin
Assign neurons to represent each variable in $S_i \in [S_1, S_2, S_3, \dots, S_b]$;
  while $(i \leq NT)$
  Form the initial states using Eq (4.1);
  Learning phase
  Define $L_{T_{MR1,3SAT}}$ using Eqs (4.2) and (4.3);
    for $S_i \in [S_1, S_2, S_3, \dots, S_b]$ **do**
    Check clauses satisfaction;
      if $L_{T_{MR1,3SAT}} = 0$
      C_i is satisfied;
      else
      C_i is not satisfied;
      end if
    Compute synaptic weights using WA method;
    Store the synaptic weight in CAM;
    end for
  Retrieval phase
  Compute L_i using Eq (4.4);
  Compute $L_{T_{MR1,3SAT}}$ using Eq (4.5);
  Compute $L_{T_{MR1,3SAT}}^{min}$ using Eq (4.6);
    for $S_i^f \in \left[S_1^f, S_2^f, S_3^f, \dots, S_b^f\right]$ **do**
    Check the quality of the final neuron state using Eq (4.8);
      if $S_i^f \rightarrow Obj(M_{GS}, M_{DS}, Min_s)$
      Assign as S_i^{af};
      else
      do Intelligent Artificial Bee Colony;
      end if
    end for
  end while
Output: Final neuron states achieved multi-objective function

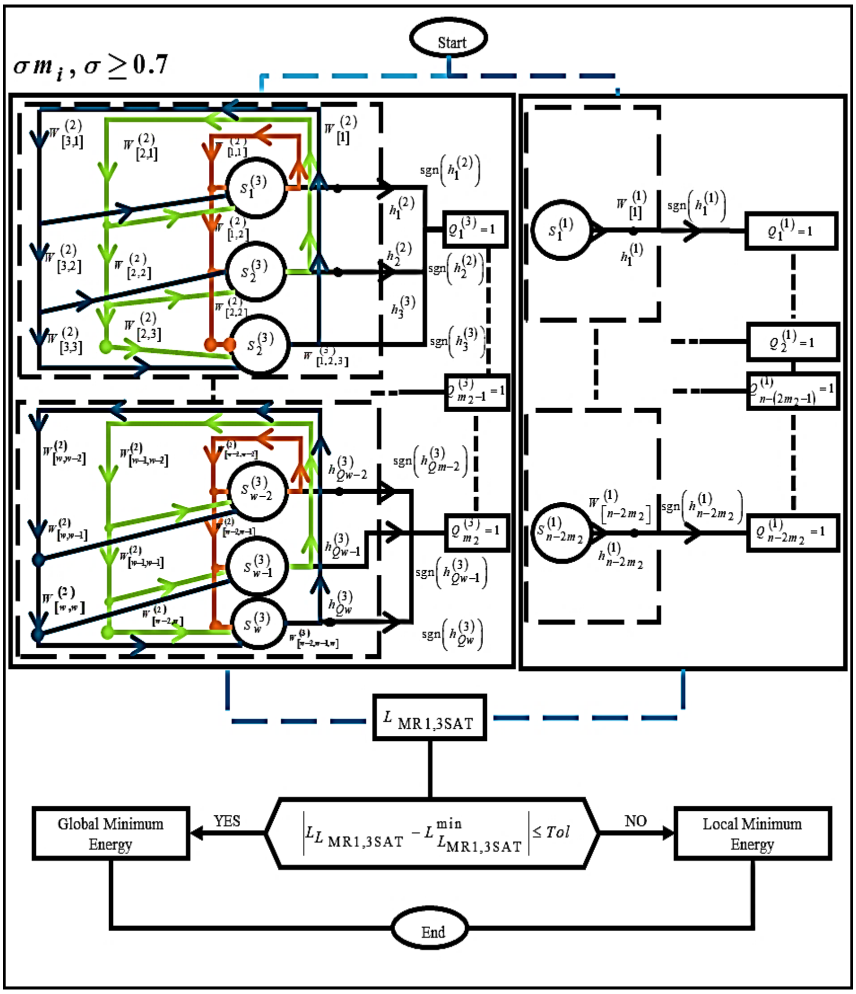

Figure 1. The schematic diagram of DHNN-MR1,3SAT.

5. Existing *k* satisfiability based reverse analysis method

The k-satisfiability based reverse analysis (kSATRA) model transforms dataset features into logical inference rules. Kho et al. [27] extended the foundational reverse-analysis approach of Sathasivam and Abdullah [26] by introducing a 2-satisfiability based reverse analysis (2SATRA) method. This method allowed incorporating logical rules by enhancing the generalizability and robustness of the inferred behavioral model [27]. As the DHNN operates in bipolar state space, all attributes must be encoded within the interval [-1,1]. Proper attribute selection is essential, as it identifies the key influencing factors within the dataset. To address kSATRA's lack of an output-neuron selection mechanism, Jamaludin et al. [31] and Kasihmuddin et al. [32] employed supervised learning techniques to improve attribute selection. This supervised method also removes irrelevant attributes that may lead to overfitting in the induced logic. As part of this method, the optimal logic rule set $P_{k\mathrm{SAT}}^{best}$ is determined by examining the structure of k-order clauses associated with $P_{train} = 1$. The formal definition of the formulation of $P_{k\mathrm{SAT}}^{best}$ as per Eq (5.1):

$$P_{k\mathrm{SAT}}^{best} = max\left[n\left(C_i^{(k)}\right) \rightarrow P_{train} = 1\right]. \tag{5.1}$$

Here, $n\left(C_i^{(k)}\right)$ represents the frequency of clause structures that correspond to $P_{train} = 1$. To illustrate an example, where A and B are two attributes, with values $A \in \{-1,1\}$ and $B \in \{-1,1\}$. In the case of 2-SAT, the possible clause structures include $C_1 = A \vee B$, $C_2 = \neg A \vee B$, $C_3 = A \vee \neg B$, and $C_4 = \neg A \vee \neg B$. If the current values of A and B are -1 and 1, respectively, kSATRA will select $C_2 = \neg A \vee B$ as the clause that best represents that instance in the dataset. Using this approach, kSATRA transforms the entire dataset into 2-SAT logical form based on the state of each attribute. After that, it evaluates all generated 2-SAT clauses and identifies the one that appears most frequently. This most common clause structure is referred to as $P_{k\mathrm{SAT}}^{best}$, which is defined in Eq (5.1).

Once $P_{k\mathrm{SAT}}^{best}$ is obtained, it can be trained by determining the optimal synaptic weights of the DHNN. The logical structure of $P_{k\mathrm{SAT}}^{best}$, along with its corresponding synaptic weights, is then stored in the DHNN as content-addressable memory. Afterward, the induced logic $P_{k\mathrm{SAT}}^{induced}$ is generated based on the local field computation. The $P_{k\mathrm{SAT}}^{induced}$ is then compared with P_{test} to evaluate $P_{k\mathrm{SAT}}^{induced}$ quality. Importantly, the $P_{k\mathrm{SAT}}^{induced}$ achieving the highest accuracy will be selected to represent the behavior of the dataset. Note that the $Q_{k\mathrm{SAT}}^{i}$ with the highest accuracy will be chosen to explain the behavior of the dataset. Figure 2 illustrates the overall workflow of kSATRA, which consists of several phases involved in the logic mining process.

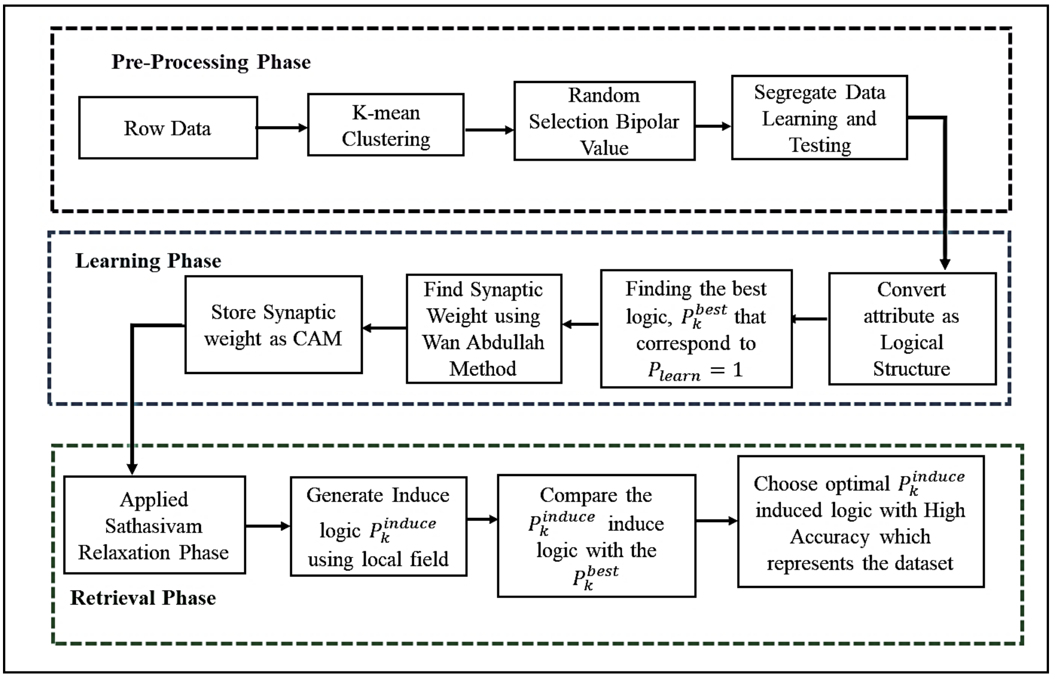

Figure 2. The general process of kSATRA.

6. Multi-unit major random 1,3-satisfiability based reverse analysis method

To enhance the effectiveness of the original kSATRA approach, this study introduces new logic mining technique called the multi-unit major random 1,3-satisfiability based reverse analysis method (MR1,3SATRAμ). The main goal of this method is to extract significant knowledge from a dataset and transform it as logical rules, with a focus on identifying the most relevant attributes. The efficiency of logic mining is largely determined by the ability of DHNN-MR1,3SAT combined with the reverse analysis strategy to process data and encode it into symbolic logic. The MR1,3SATRAμ approach is organized into three distinct stages: pre-processing, training, and retrieval.

