



Research article

Unsupervised logic mining with a binary clonal selection algorithm in multi-unit discrete Hopfield neural networks via weighted systematic 2 satisfiability

Nurul Atiqah Romli¹, Nur Fariha Syaquina Zulkepli^{1,*}, Mohd Shareduwan Mohd Kasihmuddin¹, Nur Ezlin Zamri², Nur 'Afifah Rusdi^{1,3}, Gaeithry Manoharam¹, Mohd. Asyraf Mansor⁴, Siti Zulaikha Mohd Jamaludin¹ and Amierah Abdul Malik⁴

¹ School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

² Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

³ Institute of Engineering Mathematics, Universiti Malaysia Perlis, Kampus Pauh Putra, 02600 Arau, Perlis, Malaysia

⁴ School of Distance Education, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

* **Correspondence:** Email: farihasyaqina@usm.my; Tel: +6046532671.

Abstract: Evaluating behavioral patterns through logic mining within a given dataset has become a primary focus in current research. Unfortunately, there are several weaknesses in the research regarding the logic mining models, including an uncertainty of the attribute selected in the model, random distribution of negative literals in a logical structure, non-optimal computation of the best logic, and the generation of overfitting solutions. Motivated by these limitations, a novel logic mining model incorporating the mechanism to control the negative literal in the systematic Satisfiability, namely Weighted Systematic 2 Satisfiability in Discrete Hopfield Neural Network, is proposed as a logical structure to represent the behavior of the dataset. For the proposed logic mining models, we used ratio of r to control the distribution of the negative literals in the logical structures to prevent overfitting solutions and optimize synaptic weight values. A new computational approach of the best logic by considering both true and false classification values of the learning system was applied in this work to preserve the significant behavior of the dataset. Additionally, unsupervised learning techniques such as Topological Data Analysis were proposed to ensure the reliability of the selected attributes in the model. The comparative experiments of the logic mining models by utilizing 20 repository real-life

datasets were conducted from repositories to assess their efficiency. Following the results, the proposed logic mining model dominated in all the metrics for the average rank. The average ranks for each metric were Accuracy (7.95), Sensitivity (7.55), Specificity (7.93), Negative Predictive Value (7.50), and Mathews Correlation Coefficient (7.85). Numerical results and in-depth analysis demonstrated that the proposed logic mining model consistently produced optimal induced logic that best represented the real-life dataset for all the performance metrics used in this study.

Keywords: discrete Hopfield neural network; improved logic mining; binary clonal selection algorithm; weighted systematic 2 satisfiability; topological data analysis

Mathematics Subject Classification: 68N17, 68R07, 68T27

1. Introduction

In this era, the strategy or approach for analyzing data is abundant. Logic mining is one of the strategies that is relatively new with the aim to extract patterns and determine the overall behavior of a data. This particular strategy composed from several components, which are logical rule, Artificial Neural Network (ANN), and reverse-analysis method. The basis of logic mining is quite straightforward. Logical rule will be the form of symbolic representation that exploits the structure of the data via reverse-analysis method. The choice of ANN is the medium of optimization to ensure the logical rule is being effectively learn and produces induced logic as the final output of the network to classify new records with a good level of accuracy. As a whole, these components are dependent on each other, and if any one of the components is missing, the performance of the logic mining will be negatively affected. In this context, it is crucial to not pick one over the other. Therefore, more investigation should be carried out to identify what improvements can be made to the existing logic mining to ensure optimal classification tasks can be executed.

Over the years, Discrete Hopfield Neural Network (DHNN) has showed credible results as one of the components in logic mining. Introduced by Hopfield and Tank [1], the network is classified as one of the symmetric networks with feedback mechanism. Although the proposed DHNN exhibit good performance in optimizing the travelling salesman problem (TSP), the issue of black-box nature of the model is poorly addressed. Abdullah [2] attempted to address this issue by implementing Horn Satisfiability (HornSAT) to governs the neurons in DHNN. The integration of HornSAT into DHNN can be verified using the proposed Wan Abdullah (WA) method. The work showed notable findings, however, due to structural components of HornSAT which comprises of redundant variables, the output of the DHNN became insignificant. Therefore, Kasihmuddin [3] addresses the previous work by incorporating non-redundant logical rule of 2 Satisfiability (2SAT) as symbolic rule in DHNN. Due to the nondeterministic polynomial (NP) problem property of the DHNN-2SAT model, the work capitalizes Genetic Algorithm (GA) to aid the network in learning the logical rule. As the number of neurons increases, the GA is able to locate the satisfied interpretations of the logical rule, which resulted in correct synaptic weight values. As compared to Exhaustive Search (ES), the proposed GA attained acceptable results with almost 90% global minima energy solutions. This work has brought attention to other researchers to capitalizes other evolutionary and swarm-based algorithms as the learning algorithm to learn Satisfiability (SAT) logical rule in DHNN. As an example, the work by Mansor et al. [4] extended the previous work by implementing Artificial Immune System (AIS) in the

learning phase of DHNN. The proposed AIS in DHNN attained high accumulation of global minima solutions with very minimal learning errors. Subsequently, Sathasivam et al. [5] proposed the Election Algorithm (EA) in the learning phase of DHNN to learn Random 2 Satisfiability (RAN2SAT) logical rule. The proposed EA possessed multiple global and local search operators which is beneficial for the network in locating interpretation even when the number of neurons is high. The implementation of EA has brought new level in optimizing logical rule with high efficiency. However, all the mentioned works focused only on improving the learning operations in DHNN. Little attention was given towards improving the retrieval phase of DHNN.