In the pre-processing phase, each dataset entry $S_{i,j}$ from the raw is transform into a bipolar form through *k*-means clustering. This transformation to ensure that the final neuron state in the DHNN converges properly [32,33]. Selecting ***N*** optimal attributes is crucial to prevent the induced logic from overfitting due to the presence of irrelevant features [34]. To enhance the quality of the induced logic, MR1,3SATRAμ uses Jaccard analysis to identify the most relevant ***N*** attributes. As highlighted by Zamri et al. [35], Jaccard analysis measures the similarity between positive instances in the dataset and the dependent (decision) attribute. Attributes that show a strong correlation with the decision attribute are selected to represent the dataset, denoted as $S_{i,j}$. In other words, this Jaccard analysis approach helps identify the key attributes that will guide the construction of the MR1,3SAT logic. The formula for the Jaccard analysis is provided in Eqs (6.1) and (6.2):

$$JFS_{i,j} = \frac{\sum_{j=1}^{n} a_j}{\sum_{j=1}^{n} a_j + \sum_{j=1}^{n} b_j + \sum_{j=1}^{n} c_j}, \tag{6.1}$$

where

$$J_{S_i^{output}, S_{i,j}} = \left\{ \left(S_i^{output}, S_{i,j} \right), \quad i = 1,2,\ldots\ldots,m \ \ and \ \ j = 1,2,\ldots\ldots,n \right\},$$
$$a_{i,j}(e_j) = \begin{cases} 1, & (1,1), \\ 0, & Otherwise, \end{cases} \quad b_{i,j}(e_j) = \begin{cases} 1, & (1,-1), \\ 0, & Otherwise, \end{cases} \quad c_{i,j}(e_j) = \begin{cases} 1, & (-1,1), \\ 0, & Otherwise. \end{cases} \tag{6.2}$$

S_i^{output} represents the decision output for each entry. A *JHS* value indicates significant dissimilarity between the output and $S_{i,j}$, where $S_{i,j}$ denots the value of attribute *S* for the *i*th entry and the *j*th attribute [35]. Attributes with elevated *JFS* values will be prioritized for selection. $\{a_{i,j}, b_{i,j}, c_{i,j}\}$ denotes the scoring mechanism measures the similarity differences between the *i*th entry of the independent attribute and the *i*th entry of the dependent attributes across the dataset with *n* total number of entries.

As indicated in Eq (6.2) (e_{ij}) entries with the highest *Jacc* will be selected. In this context, $L_{MR1,3SAT}$ in MR1,3SATRAμ can capture more irregular and inconsistent patterns by providing valuable information to the CAM in DHNN. The final classification results will be based on the average of all cross-validation folds in MR1,3SATRAμ.

In this study, *JHS* is employed to rank the attributes according to their similarity and relevance within the dataset. The top ten attributes are selected for subsequent analysis. The choice of ten attributes aims to balance model interpretability and computational efficiency. Selecting a smaller number of attributes may omit important symptom information, while selecting a larger number of attributes would significantly increase the dimensionality of the logical clause search space in the MR1,3SAT framework, potentially introducing redundancy and increasing computational complexity in the DHNN training process. Therefore, the selection of ten attributes provides a reasonable trade-off that preserves informative symptom patterns while maintaining manageable computational complexity and interpretable logical rule extraction.

During the training phase, MR1,3SATRA introduces an alternative approach for computing the best logic by aiming to maximize both true positive (*TP*) and true negative (*TN*) outcomes. Initially, the DHNN generates p unique logical combinations based on the $T_{MR1,3SAT}$ clause structure. An example of such a generated logical combination is provided in Eq (3.4). These p logical combinations, denoted as $P_{\text{MR1,3SAT}}^{best}$, are then evaluated against all entries in the training dataset, P_{train} to calculate the components of the confusion matrix: *TP*, *TN*, false positive (*FP*), and false negative (*FN*).

Figure 3 presents the classification outcomes in the confusion matrix during the MR1,3SATRAμ training phase.

		Training outcome generated from the $P_{\text{MR1,3SAT}}^{best}$	
		Positive (1)	Negative (-1)
Outcome from the P_{train}	Positive (1)	TP	FN
	Negative (-1)	FP	TN

Figure 3. Classification outcomes of MR1,3SATRAμ in the learning phase by confusion matrix.

The optimal logic is identified as the combination producing the greatest sum of *TP* and *TN*, ensuring the dataset is represented through the most effective logical structure. The combination selected as $P_{\text{MR1,3SAT}}^{best}$ is the one that best captures the overall trend in P_{train} by maximizing the sum of *TP* and *TN*, as expressed in

$$P_{\text{MR1,3SAT}}^{best} = max\left[\left(P_{\text{MR1,3SAT}}^{best}, P_{train}\right) \rightarrow (TP + TN)\right]. \tag{6.3}$$

Next, the optimal logic $P_{\text{MR1,3SAT}}^{best}$ is trained within the DHNN-MR1,3SAT model. During the retrieval phase, the final neuron states, represented as S_i^f, are derived through the formulations in Eq (6.4). These final states are then mapped to induced variables according to the following definition:

$$S_i^{induced} = \begin{cases} A_i, S_i^f = 1, \\ \neg A_i, S_i^f = -1. \end{cases} \tag{6.4}$$

The set of induced rules $S_i^{induced}$ forms the final logic representation, capturing the data patterns identified by DHNN-MR1,3SAT. Next, $P_{\text{MR1,3SAT}}^{induced}$ is applied to the test data P_{test} to compute the confusion matrix components.

Figure 4 shows how the confusion matrix is classified for MR1,3SATRAμ in the retrieval phase.

		Retrieval outcome generated from the $P_{\text{MR1,3SAT}}^{induced}$	
		Positive (1)	Negative (-1)
Outcome from the P_{test}	Positive (1)	TP	FN
	Negative (-1)	FP	TN

Figure 4. Classification outcomes of MR1,3SATRAμ in the retrieval phase by confusion matrix.

All the variables in the retrieved $P_{\mathrm{MR1,3SAT}}^{induced}$ are permuted, to widens the search space and improves the possibility of deriving optimal induced logic $P_{\mathrm{MR1,3SAT}}^{induced}$. Thus, the structure of $T_{MR1,3SAT}$ is not limited to the standard model presented by Kho et al. [27], but instead allows for the generation of more varied logical structures.

7. The proposed intelligent artificial bee colony algorithm in optimizing retrieval phase of DHNN-MR1,3SAT

In this thesis, the intelligent artificial bee colony (IABC) algorithm is utilized as an optimization mechanism in the retrieval phases to enhance the quality final neuron state. In the learning phase, EA are employed to obtain a valid solution state, as described by Sathasivam et al. [51]. The IABC method begins with the initialization stage, where the population of food sources is established according to Eq (7.1), as outlined below:

$$FS_i\left(S_i^{bf}\right) = \begin{cases} 1, & tanh(\,h_i) \geq 0, \\ -1, & Otherwise. \end{cases} \tag{7.1}$$

In Eq (7.1), S_i^{bf} represents the final neuron states before optimization according to $S_i^{bf} \in [1, NN]$, while $fs_i \in [1, NT]$. Here, *NN* denotes the total number of neurons, while *NT* represents the number of trials conducted. The initialization of the population of food source before optimization $P_B\left[FS_i\left(S_i^{bf}\right)\right]$ is obtained by squashing the updated local field, h_i using HTAF can be described in Eq (7.2):

$$\begin{aligned} &FS_1 \colon S_1^{bf} S_2^{bf} S_3^{bf} S_4^{bf} S_5^{bf} \ldots\ldots\ldots S_{NN}^{bf} \\ &FS_2 \colon S_1^{bf} S_2^{bf} S_3^{bf} S_4^{bf} S_5^{bf} \ldots\ldots\ldots S_{NN}^{bf} \\ &FS_3 \colon S_1^{bf} S_2^{bf} S_3^{bf} S_4^{bf} S_5^{bf} \ldots\ldots\ldots S_{NN}^{bf} \\ &FS_4 \colon S_1^{bf} S_2^{bf} S_3^{bf} S_4^{bf} S_5^{bf} \ldots\ldots\ldots S_{NN}^{bf} \\ &. \\ &. \\ &. \\ &FS_{NT} \colon S_1^{bf} S_2^{bf} S_3^{bf} S_4^{bf} S_5^{bf} \ldots\ldots\ldots S_{NN}^{bf}. \end{aligned} \tag{7.2}$$

After initialization of the population of the food source (FS_i), the IABC applied the first two objective functions as defined in Eqs (7.2) and (7.3) to evaluate finalized neuron states with emphasis on both the diversification of negativity M_{ds} and the attainment of global minimal solutions M_{gs}. In particular, this study seeks to ensure solution diversity by enforcing a negativity degree of *r%*, whereby the refined final neuron states must contain at least *r*%, negative literals across the clause set. This mechanism prevents repetitive neuron states and promotes greater variation in the induced logic. The percentage of the negativity determines the IABC by having at least *r%* of the clauses contain about one negated state for each FS_i with Eq (7.2) can declare the diversity solution as

$$M_{ds} = r\ \left[\sum_{i=1}^{v} ds\ (C_i^*)\right], \text{ such that } r = [0.5] \tag{7.3}$$

under the condition

$$M_{ds_i}(C_i^*) = \begin{cases} 1, & \left(S_i^{af}, S_j^{af}, S_k^{af}\right) \neq 1, \\ 0, & Otherwise. \end{cases}$$

Referring to Eq (7.3), v represents the number of clauses C_i^*, while r defines the diversity ratio in terms of negativity of the final neuron sate, which is specified within the range of 0 to 1. According to Eq (7.3), each solution string must contain at least one negative literal in a clause C_i^* to be considered diversified. An alternative approach to the global minima ratio is to compute the satisfied interpretation for each FS_i. In this case, an FS_i is considered to achieve a global minima solution if it attains the maximum satisfied interpretation. The total number of global minima solutions can be determined using Eqs (7.4) and (7.5), as presented below:

$$M_{gs} = \sum_{i=1}^{v} ds_i\,(C_i^*), \tag{7.4}$$

under the condition

$$M_{gs_i}(C_i^*) = \begin{cases} 1, & satisfied, \\ 0, & otherwise. \end{cases} \tag{7.5}$$

The updated position of Bee_i within the food source population is determined through iterative search operations performed by employed bees, onlooker bees, and scout bees. In the employed bee phase, candidate solutions FS_i (new updated neuron state) are explored in the neighborhood of the current food source, which corresponds to the existing final neuron state stored in memory. Equation (7.6) updates the position of the FS_i, and is formulated as below:

$$v_{ij} = x_{ij} \,\overline{\wedge}\, \left(\phi_{ij}\left(x_{ij} \,\overline{\wedge}\, x_{kj}\right)\right), \tag{7.6}$$

such that

$$\phi_{ij} = \begin{cases} -1, & rand(0,1) < 0.5, \\ 1, & rand(0,1) \geq 0.5. \end{cases} \tag{7.7}$$

As per Eq (7.6), v_{ij} represents the new food source FS_i, and x_{kj} denotes as random food source FS_i. Here, $\overline{\wedge}$ represents the NAND logic gate operator as shown in Table 1. According to Eq (7.7), the value of ϕ_{ij} is executed depending on the generated random value. If the fitness value of the new food sources v_{ij} proves to be better than the existing food source x_{ij}, the employed bees discard the memory of the existing food source x_{ij} and replace the existing position of the food source. Conversely, if x_{ij} cannot be improved, the employed bee will attempt up to five trials to enhance the new food source. During this phase, exploration is performed, where employed bees search within the hive's environment (solution space) to identify potential food sources (solutions FS_i).

The configuration of the possible input and output variables for each logic gate can be determined based on the truth table in Table 1.

Table 1. The truth table of NAND.

NAND ($\overline{\wedge}$)		
A	**B**	**Output**
-1	-1	1
1	-1	1
-1	1	1
1	1	-1

Upon completing the search process, employed bees return to the hive and share information about food source locations and nectar amounts through the dance area characterized by multiple interactions. Onlooker bees evaluate this information and select food sources according to the probability proportional to nectar quality. In this context, the selection process is performed using a roulette wheel mechanism, where each segment corresponds to the FS_i fitness values. The P_i probability of fitness is formulated in

$$P_i = \frac{f_{FS_i}}{\sum_{i=1}^{NN} f_{FS_i}}. \tag{7.8}$$

Referring to Eq (7.8), if the random value ranges from 0 to 1 is less than P_i, the position of the new food source will be updated using Eq (7.6). Otherwise, the position of the bee remains. Employed bees with higher fitness f_{FS_i} values are more likely to be selected by onlooker bees. Once an employed bee is chosen by an onlooker bee, indicating positive feedback a new food source (FS_i) is generated using Eqs (7.6) and (7.8) until the maximum limit is reached.

If the food source (FS_i) generated by the onlooker bee fails to achieve the highest fitness value, this indicates that it is not an optimal food source. So, the scout bee will initiate a new iteration until the maximum limit is reached to achieve a new food source. If a suboptimal solution is obtained, the algorithm initiates a new iteration to generate a fresh population of bees that allows exploration of new food sources in the search space. Intelligent mutation is a powerful optimization strategy the combination of the strengths of GA [50]. This approach begins by selecting promising individuals from a GA-initialized population based on their fitness scores. This probabilistic model guides the mutation process to ensure that introduced variations are more likely to enhance solution quality while maintaining diversity. Mutated individuals are evaluated, and those meeting predefined fitness criteria are implemented back into the population of the bees. This iterative process of selection, modeling, mutation, and evaluation creates a robust search strategy, effectively balancing exploration and exploitation to reach optimal solutions.

Another mechanism introduced in the IABC is the intelligent mutation concept. The intelligent mutation mechanism introduces a random change in the x_{ij} values (FS_i), thereby enhancing their diversity and contributing to overall improvement. Due to the random changes introduced by intelligent mutation, it primarily focuses on exploiting localized search operations to find a solution. To further improve the exploration capability of the search process, intelligent mutation operator is incorporated into the ABC algorithm, which becomes IABC. The mutation operator randomly modifies selected neuron states with a mutation probability ρ_m, which controls the likelihood of perturbation in candidate solutions. This mechanism introduces controlled variability into the solution population while preserving the general structure of the neuron states. After mutation is applied, the candidate solution is evaluated using the defined fitness function. The mutated solution is accepted only if its fitness value is equal to or greater than that of the original solution otherwise, the original solution is retained. This acceptance mechanism ensures that the mutation process does not degrade the overall solution quality while helping the algorithm avoid premature convergence and maintain diversity within the search space. In this context, this algorithm will intelligently mutate the solution to meet the requirements specified in Eq (7.9). The intelligent mutation mechanism is exclusively applicable to the third-order clause of MR1,3SAT as the first-order clause has been intentionally separated from the optimization process. Therefore, the solution with the highest fitness value obtained from the food source will be selected for the next stages. This optimal solution will serve as the initial population for the next generation. Equation (7.9) represents a mutation:

$$FS_i = \rho_m \ (v_{ij}), \tag{7.9}$$

whereby ρ_m is the probability mutation rate of the unsatisfied clauses from GA. Equation (7.8) aligns with the highest probability of the search space by guiding the fitness function of FS_i.

Figure 5 presents the complete implementation of MR1,3SATRAμ, encompassing both the pre-processing and retrieval phases, where the dotted lines denote the starting point of each phase.

The corresponding pseudocode of the proposed MR1,3SATRAμ is outlined in Algorithm 2.

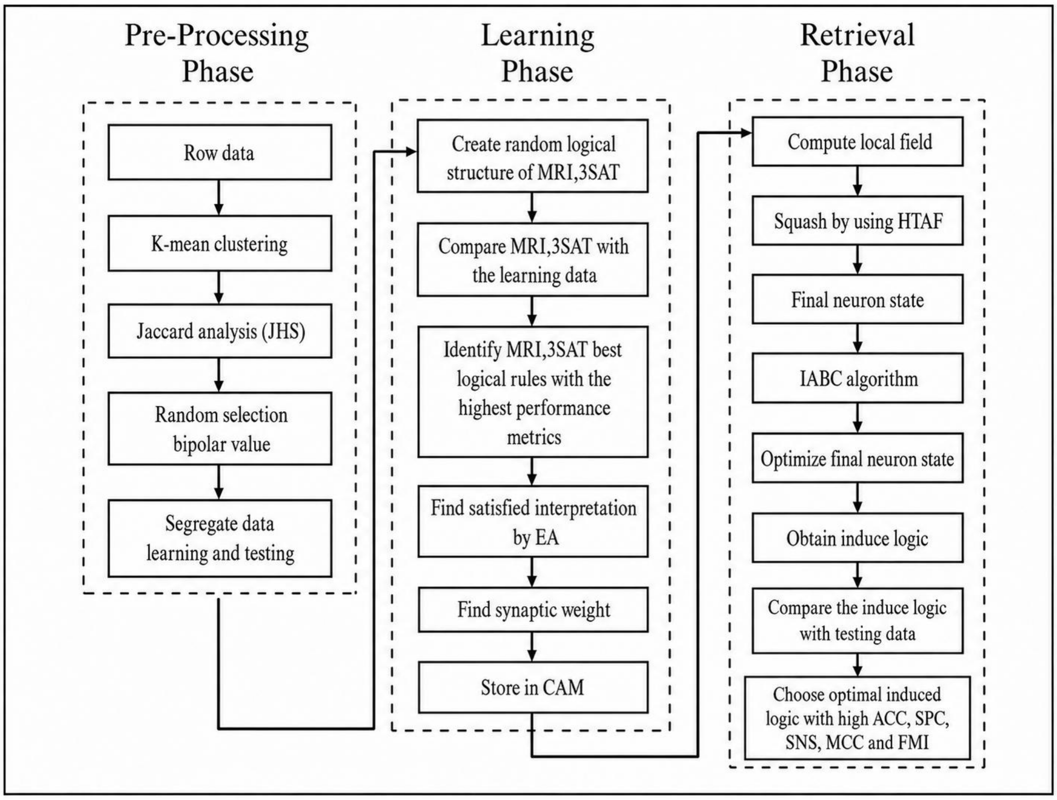

Figure 5. The flowchart of the proposed MR1,3SATRAμ.

Algorithm 2. The pseudocode of MR1,3SATRAμ

Input: Set all attributes of $A_1, A_2, A_3, \ldots, A_n$ with respect to P_{train}, P and NT.
Begin
Initialize algorithm parameters;
Pre-processing Phase
Find the *p*-value for each attribute;
  for $(\boldsymbol{n}(\boldsymbol{S_i}) = \boldsymbol{NN})$ **do**
    if $(\boldsymbol{JHS} \geq \boldsymbol{averagemin}())$
    Assign $\boldsymbol{A_i}$ as $\boldsymbol{S_i}$;
    else
    Not selected as S_i;
    end if
  end for
  while $(i \leq P)$ **do**
  Learning Phase
  Generate $P_{MR1,3SAT}^{b_i}$;
    for $(\boldsymbol{i} \leq \boldsymbol{p})$ **do**
    Compare the outcome from the P_{train} and the generated $P_{MR1,3SAT}^{b_i}$;
    Select $\boldsymbol{P_{MR1,3SAT}^{best}}$ using Eq (5.1);
    end for
  Check the clause satisfaction for $\boldsymbol{P_{MR1,3SAT}^{best}}$;
  Compute the synaptic weight associated with $\boldsymbol{P_{MR1,3SAT}^{best}}$ using WA method
  Store the synaptic weight and $\boldsymbol{P_{MR1,3SAT}^{best}}$ in CAM;
  Retrieval Phase
  Initialize the final neuron state;
    for $(\boldsymbol{t} \leq \boldsymbol{NT})$**do**
    Compute L_i using Eq (4.5);
    Convert S_i^f to bipolar state using Eq (4.6);
    Transform S_i^f into logical form using Eq (6.4) with respect to the attributes;
    Combine S_i^f to generate induced logic $P_{\mathrm{MR1,3SAT}}^{induced}$;
    Compare the outcome of the $P_{\mathrm{MR1,3SAT}}^{induced}$ with P_{test};
    $\boldsymbol{t} \leftarrow \boldsymbol{t} + \boldsymbol{1}$;
    end for
  $i \leftarrow i + 1$;
  end while
Output: The best induced logic $P_{\mathrm{MR1,3SAT}}^{induced}$ with the highest *Acc, Spec, Sen, FMI and MCC*

8. Experimental setup

The simulation was performed to assess the capability of MR1,3SATRAμ in generalizing the most effective induced logic that characterizes the behavior of the dataset. Following cross validation to identify the most significant attributes, MR1,3SATRAμ was implemented in the logic mining process using C++ programming, developed and compiled in DEV C++ Version 5.11. All experimental procedures, including the implementation of the algorithm, logic formulation, and performance evaluation, were executed through custom developed C++ codes. The simulations were conducted on a Windows 10 system configured with an AMD Ryzen 5 3500U processor, Radeon Vega Mobile Graphics, and 8GB of RAM.