The pioneer work on integrating the concept of satisfiability in DHNN was proposed by Abdullah [2]. This innovative approach employs HornSAT to characterize the neurons within the DHNN. Additionally, this work also laid to the formulation of WA method whereby the minimization of the cost function can be derived based on the inconsistency of the logical rule. In this context, the synaptic weight can be obtained by comparing the cost function and the Lyapunov energy function. The successful incorporation of HornSAT into DHNN marked a significant advancement in the field, inspiring subsequent contributions by Kasihmuddin et al. [3] and Mansor et al. [4]. However, the restriction of the literals in each clause imposes limitations on the flexibility of logical rules, leading to potential overfitting issues. Addressing these constraints, Sathasivam et al. [6] proposed a novel approach by introducing the first non-systematic logic focusing on first and second-order clauses. The incorporation of different order logics introduces variation in synaptic weight values, resulting in more diverse solutions. The advantages offered by non-systematic logic have led to a substantial increase in the number of SAT formulations. Karim et al. [7] built upon the foundation laid by Sathasivam et al. [6] by introducing higher order Random k Satisfiability (RAN k SAT) which capitalize first, second and third-order clauses. Besides that, Guo et al. [8] delved into the capabilities of systematic and non-systematic logical rules, introducing a novel variant of SAT known as Y-Type Random 2 Satisfiability (YRAN2SAT). This variant concentrates on first and second-order clauses, embodying both systematic and non-systematic structures under the formulation of SAT. The integration of YRAN2SAT into DHNN has been reported to successfully retrieve final neuron states with a high degree of diversity compared to all existing models. In another work, Zamri et al. [9] proposed a new perspective on SAT by focusing on the distribution of negative literals within the clauses. In this study, another layer was introduced into the DHNN which is specifically to determine the distribution of negative literals. Amplifying the presence of negative literals is crucial for gaining a deeper understanding of the negative links within neuron connections and contributes to exploring the alternative optimal final neuron state with more diversified solutions. While all the SAT formulations mentioned contribute to more positive results, the application of SAT into the DHNN in the context of logic mining remains uncertain.

Logic mining is a subset of data mining which was specialized in addressing classification problems. This logic mining approach aims to extract the knowledge and behavior of the datasets. The earliest work on logic mining was proposed by Sathasivam and Abdullah [10] by introducing Reverse Analysis (RA) method. The RA method aims to represent students' performance in each subject and describe the datasets. However, a major drawback of this approach is that it can lead to an infinite number of induced logics, raising questions about its capability to effectively represent patterns in the dataset. To address this problem, Kho et al. [11] proposed the 2 Satisfiability Reverse Analysis method (2SATRA) based on the 2SAT logical rule. This approach is capable of extracting the best single induced logic for classifying dataset behavior in League of Legends (LoL) games. Moreover, Zamri et al. [12] introduced a higher-order logic mining model known as 3SATRA, which utilizes the 3SAT

logical rule and employs the Clonal Selection Algorithm (CSA) as a learning algorithm. The 3SATRA model extracts optimal induced logic from data related to Amazon employees' resource access. Despite both models using random attribute selection, they were able to produce high-quality induced logic with a high level of accuracy.

In another study of systematic logical rule, Jamaludin et al. [13] conducted a study on systematic logical rules and proposed the Energy-based k Satisfiability Reverse Analysis (EkSATRA) method as an alternative approach to extract the correct recruitment factors that contribute to positive recruitment in a Malaysian insurance company. This method is based on 2SAT and 3SAT logical rules. During the retrieval phase of DHNN, an energy operator was utilized to transform final neuron states with global minimum energy into induced logic. EkSATRA achieved optimal best induced logic retrieval with an accuracy of 63.3% from the e-recruitment dataset. However, a limitation in all these studies is the arrangement of attributes within the logical rule structure. The standard attribute arrangement leads to low accuracy values due to limited connectivity among the attributes and reduced interpretability from randomized attribute selection. Jamaludin et al. [14] propose P2SATRA, a permutation operator in 2SATRA that expands solution space in finding the best induced logic. P2SATRA outperforms existing logic mining in various performance metrics. Despite utilizing permutation between attributes, P2SATRA does not employ any feature selection method before embedding data entries as neurons in DHNN, leading to learning with insignificant attributes and resulting in high redundancy and low relevancy classification. Kasihmuddin et al. [15] introduced the first Supervised Logic Mining model (S2SATRA), which included a statistical analysis using correlation to select the best attributes for representing the datasets. This approach removed unnecessary attributes before embedding into DHNN and outperformed existing logic mining models in terms of accuracy and precision. Another supervised learning approach for selecting optimal attributes was proposed by Jamaludin et al. [16] with a log linear analysis (A2SATRA) aimed at countering the superiority of S2SATRA in identifying significant attributes contributing to optimal quality and production of induced logic.

In addition, Rusdi et al. [17] and Manoharam et al. [18] have also utilized correlation and log linear for attribute selection during the pre-processing phase. These studies proposed a multi-unit 3SATRA in DHNN that introduces the best logic during the learning phase by considering the highest true positive and true negative outcomes. As a result, more induced logic will be produced during the retrieval phase, increasing the probability of retrieving optimal induced logic. Their findings statistically prove that the proposed model is more effective than existing logic mining models. Even though the induced logic produced during the retrieval phase has been improved, the learning phase is not yet optimal. As a result, current research in logic mining has extended to utilizing a metaheuristic algorithm to enhance the learning phase of DHNN. Additionally, Alway et al. [19] proposed a new logic mining model with a learning algorithm called Hybrid Exhaustive Search (2HESRA). The goal is to expand the search space and select the best logic based on the highest summation of true positives and true negatives. Furthermore, correlation analysis has been used to select significant attributes for dataset representation and the induced logic of 2HESRA is based on achieving the highest accuracy. The logic mining model proposed by Zamri et al. [20] combines the logic mining model with a multi-objective learning algorithm called Modified Niche Genetic Algorithm (r 2SATMRA). The existing RA method was also modified by changing the mechanism of finding best logic that known as super logic and similarity indexes were considered to investigate the relevancy of selected attributes. The statistical analysis showed significant performance for all metrics and demonstrated the superiority of r 2SATMRA over the existing logic mining model. Previous research has shown that the performance

of the logic mining model is influenced by factors such as data preparation and selecting the best logic during optimal learning. More attention should be given in the learning phase of logic mining to ensure the expansion of search space that can improved the quality of the retrieve induced logic.

In light of all the above challenges, we proposed a novel logic mining model based on an unsupervised feature selection method, a new perspective of finding best logic, and a multi-objective retrieval method. Thus, the contributions of this paper are listed as follows:

- (1) To propose Weighted systematic 2 Satisfiability logical rule as the logical structure of the logic mining model. A weighted feature is implemented in Weighted Systematic 2 Satisfiability to control the distribution of negative literals with defining ratio. By capitalizing the formulated Weighted Systematic 2 Satisfiability, the network produced more diversification of solutions.
- (2) To propose an alternative attribute selection method using an unsupervised approach. Topological data analysis is an unsupervised approach used for the attribute selection method. The proposed method capitalizes on the similarity behavior among attributes using clustering analysis.
- (3) To propose a new computation of best logic using the implementation of new objective function during the pre-processing phase of the logic mining. In this context, the new best logic with pre-defined weighted values will consider both true and false classification of the learning data.
- (4) To propose a binary Clonal Selection Algorithm as a retrieval algorithm which have high fitness and diversity. The fitness of the final solution highlights the satisfiable property of Weighted Systematic 2 Satisfiability which is directly mapped to the global minima solution. The diversity is based on the satisfied final solution with the highest distribution of negative states.
- (5) To evaluate the performance of the proposed logic mining model in extracting various real-life datasets. The performance of the proposed logic mining will be tested using various metrics and will be compared with other existing state-of-the-art logic mining.

In this paper, the organization of the section is presented as follows: The contributions of this paper are supported by the motivation and related works involved which are explained in Section 2. In Section 3, we describe the logical representation used in the proposed logic mining model. Then, in Section 4, we discuss the implementation of the logical structure in the DHNN, and we illustrate the retrieval algorithm used in DHNN in Section 5. In Section 6, we explain the process of the proposed logic mining model, and we focus on the experimental framework of the proposed logic mining model in Section 7. Following this, discuss the findings attained by all the logic mining models in Section 8. Last, in Section 9, we conclude our findings.

2. Motivation

There are several limitations in the current logic mining model that allow us to improve the effectiveness of the current logic mining model. First, the logic applied in the current logic mining ignore the distribution of the negative literal. Second, the computation of finding the best logic in the learning phase is focused only on the frequency of True Positive. Despite of that, there is no effort in the current logic mining model to expand the solution space of the network. Therefore, this section will describe in detail the needs to cater the limitations in the current logic mining model.

2.1. Ineffective structure of the logical representation

The study on SAT structure in DHNN model has increased exponentially. There are many works addressing the issue on which SAT actually has the ability to govern the neurons in DHNN. Notably, this has resulted in two major domains which are systematic SAT and non-systematic SAT in DHNN. Initially, Kasihmuddin et al. [3] proposed systematic SAT of 2SAT as the neuron representation in DHNN. Although the work showed credible results, the generalizability of the work is questionable. The variables of the 2SAT logical rule was generated at random whereby there is no distinction between the positive and negative literals. By capitalizing WA method, more positive variables definitely directed towards the positive magnitude of the synaptic weight values, whereas a high number of negative variables will direct towards the negative magnitude of the synaptic weight values. With no effort in investigating the impact of either more positive or negative variables to the neuron links, this leads to poor understanding of the proposed 2SAT as logical representation to the DHNN. Until recently, Zamri et al. [9] proposed non-systematic SAT of Weighted Random 2 Satisfiability (*r*2SAT) in DHNN to address the issue encountered by the previous work. The proposed *r*2SAT capitalized the non-systematic SAT structure by incorporating first-order and second-order clauses in the logical rule. This effort was to counter the rigidity issue of the previous 2SAT. Additionally, the work analyzes different distribution of negative variables in the logical rule within the range of 10% to 90% as compared to the positive variables. In order to effectively generate the logical rule, a logic phase was introduced. As opposed to other logical rules, the proposed *r*2SAT showed the best performance in terms of neuron variability which is a crucial indicator to identify whether DHNN produces overfitting solutions or not. Unfortunately, there are two major drawbacks. First, the existence of first-order clauses in the logical rule increases the possibility of suboptimal synaptic weight management. Particularly when the number of neurons is high, DHNN often unable to locate the satisfied interpretation of the logical rule and resulted in random valued synaptic weight which disrupts the overall quality of the retrieved final neuron states. Second, the proposed logic phase added another processing element in the existing DHNN which makes the network more complex. The logic phase has to be operated iteratively until the right structure of *r*2SAT is generated. This will contribute to high computational cost. In addition, little attention was given in the development of formulating systematic logic with additional features to control the distribution of positive and negative literals before being encoded as an symbolic rule. Hence, it is imperative to produce a logical rule without any regards to the first-order clauses with generalizability property of different distribution of negative and positive variables in the logical rule. Therefore, we introduced a new logical rule namely Weighted Systematic 2 Satisfiability logical rule with a weighted feature to control the distribution of the negative literals.