8.1. Datasets

The Alzheimer's disease neuroimaging initiative (ADNI) dataset was acquired from the official website of the ADNI (https://adni.loni.usc.edu/) after user registration to ensure authorized and unbiased data access. The dataset was downloaded on December 24, 2023, and comprised 28 attributes, including 27 independent attributes and 1 dependent attribute. It captured symptoms associated with adverse events experienced by individuals with Alzheimer's disease, with the dependent attribute representing the diagnosis determined based on the 27 symptomatic attributes.

Table 2 provides detailed information about the ADNI dataset used for the experiments. The dataset attributes are represented in a discrete bipolar form (1 and -1) to ensure compatibility with the DHNN structure. To achieve this representation, continuous attribute values were first normalized and then discretized using the *k*-means clustering technique, as described in Choi et al. [52]. This transformation enables the datasets to be encoded into binary into neuron states suitable for logical rule representation in the MR1,3SAT model. For missing data, values were substituted with a random bipolar value (1 or -1) when the proportion of missing entries was relatively small. According to Choi et al. [52], if more than 10% of the dataset is contains missing value, more sophisticated imputation techniques should be employed to avoid introducing bias into the learning process. Table 3 provides an overview of the characteristics of the datasets used in the experimental analysis.

Table 2. Datasets used for experimental analysis.

Condition	Information
ADNI Website	Website (https://adni.loni.usc.edu/)
Date	24 December 2023
Number of features	28
Number of instances	4884
Number of independent variables	27
Number of dependent variables	1
Outcome based on the symptoms	Depression mood

Table 3. The characterization of all the datasets employed in the experiment.

Attribute label	Name of attribute	Bipolar value
MD 1	AXN (Nausea)	1 Yes (Nausea) -1 No (Nausea)
MD 2	AXVOMIT (Vomiting)	1 Yes (Vomiting) -1 No (Vomiting)
MD 3	AXDIARRH (Diarrhoea)	1 Yes (Diarrhoea) -1 No (Diarrhoea)
MD4	AXCONSTP`(Constipation)	1 Yes (Constipation) -1 No (Constipation)
MD5	AXABDOMN (Abdominal discomfort)	1 Yes (Abdominal discomfort) -1 No (Abdominal discomfort)
MD6	AXSWEATN (Sweating)	1 Yes (Sweating) -1 No (Sweating)
MD7	AXDIZZY (Dizziness)	1 Yes (Dizziness) -1 No (Dizziness)
MD8	AXENERGY (Low energy)	1 Yes (Low energy) -1 No (Low energy)
MD9	AXDROWSY (Drowsiness)	1 Yes (Drowsiness) -1 No (Drowsiness)
MD10	AXVISION (Blurred vision)	1 Yes (Blurred vision) -1 No (Blurred vision)
MD11	AXHDACHE (Headache)	1 Yes (Headache) -1 No (Headache)

Continued on next page

Attribute label	Name of attribute	Bipolar value
MD12	AXDRYMTH (Dry mouth)	1 Yes (Dry mouth) -1 No (Dry mouth)
MD13	AXBREATH (Shortness of breath)	1 Yes (Shortness of breath) -1 No (Shortness of breath)
MD14	AXCOUGH (Coughing)	1 Yes (Coughing) -1 No (Coughing)
MD15	AXPALPIT (Palpitations)	1 Yes (Palpitations) -1 No (Palpitations)
MD16	AXCHEST (Chest pain)	1 Yes (Chest pain) -1 No (Chest pain)
MD17	AXURNDIS (Urinary discomfort (e.g., burning)	1 Yes (Urinary discomfort) -1 No (Urinary discomfort)
MD18	AXURNFRQ (Urinary frequency)	1 Yes (Urinary frequency) -1 No (Urinary frequency)
MD19	AXANKLE (Ankle swelling)	1 Yes (Ankle swelling) -1 No (Ankle swelling)
MD20	AXMUSCLE (Musculoskeletal pain)	1 Yes (Musculoskeletal pain) -1 No (Musculoskeletal pain)
MD21	AXRASH (Rash)	1 Yes (Rash) -1 No (Rash)
MD22	AXINSOMN (Insomnia)	1 Yes (Insomnia) -1 No (Insomnia)
MD23	AXCRYING (Crying)	1 Yes (Crying) -1 No (Crying)

Continued on next page

Attribute label	Name of attribute	Bipolar value
MD24	AXELMOOD (Elevated mood)	1 Yes (Elevated mood) -1 No (Elevated mood)
MD25	AXWANDER (Wandering)	1 Yes (Wandering) -1 No (Wandering)
MD26	AXFALL (Fall)	1 Yes (Fall) -1 No (Fall)
MD27	AXOTHER (Other/Diagnosis)	1 Yes (Other/Diagnosis) -1 No (Other/Diagnosis)
MR_D	AXDPMOOD (Depressed mood)	1 Yes (Depressed mood) -1 No (Depressed mood)

For evaluation, the dataset was divided using the train-test split method following the approach proposed by Vu et al. [53], where 60% of the data was allocated for training, and the remaining 40% was reserved for testing. To further improve the robustness and reliability of the results, a 5-fold cross validation procedure was implemented, following the methodology adopted Sejuti and Islam [54]. During each fold, the model was trained using the training subset and evaluated on the corresponding validation subset. A fixed random seed was applied during the data partitioning process to ensure experimental reproducibility. In the proposed MR1,3SATRAμ model, Jaccard similarity analysis was employed to identify the most informative attribute for constructing logical clauses. Based on this analysis, the ten most significant attributes were selected for the logic rule structure. The number of neurons in the neural network model corresponded directly to the number of selected attributes, resulting in a minimal network consisting of six neurons. This configuration facilitates logic mining by integrating first- and third-order logic, thereby enhancing the model's ability to capture complex relationships among dataset attributes while maintaining interpretable logical representations. Table 4 presents the ten attributes selected using the Jaccard similarity analysis.

Table 4. Attribute selection based on Jaccard similarity analysis.

Attribute label	Description of attribute	Bipolar value
A	AXN (Nausea)	1 Yes (Nausea) -1 No (Nausea)
B	AXVOMIT (Vomiting)	1 Yes (Vomiting) -1 No (Vomiting)

Continued on next page

Attribute label	Description of attribute	Bipolar value
C	AXSWEATN(Sweating)	1 Yes (Sweating) -1 No (Sweating)
D	AXVISION (Blurred vision)	1 Yes (Blurred vision) -1 No (Blurred vision)
E	AXOTHER (Other/Diagnosis)	1 Yes (Other/Diagnosis) -1 No (Other/Diagnosis)
F	AXPALPIT (Palpitations)	1 Yes (Palpitations) -1 No (Palpitations)
G	AXCHEST (Chest pain)	1 Yes (Chest pain) -1 No (Chest pain)
H	AXURNDIS (Urinary discomfort (e.g., burning)	1 Yes (Urinary discomfort) -1 No (Urinary discomfort)
I	AXCRYING (Crying)	1 Yes (Crying) -1 No (Crying)
J	AXELMOOD (Elevated mood)	1 Yes (Elevated mood) -1 No (Elevated mood)
K	AXWANDER (Wandering)	1 Yes (Wandering) -1 No (Wandering)
MR_D	AXDPMOOD (Depressed mood)	1 Yes (Depressed mood)

8.2. Performance metrics

The performance of all logic mining models was assessed using multiple evaluation metrics derived from the confusion matrix, including accuracy (*Acc*) [55], precision (*Prec*) [56], specificity (*Spec*) [57], Matthews correlation coefficient (*MCC)* [57], and Fowlkes Mallows index (*FMI*) [56]. The mathematical formulations of these metrics are provided in Eqs (8.1)–(8.5):

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}, \tag{8.1}$$

$$Sen = \frac{TP}{TP+FP}, \tag{8.2}$$

$$Spec = \frac{TN}{TN+FP}, \tag{8.3}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \tag{8.4}$$

$$FMI = \sqrt{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}. \tag{8.5}$$

Acc and Sensitivity (*Sen*) are commonly employed to evaluate the effectiveness of logic mining classification models. *Acc* quantifies the proportion of correctly classified outcomes, combining TP and TN relative to the total number of predictions. A high *Acc* value suggests that the induced logic aligns well with the actual dataset outcomes. On the other hand, *Sen* effects the model capability to correctly detect positive cases, obtained by dividing TP by the sum of TP and FN. A high *Sen* value reflects the model's effectiveness in retrieving logic rules that represent positive cases while minimizing the omission of relevant instances.

Spec evaluates the model's ability to correctly identify negative cases. It is defined as the proportion of TN to the sum of TN and FP. A higher *Spec* indicates that the model consistently recognizes the negative (–1) class, which is particularly important when dealing with imbalanced datasets. To get a single measure of overall performance, we use *MCC*. *MCC* takes into account TP, TN, FP and FN all at once, making it a solid gauge of model quality even when classes are not equally represented. As Chicco and Jurman [55] note, an *MCC* close to 1 signals perfect classification, around *0* means the model is no better than random guessing, and near –1 indicates systematic misclassification. Finally, we include the *FMI* to assess how closely our predicted labels match the true labels. Luque et al. [56] mentioned that higher FMI values correspond to stronger agreement between prediction and reality.

8.3. Logic mining baseline models

To evaluate the effectiveness of the proposed MR1,3SATRAμ, comparisons are made against several established logic mining approaches. These baseline models employ different strategies for integrating SAT formulations with DHNN. A concise overview of each method is provided below:

(a) **RA** [26]: One of the earliest logic mining methods based on HornSAT. This method does not involve a pre-processing stage; instead, it directly generalizes the induced logical rules. In this study, RA is employed as a systematic second-order logic approach during pre-processing, with only HornSAT compliant $Q_{induced}^{i}$ single induced rules retained int the retrieval. In addition, RA adopts an objective function similar to that used in 2SATRA.