2.2. Ineffective computation of the best logic

One of the crucial aspects in logic mining involves the selection of the best logic. This is because the best logic will represent the datasets and will be learned by the network to obtained induced logic. Therefore, it is important to have an optimal best logic which can influence the quality of retrieved induced logic. Most previous studies on logic mining solely focused on determining the best logic based on the true positive outcome. The clauses with the highest frequency of learning data with positive outcomes are selected, and the combination of these clauses generates the best logic. However,

a significant challenge with this approach arises when dealing with imbalanced datasets, where there might be a majority of positive outcomes or a majority of negative outcomes. In this case, applying the conventional approach on finding the best logic may lead to bias solutions. To encounter this issue, Alway et al. [19] and Zamri et al. [20] proposed the new approach of finding the best logic by considering both positive and negative outcomes in the learning data. In this approach, the best logic is selected based on the highest summation of the positive and negative outcomes with respect to the generated logical rule. However, a primary concern is that relying on a single best logic may result in retrieving only a singular induced logic. To overcome this limitation, Manoharam et al. [18] and Rusdi et al. [17] proposed multi-unit DHNN. The main idea of this approach is to expand the search space in finding the best induced logic. While both studies utilize ten best logics, the approach involves solely focusing on the summation of outcomes. This approach may lead to an overfitting issue, potentially generating redundant induced logic. Motivated by these considerations, we assert that incorporating a broader set of objective functions in finding the best logic would provide multiple perspectives on the data. In this case, the sets of induced logic obtained may extract different knowledge from the dataset and result to a more comprehensive understanding of the dataset as well as more robust logic mining results. Therefore, we propose multiple best logic that considers both true and false classification of learning data. Each best logic focuses on the highest value of performance evaluation with respect to the learning data. The performance evaluation consider in the best logic will affect the performance of the retrieved induced logic.

2.3. Ineffective learning and retrieval phase of DHNN

Optimizing DHNN is crucial to ensure that the network only retrieved optimal final neuron states before converted into induced logic. Several logic mining models in the literature focus on enhancing the learning capabilities of DHNN in doing logic mining. For instance, Zamri et al. [12] proposed Clonal selection algorithm in the learning phase of 3SATRA. In another study by Alway et al. [19], the learning phase of 2HESRA was optimized by using Hybrid Exhaustive Search approach while Zamri et al. [20] focused on optimizing the learning phase with the Modified Niche Genetic Algorithm. These models primarily aim to obtain optimal synaptic weights, ensuring that the final neuron states achieve global minima energy. However, a notable issue in these models is their concentration on obtaining optimal synaptic weights, raising questions about the quality of the final neuron states, particularly in terms of diversification of the solution string which may lead to overfitting. Additionally, there has been limited effort to optimize the retrieval capabilities of DHNN, addressing this limitation. On the other hand, previous studies have focused on improving either the learning or retrieval phase individually, and little attention was given on optimizing both phases. Motivated by these considerations, this study proposes an optimal logic mining model with the capability to guarantee optimal synaptic weights during the learning phase, leading to optimal induced logic without jeopardizing the quality of the solutions. To achieve this, the Election Algorithm proposed by Sathasivam et al. [5] is incorporated during the learning phase, ensuring the acquisition of optimal synaptic weights. Additionally, a binary Clonal Selection Algorithm is introduced in the retrieval phase to ensure the diversification of the solution string in terms of negativity, ultimately aiming for the attainment of global solutions with a low similarity index.

2.4. Ineffective attribute selection

One of the primary concerns in the pre-processing phase of the logic mining process is the effectiveness of attribute selection. The main idea behind applying an attribute selection approach is to identify the most influential attributes that significantly contribute to the output. Without optimal attribute selection, the quality of the induced logic obtained remains uncertain. For instance, E2SATRA proposed by Jamaludin et al. [13] capitalize random attribute selection that leads to the creation of 2SAT logic. While the induced logic guarantees achieving global minima energy during the retrieval phase, there is a high risk of selecting non-significant attributes. Consequently, this situation leads to a question what will happen if the model selects the wrong attribute? In this context, the interpretability of the retrieved induced logic may be compromised, as the model could extract incorrect knowledge about the data. This occurs when the network learns from the wrong attributes, leading to suboptimal induced logic. To encounter this issue, several works in literature introduced supervised learning methods to select the most significant attributes for logic creation. For example, Jamaludin et al. [16] and Manoharam et al. [18] proposed log linear approach while Kasihmuddin et al. (2022), Rusdi et al. [17] and Alway et al. [19] employed an association analysis to extract attributes representing the behavior of the dataset. However, these supervised learning approaches may lead to random selections if they do not meet statistical conditions during the correlation test. Since the model is unable to select the best attributes from the dataset, the chances of being unable to interpret the logical rule are high. Consequently, this will reduce the quality of the induced logic, leading to a lower accuracy value. Motivated by these challenges, this paper introduces a relatively new unsupervised approach using Topological Data Analysis (TDA) to select attributes without relying on statistical analysis. Notably, the proposed logic mining approach consistently selects attributes that give important affect to the behavior of the dataset.

3. Weighted systematic 2 satisfiability

Weighted Systematic 2 Satisfiability ($rS2SAT$) is a new class of systematic SAT formula by considering of 2 literals per clause. The two important components of $rS2SAT$ are the logic is a systematic SAT expressed in Conjunctive Normal Form (CNF) and there is an insertion of a weighted feature to control the distribution of negative literals in the form of ratio (r). The aim of considering this weighted feature is to ensure the generation of $rS2SAT$ is according to the desired r . In comparison to the random distribution of negative literals, the weighted feature is able to generate a consistent number of negative literals with no issue of literal repetition throughout the logic. To further discuss the novelty of the $rS2SAT$, the general equation of the logical structure of $rS2SAT$ is presented in Eq (1).

$$P_{rS2SAT} = \bigwedge_{i=1}^y C_i, \quad (1)$$

where y is the total number of clauses in the P_{rS2SAT} and C_i is formulated as in Eq (2).

$$C_i = \bigvee_{i=1}^n (A_i, B_i), \quad (2)$$

whereby n is the total number of literals in the P_{rS2SAT} and $n = 2y$. Note that, there is no redundant literals in $rS2SAT$ to ensure optimal synaptic weight management by WA method. Each literal of

$\{A_i, B_y\}$ in P_{rS2SAT} holds discrete form of interpretations in bipolar values of $(1, -1)$ where it represents of True and False interpretation, respectively. Additionally, the distribution of literals in each C_i can be either negative ($\neg A$) or positive (A).

As mentioned, the distribution of negative literals in P_{rS2SAT} is controlled by a weighted ratio or can be defined as r . In other words, r is the ratio of negative literals exists in P_{rS2SAT} . The main factor of introducing r is to control and increase the diversification of negative literals in the logic. Worth mentioning that, r can generate dynamic number of negative literals for any determined n . As a logical rule in DHNN, the advantage of r is able to force the network in learning negative neuron connections. Hence, this will improve the variation issues in the synaptic weight management. Notably, this paper proposes r with the range of $r = [0.1, 0.9]$, and a step size of $\Delta r = 0.1$. The reason why $\Delta r = 0.1$ is chosen because when the step size is too large, this will increase the margin of negative literals for the same n [21]. Consequently, several amounts of negative literals will be disregarded. However, if the value of step size is too small, P_{rS2SAT} will generate redundant number of negative literals for certain r . Subsequently, the distribution of the negative literals in the generation of P_{rS2SAT} follows random distribution with the desired r . The total number of desired negative literals in P_{rS2SAT} can be computed based on Eq (3).

$$\lambda = \lfloor rn \rfloor, \quad (3)$$

whereby the mathematical notation of $\lfloor \ \rfloor$ represents the rounding down with the floor function. This means that the value of λ is the rounding down product of r and defined n . Note that, n is always greater and equal to λ . The generation of P_{rS2SAT} with respect to λ is optimized by using Eq (4).

$$|\kappa - \lambda| = 0, \quad (4)$$

where κ is the total weights in the logical structure of P_{rS2SAT} and is evaluated as in Eq (5).

$$\kappa = \sum_{i=1}^n \Gamma_i. \quad (5)$$

Γ_i is the weight of the literals in the $rS2SAT$ and the weight for each literal is illustrated in Eq (6).

$$\Gamma_i = \begin{cases} 0, & \text{if } A_i \\ 1, & \text{if } \neg A_i \end{cases}. \quad (6)$$

The value of κ must always aligned with the desired λ to ensure that the generation of P_{rS2SAT} is optimal. Table 1 shows the example of P_{rS2SAT} for $n = 10$ in all r .

Table 1. The example of P_{rS2SAT} in all r for $n = 10$.

r	Example of P_{rS2SAT}
0.1	$P_{rS2SAT} = (A_1 \vee B_1) \wedge (A_2 \vee B_2) \wedge (A_3 \vee \neg B_3) \wedge (A_4 \vee B_4) \wedge (A_5 \vee B_5)$
0.2	$P_{rS2SAT} = (A_1 \vee B_1) \wedge (A_2 \vee B_2) \wedge (\neg A_3 \vee B_3) \wedge (A_4 \vee B_4) \wedge (\neg A_5 \vee B_5)$
0.3	$P_{rS2SAT} = (\neg A_1 \vee \neg B_1) \wedge (\neg A_2 \vee B_2) \wedge (A_3 \vee B_3) \wedge (A_4 \vee B_4) \wedge (A_5 \vee B_5)$
0.4	$P_{rS2SAT} = (A_1 \vee \neg B_1) \wedge (\neg A_2 \vee \neg B_2) \wedge (A_3 \vee B_3) \wedge (A_4 \vee B_4) \wedge (A_5 \vee \neg B_5)$
0.5	$P_{rS2SAT} = (A_1 \vee B_1) \wedge (\neg A_2 \vee B_2) \wedge (\neg A_3 \vee \neg B_3) \wedge (A_4 \vee \neg B_4) \wedge (\neg A_5 \vee B_5)$
0.6	$P_{rS2SAT} = (A_1 \vee \neg B_1) \wedge (\neg A_2 \vee \neg B_2) \wedge (\neg A_3 \vee \neg B_3) \wedge (A_4 \vee \neg B_4) \wedge (A_5 \vee B_5)$
0.7	$P_{rS2SAT} = (A_1 \vee \neg B_1) \wedge (\neg A_2 \vee \neg B_2) \wedge (\neg A_3 \vee \neg B_3) \wedge (\neg A_4 \vee \neg B_4) \wedge (A_5 \vee B_5)$
0.8	$P_{rS2SAT} = (A_1 \vee \neg B_1) \wedge (\neg A_2 \vee \neg B_2) \wedge (\neg A_3 \vee \neg B_3) \wedge (\neg A_4 \vee \neg B_4) \wedge (A_5 \vee \neg B_5)$
0.9	$P_{rS2SAT} = (A_1 \vee \neg B_1) \wedge (\neg A_2 \vee \neg B_2) \wedge (\neg A_3 \vee \neg B_3) \wedge (\neg A_4 \vee \neg B_4) \wedge (\neg A_5 \vee \neg B_5)$

The proposed P_{rS2SAT} will act as the symbolic language to govern the network whereby. P_{rS2SAT} possess satisfied interpretations that mapped to either true or false logical outcome. Therefore, P_{rS2SAT} will be embedded to DHNN as a proposed logical rule.

4. Weighted systematic 2 satisfiability in the discrete Hopfield neural network

In this section, the implementation of the proposed $rS2SAT$ as neurons in DHNN will be further discussed which can be denoted as U2SAT model. The proposed U2SAT model consists of two major phases: The learning phase and the retrieval phase. In this paper, the WA method [2] will be capitalized in the learning phase of the proposed model. The retrieval phase will generate the final neuron states of the network which also mapped to the final energy profile of the network. Notably, each literal of $rS2SAT$ will represent each unit of neurons in DHNN. When the correct structure of P_{rS2SAT} is generated, the P_{rS2SAT} will then proceed to the learning phase of U2SAT. Equations (7) and (8) presented the cost function of U2SAT in consideration of WA method.