(b) **2SATRA** [27]: This pioneering logic mining model employs clauses consisting of 2 literals, denoted as Q_{2SAT}. During the pre-processing phase, Q_{2SAT}^{best} is formulated by selecting the most frequent literals from $C_i^{(2)}$ that satisfy $Q_{train} = 1$. However, the formulation of Q_{2SAT} is based on randomized attribute selection. In the retrieval phase, the final induced logic, represented as $Q_{2\mathrm{SAT}}^{i}$, is generalized to characterize the dataset.

(c) **3SATRA** [29] is the higher order logic mining approach that capitalize this method applies clauses consisting of three literals or Q_{3SAT}. In the pre-processing phase, Q_{3SAT}^{best} is formulated by considering the most frequency of $C_i^{(3)}$ that obtained $Q_{train} = 1$. However, the formulation of Q_{3SAT} is based on randomized attribute selection. Nevertheless, 3SATRA is capable of generating Q_{3SAT}^{i} logical rules that capture the underlying dataset representation.

(d) **E2SATRA** [30] is the energy-based logic mining that consider $Q_{2\mathrm{SAT}}^{i}$ with global minima energy. E2SATRA capitalizes on $H_{Q_{2SAT}}^{min}$ to ensure the generated $Q_{2\mathrm{SAT}}^{i}$ is a global solution. In the retrieval phase of the E2SATRA, the energy for all the final neuron state will be checked before being selected as $Q_{2\mathrm{SAT}}^{i}$. This process can ensure that the $Q_{2\mathrm{SAT}}^{i}$ retrieved by E2SATRA always achieves the global minimum energy.

(e) **P2SATRA** [31] is the improved conventional 2SATRA with permutation operator that considers possible attribute arrangement in $C_i^{(2)}$. P2SATRA uses Q_{2SAT} as a logical rule to represent the relationship of the dataset. In this context, P2SATRA capitalizes on expanding the search space in finding $Q_{2\mathrm{SAT}}^{i}$ by rearranging the selected attributes to increase the possibility of obtaining the best induced logic.

(f) **S2SATRA** [32] extended P2SATRA by incorporating a supervised attribute selection approach based on Pearson chi-square correlation analysis. This method helped identify attributes with strong associations to the target class, leading to enhanced performance that surpassed that of previous logic mining models.

(g) **A2SATRA** [33]: This supervised approach enhances attribute selection using a log-linear model and integrates permutation operators to widen the search $Q_{2\mathrm{SAT}}^{i}$. A2SATRA emphasizes careful selection and flexible arrangement of feature $C_i^{(2)}$, which improves classification accuracy. Unlike some other methods, the final neuron states here are not scaled according to fixed conditions.

To ensure fairness and avoid potential bias, all baseline models were implemented under the same DHNN process as MR1,3SATRAμ. Table 5 summarizes the key parameters applied across all models.

Table 5. Key parameters applied in logic mining model.

Parameter	MR13SATRAμ	S2SATRA	A2SATRA	E2SATRA	P2SATRA	RA	2SATRA	3SATRA
Number of neuron (*NN*)	9	6	6	6	6	6	6	9
NC	3	3	3	3	3	3	3	3
Neuron combination	100	100	100	100	100	100	100	100
Feature selection	Jaccard analysis	Correlation	Log linear	Random	Random	Random	Random	Random
Energy Tol	-	-	-	0.001	-	-	-	-
Learning iteration	100	100	100	100	100	100	100	100
Learning method	EA	ES	ES	ES	ES	ES	ES	ES
Selection rate	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Number of trials (NT)	100	100	100	100	100	100	100	100
CPU time	24 hours	24 hours	24 hours	24 hours	24 hours	24 hours	24 hours	24 hours
Logical permutation (P)	100	100	100	-	100	-	-	-
Activation function	HTAF	HTAF	HTAF	HTAF	HTAF	HTAF	HTAF	HTAF
Significance value (*p*)	0.05	0.05	0.05	-	-	-	-	-

9. Results and discussion

This section evaluates the performance of MR1,3SATRAμ in comparison with established logic mining approaches, namely 2SATRA, 3SATRA, A2SATRA, S2SATRA, E2SATRA, RA, and P2SATRA. The evaluation relies on multiple metric evaluation metrics, including *ACC*, *Sen*, *Spec*, *MCC*, and the *FMI* to ensure a well-rounded assessment. Collectively, these measures allow for a detailed analysis of how effectively each model performs across different datasets.

9.1. Effectiveness MR1,3SAT with the IABC in the retrieval phase

Figure 6 shows the results of the performance of a retrieval phase with and without IABC across four metrics: root mean square error (RMSE), mean analysis error (MAE), local minima ratio, and global minima ratio. The X-axis represents the number of neurons, while the Y-axis indicated the accuracy (%). The RMSE and MAE values remain consistently low when IABC is applied while they increase significantly without IABC indicating poor retrieval accuracy. The local minima ratio increases rapidly without IABC, suggesting the model gets trapped in suboptimal solutions. Meanwhile, the bottom right graph demonstrates that the global minima ratio reaches and maintains 100% with IABC, unlike the version without IABC, which fails to reach global optima. These results clearly highlight the effectiveness of IABC in enhancing retrieval accuracy and optimization stability. This enhanced retrieval performance ultimately supports the model in generating more reliable and diverse solutions, which is crucial for achieving better classification outcomes in the subsequent decision-making or pattern recognition tasks.

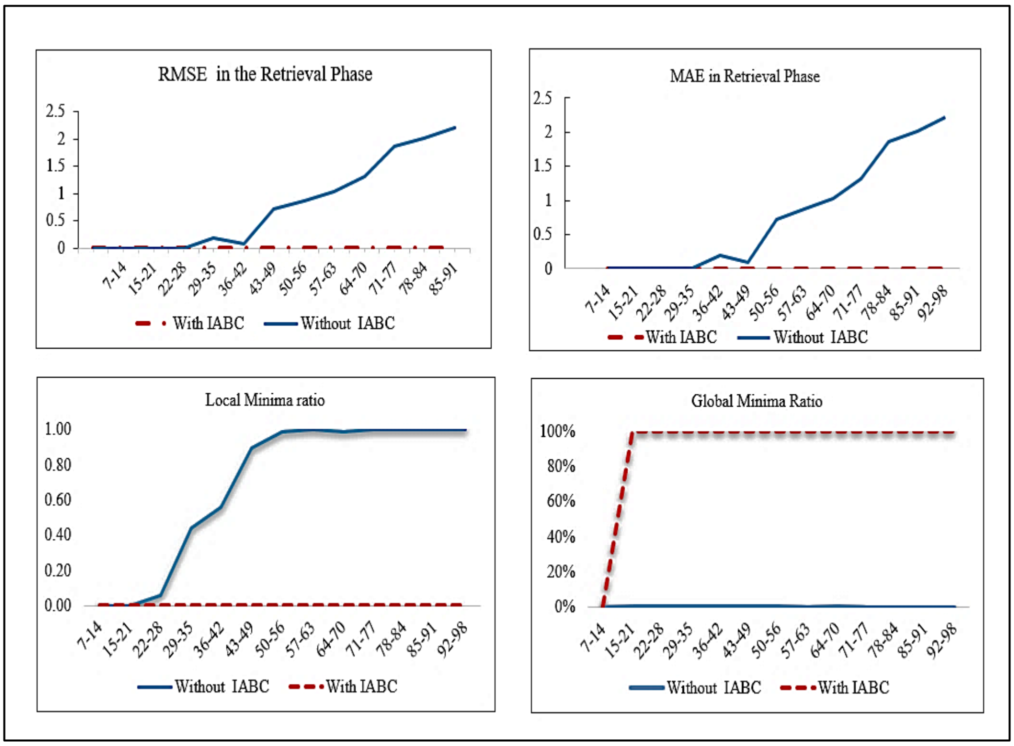

Figure 6. The performance of IABC in the retrieval phase.

9.2. *Accuracy* (*Acc*)

According to Figure 7, the MR1,3SATRAμ model achieves an accuracy of approximately 90%, demonstrating strong predictive capability in identifying depression-related AE symptoms in AD patients. This high accuracy arises from the balanced consideration of both TP and TN, allowing the induced logic to reflect meaningful symptom patterns instead of overfitting toward one class.

The higher accuracy induced logic obtained by the model is as per Eq (9.1).

$$P_{Acc}^{Induced} = (I \vee C \vee \neg G) \wedge (F \vee H \vee \neg D) \wedge (B \vee E \vee J) \wedge A. \tag{9.1}$$

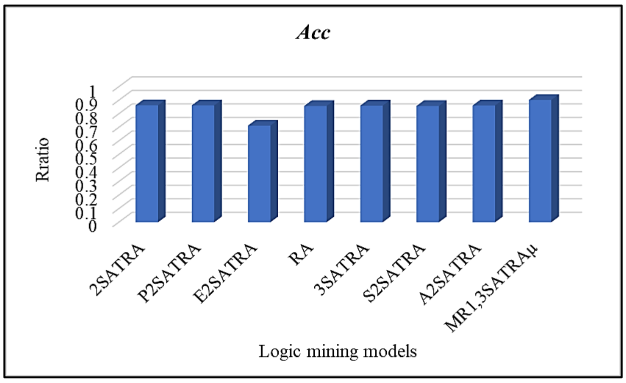

Figure 7. *Acc* outcomes across the evaluated logic mining models.

Through the proposed model and method, this study predicts depression moods related to adverse events caused by medical treatments, medications, vaccines, medical devices, or procedures, resulting in $P_{Acc}^{Induced} = 1$. Based on the study by Rizzo et al. [36], the best induced logic is the most representative logical rule extracted from the dataset, ensuring both accuracy and transparency. The selection is based on induced logic that maximizes true positive and true negative classifications in the training set while also attaining the highest accuracy on the testing set. This makes it the most reliable logical representation of the dataset's behavior, balancing both performance and interpretability. The higher accuracy of the induced logic indicates that symptoms such as crying reflect emotional distress, pain, or psychological strain experienced during adverse events.