$$E_{P_{rS2SAT}} = \frac{1}{4} \sum_{i=1}^y \left(\prod_{j=1}^2 L_{ij} \right), \quad (7)$$

$$L_{ij} = \begin{cases} (1 - S_{A_i}), & \text{if } \neg A_i \\ (1 + S_{A_i}), & \text{if } A_i \end{cases}, \quad (8)$$

where A_i is an example of one literal in P_{rS2SAT} . Take the formulated P_{rS2SAT} for $r = 0.3$ in Table 1 as an example to illustrate the U2SAT model. First, derive the consistency of the P_{rS2SAT} as in Eq (9).

$$\neg P_{rS2SAT} = (A_1 \wedge B_1) \vee (A_2 \wedge \neg B_2) \vee (\neg A_3 \wedge \neg B_3) \vee (\neg A_4 \wedge \neg B_4) \vee (\neg A_5 \wedge \neg B_5). \quad (9)$$

Then, the formulation of $E_{P_{rS2SAT}}$ based on Eq (9) is formulated in Eq (10).

$$\begin{aligned} E_{P_{rS2SAT}} &= \frac{1}{4}(1 + S_{A_1})(1 + S_{B_1}) + \frac{1}{4}(1 + S_{A_2})(1 - S_{B_2}) \\ &\quad + \frac{1}{4}(1 - S_{A_3})(1 - S_{B_3}) + \frac{1}{4}(1 - S_{A_4})(1 - S_{B_4}), \\ &\quad + \frac{1}{4}(1 - S_{A_5})(1 - S_{B_5}) \end{aligned} \quad (10)$$

$$E_{P_{rS2SAT}} = \frac{5}{4} + \frac{1}{4} \begin{pmatrix} S_{A_1}S_{B_1} - S_{A_2}S_{B_2} + S_{A_3}S_{B_3} + S_{A_4}S_{B_4} \\ + S_{A_5}S_{B_5} + S_{A_1} + S_{B_1} + S_{A_2} - S_{B_2} \\ - S_{A_3} - S_{B_3} - S_{A_4} - S_{B_4} - S_{A_5} - S_{B_5} \end{pmatrix}, \quad (11)$$

whereby Eq (11) is the expanded version of Eq (10). According to the WA method, the synaptic weights can be calculated by comparing the coefficients of the $E_{P_{rS2SAT}}$ with the Lyapunov energy function, which is formulated in Eq (12).

$$H_{P_{rS2SAT}} = -\frac{1}{2} \sum_i \sum_j W_{ij}^{(2)} S_i S_j - \sum_i W_i^{(1)} S_i. \quad (12)$$

Equation (13) formulates the Lyapunov energy function based on the example of P_{rS2SAT} .

$$\begin{aligned} H_{P_{rS2SAT}} &= -\frac{1}{2} \begin{pmatrix} 2W_{A_1B_1}^{(2)} S_{A_1} S_{B_1} + 2W_{A_2B_2}^{(2)} S_{A_2} S_{B_2} + 2W_{A_3B_3}^{(2)} S_{A_3} S_{B_3} \\ + 2W_{A_4B_4}^{(2)} S_{A_4} S_{B_4} + 2W_{A_5B_5}^{(2)} S_{A_5} S_{B_5} \end{pmatrix} \\ &\quad - \begin{pmatrix} W_{A_1}^{(1)} S_{A_1} + W_{B_1}^{(1)} S_{B_1} + W_{A_2}^{(1)} S_{A_2} + W_{B_2}^{(1)} S_{B_2} + W_{A_3}^{(1)} S_{A_3} \\ + W_{B_3}^{(1)} S_{B_3} + W_{A_4}^{(1)} S_{A_4} + W_{B_4}^{(1)} S_{B_4} + W_{A_5}^{(1)} S_{A_5} + W_{B_5}^{(1)} S_{B_5} \end{pmatrix}. \end{aligned} \quad (13)$$

According to Abdullah [2], to ensure correct synaptic weight values of U2SAT is obtained, the neurons (S_{A_i}, S_{B_i}) must possess bipolar values that satisfy the P_{rS2SAT} . By taking Eq (14) as an example of interpretations for P_{rS2SAT} to be satisfied ($P_{rS2SAT} = 1$) or leads to $E_{P_{rS2SAT}} = 0$.

$$\begin{pmatrix} S_{A_1}, S_{B_1}, S_{A_2}, S_{B_2}, S_{A_3}, \\ S_{B_3}, S_{A_4}, S_{B_4}, S_{A_5}, S_{B_5} \end{pmatrix} = (-1, 1, -1, -1, 1, 1, 1, -1, -1, 1). \quad (14)$$

Hence, the evaluation of $W_{A_1B_1}^{(2)}$ can be calculated in Eq (15).

$$\begin{aligned} -\frac{1}{2} \left(2W_{A_1B_1}^{(2)} S_{A_1} S_{B_1} \right) &= \frac{1}{4} (S_{A_1} S_{B_1}) \\ -W_{A_1B_1}^{(2)} (-1)(1) &= \frac{1}{4} (-1)(1) \quad . \\ W_{A_1B_1}^{(2)} &= -\frac{1}{4} \end{aligned} \quad (15)$$

The overall values of $W_{ij}^{(2)}$ and $W_i^{(1)}$ can be obtained when considering the neurons in Eq (14) and then will be stored in a unit of Content Addressable Memory (CAM). The feature of CAM is to remind the network of the optimal neuron connections. In this context, DHNN will converge to a final state for any given initial states in the local field computation. This results in the production of final neuron states with global energy solutions. In this paper, the objective function of the learning phase of U2SAT are mapped to the quality of the satisfied interpretations in terms of fitness ($E_{P_{rS2SAT}} = 0$). Therefore, EA is utilized in U2SAT to maximize the satisfied interpretation for P_{rS2SAT} that minimizing the cost function during the learning phase of DHNN.