Equation (9.1) explains the depression mood of AD during the adverse event. It is replicated as "crying, sweating, and without chest pain", "palpitations, urinary discomfort (e.g., burning), and not blurred vision", or "vomiting, other/diagnosis, and elevated mood". This rule highlights clinically relevant symptom clusters. For example, crying and sweating without chest pain, or palpitations with urinary discomfort and absence of blurred vision, strongly correlate with depressive states. These logical rules enhance interpretability and offer practical guidance for healthcare professionals in the early identification of depression. Crying behavior is frequently reported as a behavioral manifestation of depression in Alzheimer's disease, reflecting emotional dysregulation and psychological distress associated with cognitive decline [58]. Additionally, depressive and anxiety-related states are often accompanied by autonomic nervous system activation, which can manifest as physiological symptoms such as excessive sweating. Neurodegenerative changes affecting emotional and autonomic regulatory circuits may contribute to such combined behavioral and physiological symptoms. The absence of chest pain in the extracted rule further suggests that the pattern is more consistent with emotional and autonomic manifestations of depression rather than acute somatic conditions such as cardiovascular or panic-related disorders [59]. Therefore, the extracted logical clause provides an interpretable representation of symptom interactions associated with depressive mood states in the AD dataset.

9.3. Sensitivity (*Sen*)

Sen represents the proportion of actual positive cases that are correctly identified by the model. In this study, the proposed method demonstrated a *Sen* of nearly 98%, which is comparable to the 99% reported by 2SATRA. This indicates that MR1,3SATRAμ rarely misses depression cases among AD patients, making it particularly reliable for early screening and clinical decision support. The highest *Sen* performance ensures that depressive symptoms are detected immediately by minimizing false negatives and supporting timely intervention.

In Figure 8, *Sen*, also known as the true positive rate, represents the proportion of actual positive cases accurately identified by a diagnostic test or predictive model. In the context of AD dataset analysis during adverse events, achieving nearly 1 sensitivity suggests that the proposed method demonstrates an outstanding capability to detect adverse events accurately, thereby minimizing the risk of false negatives and ensuring prompt recognition of potential complications or exacerbations in AD patients.

$$P_{Sen}^{Induced} = (D \vee H \vee \neg C) \wedge (J \vee B \vee \neg I) \wedge (E \vee F \vee G) \wedge A. \tag{9.2}$$

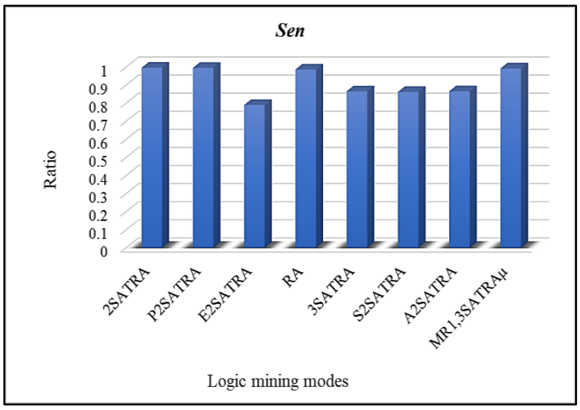

Figure 8. S*en* outcomes across the evaluated logic mining models.

This high sensitivity indicates a strong potential for early intervention and proactive management strategies, which are crucial in optimizing patient care and improving clinical outcomes in individuals with AD. By accurately identifying adverse events, healthcare practitioners can implement timely interventions such as medication adjustments or additional support services to reduce the impact of these events on cognitive function, functional status, and overall well-being in AD patients. The proposed model has a sensitivity of almost 98%. Table 6 represent performance comparison of the proposed and baseline models on the ADNI case study. The results shows that the proposed MR1,3SATRAμ model achieves the highest *Sen* value among the compared models. This result indicates that the proposed model is more effective in identifying positive instances within the dataset, demonstrating its capability in capturing relevant depression happen to the AD patients during the learning process.

Table 6. The performance of the ADNI case study for the proposed model and baseline models.

Metrics	2SATRA	P2SATRA	E2SATRA	RA	3SATRA	S2SATRA	A2SATRA	MR1,3SATRAμ
Acc	0.8631	0.8626	0.7132	0.8581	0.8608	0.8581	0.8621	**0.9023**
Sen	**0.9955**	0.9947	0.7895	0.9857	0.8650	0.8618	0.8653	0.9913
Spec	0.0244	0.0394	0.2370	0.0622	0.4663	0.4073	0.5164	**0.9888**
MCC	0.0574	0.1022	0.0583	0.1164	0.0933	0.1056	0.1041	**0.3852**
FMI	0.9285	0.9280	0.7587	0.9248	0.9270	0.9253	0.9278	**0.9346**
+/=/-	**1/0/4**	**0/0/5**	**0/0/0**	**0/0/5**	**0/0/5**	**0/0/5**	**0/0/5**	**4/0/1**
Average Rank	4.8	5.0	1.8	3.9	4.0	3.7	5.2	**7.6**
***p*-value**					3.16×10^{-5}			
Decision					Reject H_0			

9.4. *Specificity* (*Spec*)

Comparing this specificity value to the sensitivity of 90% for the proposed model suggests that the model is well-balanced correctly identify both positive (depression mood present) and negative (depression mood absent) cases. Given that the specificity for the proposed model is also 99% compared to the other model, it suggests that both models have high specificity values. Therefore, they are both effective in correctly identifying cases, where depression mood is absent shown in Eq (9.3). When comparing existing models, it is important to consider both sensitivity and specificity as well as other performance metrics to make informed decisions about their suitability for the task at hand. In this case, the high specificity of both models indicates their effectiveness in correctly classifying negative cases of depression mood. Figure 9 presents the *Spec* values of all logic mining models evaluated on the ADNI datasets. As highlighted by Singh and Singh [59], *Spec* measures the ability of a model to accurately classify negative cases. This metric is particularly important in the current study, as many of the datasets are dominated by the −1 class, with the 1 class representing the minority. High *TN* values therefore demonstrate the effectiveness of the models in addressing such class distributions even when the imbalance toward the −1 class is not consistent across all datasets.

$$P_{Spec}^{Induced} = (F \vee \neg J \vee \neg B) \wedge (D \vee \neg H \vee C) \wedge (\neg A \vee \neg E \vee \neg I) \wedge \neg G. \tag{9.3}$$

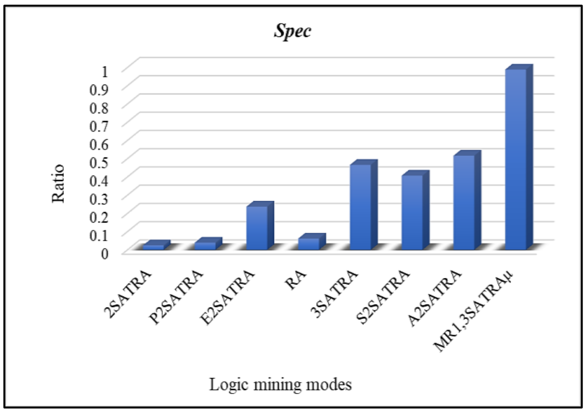

Figure 9. *Spec* outcomes across the evaluated logic mining models.

9.5. Mathews correlation coefficient

The *MCC*, which incorporates true positives, true negatives, false positives, and false negatives, provides a balanced evaluation of classification performance in imbalanced datasets, where higher values reflect greater reliability and accuracy as demonstrated when predicting depressive mood through the symptom of *AE.* This indicates that the presence of this adverse event symptom is more strongly associated with depression mood in the dataset, and the model benefits including it in its real classification task. In practical terms, a higher *MCC* means that the model accurately aligns better with the actual outcomes leading to fewer misclassifications overall. The results are a positive indication of the model effectiveness in identifying cases of depression mood, particularly when considering the specific symptoms of the adverse event. Overall, a higher *MCC* in predicting depression mood using adverse event symptoms highlights their critical role in enhancing classification accuracy and suggests important clinical implications for identifying individuals at risk or requiring intervention. Equation (9.4) explains how the proposed model obtains the higher value of *MCC* induced logic.

$$P_{MCC}^{Induced} = (\neg D \vee B \vee C) \wedge (G \vee J \vee \neg E) \wedge (F \vee \neg I \vee H) \wedge A. \tag{9.4}$$

In contrast, an overall examination of the baseline methods based on average values reveals that the *MCC* score for 2SATRA tends to be close to zero. This outcome is largely due to 2SATRA's use of a rigid, systematic logic structure that enforces a uniform variable pattern within each clause. Such structural uniformity limits the model's flexibility in capturing broader patterns within the dataset. As a result, 2SATRA struggles to generate diverse logic rules that effectively represent both positive and negative outcomes. This creates an imbalance between *TP* and *TN* classifications, which in turn results in lower *MCC* values across the majority of datasets.

Figure 10 illustrates the MCC performance across the evaluated logic mining models.

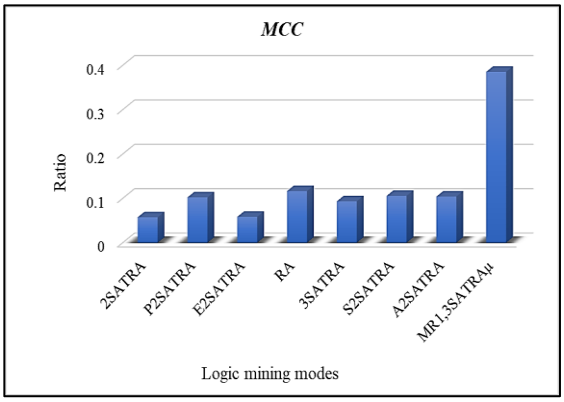

Figure 10. *MCC* outcomes across the evaluated logic mining models.

9.6. Fowlkes Mallows index

Figure 11 presents the *FMI* results of all logic mining models on the ADNI datasets. As emphasized by Shankar et al. [16], the *FMI* quantifies the correlation between predicted and actual classes thereby indicating the extent to which the induced logic reflects the true outcomes. Higher *FMI* values (approaching 1) demonstrate strong retrieval of both *TP* and *TN*, whereas lower values, nearing 0, indicate misclassifications. The *FMI* is often applied to assess logic mining models by comparing the induced logic with the ground truth outcomes, thereby reflecting how well the model captures the underlying class structure. However, *FMI* is not typically used for predicting depression mood or evaluating the relationship between depression mood and AE in datasets like ADNI. Equation (9.5) represents the highest *FMI* induced logic of MR1,3SATRAμ:

$$P_{FMI}^{Induced} = (\neg D \vee A \vee \neg G) \wedge (F \vee C \vee J) \wedge (I \vee E \vee \neg H) \wedge B. \tag{9.5}$$

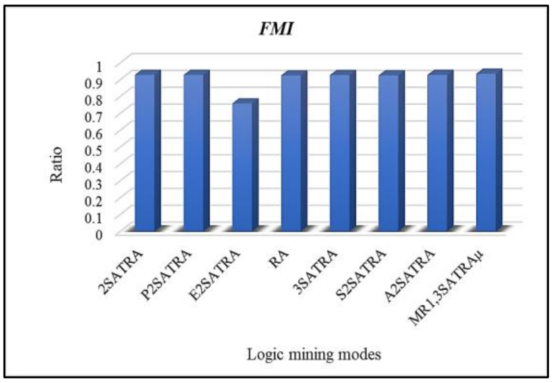

Figure 11. *FMI* outcomes across the evaluated logic mining models.

FMI values reflect the correlation between predicted and actual classes, capturing the degree of similarity between inferred logical outcomes and true outcomes. High *FMI* values approaching 1 indicate that the logic mining model effectively retrieves both *TP* and *TN*, whereas values close to 0 signify misclassification. The highest *FMI* result of 93.46% indicates a strong alignment between predicted and actual classifications of depressive mood in Alzheimer's patients. This high *FMI* score reflects the MR1,3SATRAμ model ability to correctly identify relevant patterns from the symptom attributes. The key symptoms such as crying, nausea, vomiting, and excessive sweating showed a strong association with depression mood. Although the dataset contains a higher proportion of negative (−1) instances, the proposed MR1,3SAT-based DHNN model mitigates the risk of third-order majority-class prediction through its clause-based logical representation. Instead of relying on class frequency alone, the model constructs logical clauses that capture relationships among attributes. In particular, the inclusion of higher-order clauses enables the modeling of complex symptom interactions associated with positive cases. The DHNN energy minimization process further ensures that neuron states converge toward configurations that satisfy these logical constraints. As a result, the model is able to identify meaningful patterns even under imbalanced class distributions. Furthermore, performance metrics such as the *MCC* and *FMI*, which are less sensitive to class imbalance, confirm that the proposed model maintains balanced predictive performance.

In Table 6, the performance of the ADNI dataset for the proposed model and other baseline model. Additionally, a win/equal/loss count (denoted as +/=/-) average rank and a statistical significance test (*p*-value and decision) are included to assess comparative performance. The symbols "+", "=", and "−" in Table 6 represent the comparative performance of the proposed model relative to the benchmark models across different evaluation metrics. Specifically, "+" indicates that the proposed model achieves better performance than the compared method, "=" indicates equivalent performance, and "−" indicates inferior performance. This notation provides a concise summary of the overall comparative performance across multiple experimental configurations. The "+/=/−" comparison (4/0/1) suggests that MR1,3SATRAμ outperformed four models, equal with none model and loses with one model (S2SATRA) demonstrating strong competitiveness. Furthermore, it achieved the best average rank (7.6) across all baseline models. The obtained p-value (3.16×10^{-5}) indicates strong statistical significance, thereby supporting the rejection of the null hypothesis.

10. Limitations and future research directions

Despite the promising performance of the proposed MR1,3SAT-based logic mining model, several limitations should be acknowledged. First, the incorporation of higher-order logical clauses within the DHNN increases the computational complexity of the energy evaluation process. As the number of neurons and logical clauses grows, the computational cost associated with evaluating higher-order interactions may increase. Second, scalability may become challenging when the model is applied to datasets with extremely large feature spaces. Although the Jaccard similarity analysis helps reduce redundant attributes, additional dimensionality reduction techniques may be required for very high-dimensional datasets. Third, while logic mining improves interpretability by generating symbolic logical rules, interpretability may still become complex when a large number of clauses are induced. Future research could explore rule simplification strategies to further improve the clarity of extracted knowledge. Finally, although the model incorporates optimization mechanisms to improve generalization performance, the possibility of overfitting cannot be completely eliminated when complex logical structures are learned from limited training data. Future work may investigate regularization strategies or cross-dataset validation to further enhance robustness.

11. Conclusions

The proposed MR1,3SATRAμ method demonstrated superior performance over existing approaches, particularly in managing the complexity and high dimensionality of the ADNI dataset especially when dealing with AE data comprising diverse symptoms and complications. This advancement is notable given the challenges in designing studies to evaluate disease-modifying treatments for AD especially in elderly populations with multiple medical comorbidities. Accurately estimating AE rates requires attention to classification nuances, such as distinguishing between types of rashes or varying degrees of depression. In analyzing AE rates, where symptoms like dyspnea, headache, nausea, and anxiety consistently occurred below a 93% threshold, reliable benchmarks for AD related depression symptoms were established. This study makes three significant contributions to understanding depressive mood through AE symptom analysis. First, it introduces a refined logic mining approach utilizing data preprocessing and Jaccard analysis to identify predictive attributes linked to AD development, applying MR1,3SAT and DHNN to uncover and interpret the logic relationships. Second, it links the induced logic to AE symptoms particularly for depression diagnosis by analyzing connections between selected attributes and their symptomatic expressions, further enhanced through intelligent mutation analysis. Third, the study proposes practical methods for early diagnosis, including the identification of potential biomarkers and the integration of genetic information to refine classification models. These contributions offer actionable insights for various stakeholders, such as for AD patients, and the study highlights the importance of early intervention to improve quality of life and ease family burdens. For clinicians it provides diagnostic tools to support personalized care and early detection whereby for neurologists, and for psychiatrists and researchers it delivers advanced methodologies for identifying predictors of AD, enriching both clinical practice and scientific inquiry. Ultimately, this study presents a novel and comprehensive approach to improving AD diagnosis and treatment, underscoring its value across medical and research domains.

Author contributions

G. Manoharam: Conceptualization, Project administration, Writing–original draft; M. S. M. Kasihmuddin: Resources, Funding acquisition; S. Abdeen: Visualization; N. A. Romli and N. Roslan: Formal analysis, Investigation; N. 'A. Rusdi: Methodology; M. A. Mansor: Writing–review & editing; N. E. Zamri: Supervision, Validation; Xiaoyan Liu: Software. 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.

Conflict of interest

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

References

1. S. E. Sorour, A. A. A. El-Mageed, K. M. Albarrak, A. K. Alnaim, A. A. Wafa, E. El-Shafeiy, Classification of Alzheimer's disease using MRI data based on deep learning techniques, *J. King Saud Univ. Comput. Inform. Sci.*, **36** (2024), 101940. https://doi.org/10.1016/j.jksuci.2024.101940
2. M. T. Heneka, W. M. Van der Flier, F. Jessen, J. Hoozemanns, D. R. Thal, D. Boche, et al., Neuroinflammation in Alzheimer disease, *Nat. Rev. Immunol.*, **25** (2025), 321–352. https://doi.org/10.1038/s41577-024-01104-7
3. M. Odusami, R. Maskeliūnas, R. Damaševičius, Pareto optimized adaptive learning with transposed convolution for image fusion Alzheimer's disease classification, *Brain Sci.*, **13** (2023), 1045. https://doi.org/10.3390/brainsci13071045
4. T. Prasath, V. Sumathi, Pipelined deep learning architecture for the detection of Alzheimer's disease, *Biomed. Signal Process. Control*, **87** (2024), 105442. https://doi.org/10.1016/j.bspc.2023.105442
5. S. L. Burke, T. Cadet, A. Alcide, J. O'Driscoll, P. Maramaldi, Psychosocial risk factors and Alzheimer's disease: the associative effect of depression, sleep disturbance, and anxiety, *Aging Mental Health*, **22** (2018), 1577–1584. https://doi.org/10.1080/13607863.2017.1387760
6. A. Hammar, E. H. Ronold, G. Å. Rekkedal, Cognitive impairment and neurocognitive profiles in major depression–a clinical perspective, *Front. Psychiatry*, **13** (2022), 764374. https://doi.org/10.3389/fpsyt.2022.764374
7. S. G. Potkin, The ABC of Alzheimer's disease: ADL and improving day-to-day functioning of patients, *Int. Psychogeriatrics*, **14** (2002), 7–26. https://doi.org/10.1017/S1041610203008640