The retrieval phase of U2SAT is responsible to execute the local field computation that leads to the production of final neuron state, S_i^f . The production of S_i^f can be evaluated using Eq (16) and the neuron is updated using Eq (17).

$$h_i(t) = \sum_{j=1, j \neq i}^n W_{ij}^{(2)} S_j + W_i^{(1)}, \quad (16)$$

$$S_i^f(t) = \begin{cases} 1, & \tanh(h_i) \geq 0 \\ -1 & \text{otherwise} \end{cases}, \quad (17)$$

where $S_j \in \{-1, 1\}$ is the initial neuron states. The correctness of the retrieved S_i^f in the retrieval phase is verified by using the energy profile. The energy profile of the retrieved S_i^f is formulated in Eqs (18) and (19).

$$H_{P_{rS2SAT}} = -\frac{1}{2} \sum_{i=1, i \neq j}^n \sum_{j=1, j \neq i}^n W_{ij}^{(2)} S_i^f S_j - \sum_{i=1}^n W_i^{(2)} S_i^f, \quad (18)$$

$$H_{P_{rS2SAT}}^{\min} = -\frac{1}{2} y. \quad (19)$$

Successful retrieval phase relies upon the ability of the program to obtain global minimum energy. Equation (20) formulated the condition for the energy produced in the retrieval phase.

$$\left| H_{PrS2SAT} - H_{PrS2SAT}^{\min} \right| \leq tol, \quad (20)$$

where tol is a tolerance value. If the energy produced by the program aligned with Eq (20), then S_i^f of U2SAT obtained global minimum energy. When a program obtained global minimum energy, the collection of S_i^f retrieved by U2SAT can be a generalized as an intelligent model that practice $rS2SAT$ as a logical rule.

In this paper, the objective function in the retrieval phase not only focuses on achieving the global minimum solution, but also considers diversity fitness, and the ratio of similarity index for the S_i^f produced before and after the implementation of bCSA. Therefore, we address the multi-objective function in the retrieval phase of DHNN as formulated in Eqs (21)–(24).

$$Obj(G_f, D_r, R_s), \quad (21)$$

such that

$$G_f = S_i^f, \text{ where } S_i^f \rightarrow \left| H_{PrS2SAT} - H_{PrS2SAT}^{\min} \right| \leq tol, \quad (22)$$

$$D_r \geq \omega, \quad (23)$$

$$R_s \leq \gamma. \quad (24)$$

There are three major requirements that need to be satisfied in the multi-objective function. First, the S_i^f produced must be able to achieve global minimum solution. Each S_i^f that able to achieve condition in Eq (20) is denoted as G_f . G_f is obtained using the process in Eqs (18)–(20). Second, S_i^f must have the ability to maintain the diversity fitness of the neuron state within the diversity ratio, D_r . The D_r is important to ensure that the S_i^f produced have variation of diversity based on the existences of negative neuron states. The existence of negative neuron states in S_i^f is guaranteed to be at least ω . The process of evaluating the diversity fitness is in Eqs (25) and (26).

$$F_d = \sum_{i=1}^u d_i, \quad (25)$$

$$d_i = \begin{cases} 0, & \text{if } (A_i^f, B_i^f) = (1,1), \\ 1, & \text{otherwise} \end{cases}, \quad (26)$$

where F_d measures the diversity fitness based on the frequency of d_i for each S_i^f . According to Eq (26), the scoring mechanism will be zero if all the neuron states in the S_i^f are 1 (positive). Therefore, the formulation of D_r is shown in Eq (27).

$$D_r = \frac{F_d}{y}, \quad (27)$$

where y is the total number of clauses in P_{rS2SAT} .

Third, U2SAT need to ensure that the similarity of the S_i^f produced before and after the implementation of bCSA is within the similarity ratio, R_s . The S_i^f before and after the implementation of bCSA is denoted as S_i^{fb} and S_i^{fa} , respectively. It is important to preserve the dissimilarity among all the S_i^f produced [22]. Therefore, the process of identifying the similarity of the S_i^f is using Rogers-Tanimoto (R_s) similarity index. Eqs (28)–(32) expressed the formulation of R_s used in the U2SAT.

$$R_s = \frac{a + d}{a + 2(b + c) + d}, \quad (28)$$

$$a = \begin{cases} 1, & \text{if } (A_i^{fb}, A_i^{fa}) = (1, 1), \\ 0, & \text{otherwise} \end{cases}, \quad (29)$$

$$b = \begin{cases} 1, & \text{if } (A_i^{fb}, A_i^{fa}) = (1, -1), \\ 0, & \text{otherwise} \end{cases}, \quad (30)$$

$$c = \begin{cases} 1, & \text{if } (A_i^{fb}, A_i^{fa}) = (-1, 1), \\ 0, & \text{otherwise} \end{cases}, \quad (31)$$

$$d = \begin{cases} 1, & \text{if } (A_i^{fb}, A_i^{fa}) = (-1, -1), \\ 0, & \text{otherwise} \end{cases}. \quad (32)$$

Hence, the goal for S_i^f to be selected as the new final solution (S_i^{nf}) of U2SAT must meet the condition specified in Eq (33).

$$S_i^f \rightarrow \text{Obj}(G_f, D_r, R_s). \quad (33)$$

In order to fulfil the multi-objective function, this paper proposes binary Clonal Selection Algorithm (bCSA) as a retrieval algorithm. The process of bCSA is discussed in detail in Section 5. Figure 1 displays the schematic diagram of U2SAT that describes the structure of P_{rS2SAT} in DHNN. Moreover, Algorithm 1 depicts the overall process of U2SAT.