8. S. A. Alowais, S. S. Alghamdi, N. Alsuhebany, T. Alqahtani, A. I. Alshaya, S. N. Almohareb, et al., Revolutionizing healthcare: the role of artificial intelligence in clinical practice, *BMC Med. Educ.*, **23** (2023), 689. https://doi.org/10.1186/s12909-023-04698-z
9. A. Baba, Neural networks from biological to artificial and vice versa, *Biosystems*, **235** (2024), 105110. https://doi.org/10.1016/j.biosystems.2023.105110
10. A. Krenker, J. Bešter, A. Kos, Introduction to the artificial neural networks, In: *Artificial neural networks: Methodological advances and biomedical applications*, London, UK: IntechOpen, 2011, 1–18.
11. J. J. Hopfield, D. W. Tank, "Neural" computation of decisions in optimization problems, *Biol. Cybernet.*, **52** (1985), 141–152. https://doi.org/10.1007/BF00339943
12. C. P. He, M. R. Jiang, K. Y. Shan, S. H. Yang, Z. F. Li, S. B. Wang, et al., A hardware-adaptive learning algorithm for superlinear-capacity associative memory on memristor crossbars, *Nat Commun*., **17** (2026), 3096. https://doi.org/10.1038/s41467-026-69958-0
13. Z. Q. Yu, A. M. Abdulghani, A. Zahid, H. Heidari, M. A. AImran, Q. H. Abbasi, An overview of neuromorphic computing for artificial intelligence enabled hardware-based Hopfield neural network, *IEEE Access*, **8** (2020), 67085–67099. https://doi.org/10.1109/ACCESS.2020.2985839
14. H. Taherdoost, Deep learning and neural networks: decision-making implications, *Symmetry*, **15** (2023), 1723. https://doi.org/10.3390/sym15091723
15. V. L. Kalmykov, L. V. Kalmykov, Towards explicitly explainable artificial intelligence, *Inform. Fusion*, **123** (2025), 103352. https://doi.org/10.1016/j.inffus.2025.103352
16. M. G. Shankar, C. G. Babu, H. Rajaguru, Classification of cardiac diseases from ECG signals through bio inspired classifiers with Adam and R-Adam approaches for hyperparameters updation, *Measurement*, **194** (2022), 111048. https://doi.org/10.1016/j.measurement.2022.111048
17. A. Chinnaraju, Explainable AI (XAI) for trustworthy and transparent decision-making: a theoretical framework for AI interpretability, *World J. Adv. Eng. Technol. Sci.*, **14** (2025), 170–207. https://doi.org/10.30574/wjaets.2025.14.3.0106
18. X. F. Jiang, M. S. M. Kasihmuddin, Y. L. 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. Intell.*, **17** (2024), 3317–3336. https://doi.org/10.1007/s12065-024-00936-5
19. W. A. T. W. Abdullah, Logic programming on a neural network, *Int. J. Intell. Syst.*, **7** (1992), 513–519. https://doi.org/10.1002/int.4550070604
20. S. Sathasivam, Upgrading logic programming in Hopfield network, *Sains Malaysiana*, **39** (2010), 115–118.
21. M. S. M. Kasihmuddin, M. A. Mansor, S. Sathasivam, Hybrid genetic algorithm in the Hopfield network for logic satisfiability problem, *Pertanika J. Sci. Technol.*, **25** (2017), 139–152.
22. M. A. Mansor, M. S. M. Kasihmuddin, S. Sathasivam, Artificial immune system paradigm in the Hopfield network for 3-satisfiability problem, *Pertanika J. Sci. Technol*, **25** (2017), 1173–1188.
23. S. Sathasivam, M. A. Mansor, A. I. M. Ismail, S. Z. M. Jamaludin, M. S. M. Kasihmuddin, M. Mamat, Novel random k satisfiability for $k \leq 2$ in Hopfield neural network, *Sains Malaysiana*, **49** (2020), 2847–2857. https://doi.org/10.17576/jsm-2020-4911-23
24. S. A. Karim, N. E. Zamri, A. Alway s, M. S. M. Kasihmuddin, A. I. M. Ismail, M. A. Mansor, Random satisfiability: a higher-order logical approach in discrete Hopfield neural network, *IEEE Access*, **9** (2021), 50831–50845. https://doi.org/10.1109/ACCESS.2021.3068998

25. 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://doi.org/10.1080/00207160.2021.1939870
26. S. Sathasivam, W. A. T. W. Abdullah, Logic mining in neural network: reverse analysis method, *Computing*, **91** (2011), 119–133. https://doi.org/10.1007/s00607-010-0117-9
27. L. C. Kho, M. S. M. Kasihmuddin, M. A. Mansor, S. Sathasivam, Logic mining in league of legends, *Pertanika J. Sci. Technol.*, **28** (2020), 211–225.
28. A. Alway, N. E. Zamri, M. S. M. Kasihmuddin, M. A. Mansor, S. Sathasivam, Palm oil trend analysis via logic mining with discrete Hopfield neural network, *Pertanika J. Sci. Technol.*, **28** (2020), 967–981.
29. N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, A. Always, S. Z. M. Jamaludin, S. A. Alzaeemi, Amazon employees resources access data extraction via clonal selection algorithm and logic mining approach, *Entropy*, **22** (2020), 596. https://doi.org/10.3390/e22060596
30. S. Z. M. Jamaludin, M. S. M. Kasihmuddin, A. I. M. Ismail, M. A. Mansor, M. F. M. Basir, Energy based logic mining analysis with Hopfield neural network for recruitment evaluation, *Entropy*, **23** (2021), 40. https://doi.org/10.3390/e23010040
31. 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. Continua*, **74** (2023), 2853–2870. https://doi.org/10.32604/cmc.2023.032654
32. M. S. M. Kasihmuddin, S. Z. M. Jamaludin, M. A. Mansor, H. A. Wahab, S. M. S. Ghadzi, Supervised learning perspective in logic mining, *Mathematics*, **10** (2022), 915. https://doi.org/10.3390/math10060915
33. 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. Inform. Sci.,* **34** (2022), 9011–9027. https://doi.org/10.1016/j.jksuci.2022.08.026
34. 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, *Decision Anal. J.*, **9** (2023), 100354. https://doi.org/10.1016/j.dajour.2023.100354
35. 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://doi.org/10.1016/j.eswa.2023.122307
36. L. Rizzo, D. Verda, S. Berretta, L. Longo, A novel integration of data-driven rule generation and computational argumentation for enhanced explainable AI, *Mach. Learn. Knowl. Extr.*, **6** (2024), 2049–2073. https://doi.org/10.3390/make6030101
37. A. Q. Tian, F. F. Liu, H. X. Lv, Snow Geese algorithm: a novel migration-inspired meta-heuristic algorithm for constrained engineering optimization problems, *Appl. Math. Model.*, **126** (2024), 327–347. https://doi.org/10.1016/j.apm.2023.10.045
38. M. L. Zhao, S. Q. Ni, Z. G. Du, X. Y. Wang, A. Q. Tian, X. L. Ma, Multi-objective gannet optimization algorithm for dynamic passenger flow allocation in train operation plan optimization, *IEEE Access*, **11** (2023), 103693–103711. https://doi.org/10.1109/ACCESS.2023.3318262

39. A. Q. Tian, J. S. Pan, H. X. Lv, Optimizing train scheduling in heavy-haul railways using diversified cooperative deep reinforcement learning, *Transport. Res. Record*, **2680** (2026), 286–312. https://doi.org/10.1177/03611981251364832
40. Q. Lai, M. H. Qin, Universal method for enhancing dynamics in neural networks via memristor and application in IoT-based robot navigation, *IEEE Trans. Cybernet.*, **56** (2026), 557–566, https://doi.org/10.1109/TCYB.2025.3607140
41. Q. Lai, Y. D. Xu, L. Fortuna, Generating simple cyclic memristive neural network circuit with controllable multiscroll attractors and multivariable amplitude control, *IEEE Trans. Neural Networks Learn. Syst.*, **36** (2025), 18805–18814. https://doi.org/10.1109/TNNLS.2025.3581229
42. H. Zhou, D. C. Ren, H. X. Xia, M. Y. Fan, X. Yang, H. Huang, Ast-gnn: an attention-based spatio-temporal graph neural network for interaction-aware pedestrian trajectory prediction, *Neurocomputing*, **445** (2021), 298–308. https://doi.org/10.1016/j.neucom.2021.03.024
43. M. Y. Fan, X. Q. Zhang, J. Hu, N. N. Gu, D. C. Tao, Adaptive data structure regularized multiclass discriminative feature selection, *IEEE Trans. Neural Networks Learn. Syst.*, **33** (2022), 5859–5872. https://doi.org/10.1109/TNNLS.2021.3071603
44. X. J. Li, H. Huang, H. L. Zhao, Y. D. Wang, M. X. Hu, Learning a convolutional neural network for propagation-based stereo image segmentation, *Vis. Comput.*, **36** (2020), 39–52. https://doi.org/10.1007/s00371-018-1582-y
45. 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. Inform. Sci.*, **34** (2022), 9011–9027. https://doi.org/10.1016/j.jksuci.2022.08.026
46. 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 Math.*, **9** (2024), 12090–12127. https://doi.org/10.3934/math.2024591
47. P. Foldiak, Forming sparse representations by local anti-Hebbian learning, *Biol. Cybernet.*, **64** (1990), 165–170. https://doi.org/10.1007/BF02331346
48. W. A. T. W. Abdullah, The logic of neural networks, *Phys. Lett. A*, **176** (1993), 202–206. https://doi.org/10.1016/0375-9601(93)91035-4
49. M. B. Muhammad, M. Yeasin, Eigen-CAM: visual explanations for deep convolutional neural networks, *SN Comput. Sci.*, **2** (2021), 47. https://doi.org/10.1007/s42979-021-00449-3
50. M. S. M. Kasihmuddin, M. A. Mansor, M. D. Basir, S. Sathasivam, Discrete mutation Hopfield neural network in propositional satisfiability, *Mathematics*, **7** (2019), 1133. https://doi.org/10.3390/math7111133
51. S. Sathasivam, M. A. Mansor, M. S. M. Kasihmuddin, H. Abubakar, Election algorithm for random k satisfiability in the Hopfield neural network, *Processes*, **8** (2020), 568. https://doi.org/10.3390/PR8050568
52. J. Choi, O. M. Dekkers, S. le Cessie, Comparison of different methods to handle missing data in the context of propensity score analysis, *Eur. J. Epidemiology*, **34** (2019), 23–36. https://doi.org/10.1007/s10654-018-0447-z
53. H. L. Vu, K. T. W. Ng, A. Richter, C. J. An, Analysis of input set characteristics and variances on k-fold cross validation for a recurrent neural network model on waste disposal rate estimation, *J. Environ. Manag.*, **311** (2022), 114869. https://doi.org/10.1016/j.jenvman.2022.114869
54. Z. A. Sejuti, M. S. Islam, A hybrid CNN-KNN approach for identification of COVID-19 with 5-fold cross validation, *Sensors Int.*, **4** (2023), 100229. https://doi.org/10.1016/j.sintl.2023.100229

55. D. Chicco, G. Jurman, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, *BMC Genomics*, **21** (2020), 6. https://doi.org/10.1186/s12864-019-6413-7
56. A. L. Teixeira, 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://doi.org/10.1016/j.patcog.2019.02.023
57. K. Riehl, M. Neunteufel, M. Hemberg, Hierarchical confusion matrix for classification performance evaluation, *J. Roy. Statist. Soc. Ser. C*, **72** (2023), 1394–1412. https://doi.org/10.1093/jrsssc/qlad057
58. N. C. Tunnell, S. E. Corner, A. D. Roque, J. L. Kroll, T. Ritz, A. E. Meuret, Biobehavioral approach to distinguishing panic symptoms from medical illness, *Front. Psychiatry*, **15** (2024), 1296569. https://doi.org/10.3389/fpsyt.2024.1296569
59. N. Singh, P. Singh, A hybrid ensemble-filter wrapper feature selection approach for medical data classification, *Chem. Intell. Lab. Syst.*, **217** (2021), 104396. https://doi.org/10.1016/j.chemolab.2021.104396