Algorithm 1. The Pseudocode of U2SAT

```

1  Input: Set the initial parameters such as number of trials ( $NT$ ) and maximum combination
   ( $combmax$ ).
2  Begin
3  Initialize the neuron to each variable that consists of  $A_i \in [A_1, A_2, A_3, \dots, A_n]$ ;
4  while ( $i \leq NT$ )
5  Form the initial states using Equation (2);
6  Training Phase
7  Define  $E_{P_rS2SAT}$  using Equation (7) – Equation (8);
8  for  $C_i \in [A_1, A_2, A_3, \dots, A_n]$  do
9  Check clauses satisfaction;
10 if  $E_{P_rS2SAT} = 0$ 
11  $C_i$  is satisfied;
12 else
13  $C_i$  is not satisfied;
14 end if
15 Compute synaptic weights using WA method;
16 Store the synaptic weight in CAM;
17 end for
18 Retrieval Phase
19 Compute  $h_i$  using Equation (16);
20 Compute  $H_{P_rS2SAT}$  using Equation (18);
21 Compute  $H_{P_rS2SAT}^{\min}$  using Equation (19);
22 for  $S_i^f \in [S_1^f, S_2^f, S_3^f, \dots, S_n^f]$  do
23 Check the quality of the final neuron state using Equation (33);
24 if  $S_i^f \rightarrow Obj(G_f, D_r, R_s)$ 
25 Assign as  $S_i^{nf}$ ;
26 else
27 Do binary Clonal Selection Algorithm;
28 end if
29 end for
30 end while
31 Output: Final neuron states achieved Multi-Objective Function

```

Stage 1: Initialization of β

In this stage, number of B-cells (N_β) will be generated. Note that, this stage is the consecutive process after the evaluation of Eq (20). Hence, the value of N_β is depending on the defined number of solutions produced by DHNN in the retrieval phase which is usually denoted by number of trials (NT). In this context, each solution retrieved by the network is represented by one β whereby $\beta_i = 1, 2, 3, \dots, NT$ ($N_\beta = NT$) and each neuron states in the β will be represented by a gene (g) whereby $g_i = 1, 2, 3, \dots, 2y$.

Stage 2: Affinity Evaluation of β

Based on the fundamental CSA introduced by Castro and Zuben [23], affinity evaluation is the measurement to identify the “maturing” level of the or known as the fitness evaluation process. In this stage, the fitness of each β will be evaluated based on Eqs (34)–(38) as follows:

$$\beta_{affinity} = [\beta_f, \beta_d], \quad (34)$$

whereby,

$$\beta_f = \sum_{i=1}^y F_i, \quad (35)$$

$$F_i = \begin{cases} 1, & \text{satisfied} \\ 0, & \text{otherwise} \end{cases}, \quad (36)$$

$$\beta_d = \sum_{i=1}^y D_i, \quad (37)$$

$$D_i = \begin{cases} 0, & \text{if } (g_i^{A_i}, g_i^{B_i}) = (1, 1), \\ 1, & \text{otherwise} \end{cases}, \quad (38)$$

where $\beta_{affinity}$ is the affinity value of each β with respect to both β_f and β_d . Note that F_i measures the satisfiable property of each β based on the initialized $rS2SAT$. Hence, the optimal value of β_f is always equal to the number of initialized clauses or known as (y). Second, β_d measures the diversity fitness of each β based on the existences of negative neuron states or $g_i = -1$. In this context, the frequency of D_i will be counted only when there exists a negative state for each clause in the β . As shown in Eq (38), the scoring mechanism will be zero if all g in the β are positive (neuron states is 1).

Stage 3: Affinity maturation level via a scoring strategy

Due to the multi-objective function, there are possible instances of Pareto fronts β whereby the β only possessed optimal fitness for only one of the fitness functions (either β_f or β_d) but not both. However, Pareto fronts β is not considered as an optimal solution as it is trading off one of the fitness. According to Zamri et al. [20], only single best solution with respect to fitness and diversity is plausible because the trade-offs between these two allows DHNN to produce solutions that is trapped in local optima and tends to overfit. Hence, to ensure optimal optimization of the proposed bCSA, this paper only considers single best solution of β with respect to β_f and β_d . Therefore, this stage is crucial for identifying, selecting, sorting, and prioritizing which β that has the fittest g based on the evaluated β_f and β_d . A scoring strategy is introduced to identify the “fittest” β before proceeded to the next stages of the proposed bCSA. Once the $\beta_{affinity}$ of each β is evaluated, the scoring mechanism of each β with respect to β_f and β_d will be evaluated using Eqs (39) and (40), respectively, as follows:

$$R_{\beta_f} = \begin{cases} 1, & \text{if } |y - \beta_f| = 0 \\ 0, & \text{otherwise} \end{cases}, \quad (39)$$

$$R_{\beta_d} = \begin{cases} 1, & \text{if } |y - \beta_d| = 0 \\ 0, & \text{otherwise} \end{cases}. \quad (40)$$

As an example, let's take $y = 6$. If a β_1 possessed $\beta_f = 6$ and $\beta_d = 4$, the scores for R_{β_f} and R_{β_d} are one and zero respectively. Subsequently, if β_2 possessed $\beta_f = 6$ and $\beta_d = 6$, the scores for both R_{β_f} and R_{β_d} are one. The maturation level of the β is decided based on the summation of R_{β_f} and R_{β_d} as presented in Eq (41) as follows:

$$R_{\beta} = R_{\beta_f} + R_{\beta_d}, \quad (41)$$

where $R_{\beta} \in \{0, 1, 2\}$. When $R_{\beta} = 2$, this shows that the β possessed the fittest g with respect to both fitness and diversity. These β will be stored as the S_i^f of the proposed U2SAT. The remaining β with $R_{\beta} = 0, 1$ will proceed to the next stage in bCSA. If none of the β attained maximum R_{β} , all the β will automatically proceed to the next stage of the algorithm. Notably, this algorithm will terminate once the proposed bCSA found NT number of β with maximum R_{β} .

Stage 4: Selection of β

For this stage, number of β with suboptimal R_{β} (N_{β_2}) will be selected to the next stage of bCSA. The value of N_{β_2} can be evaluated using Eq (42) as follows:

$$N_{\beta 2} = \delta_S(N_{\beta 1}), \quad (42)$$

whereby δ_S is the selection rate of (0,1) and $N_{\beta 1}$ is the number of β from Stage 3.

Stage 5: Cloning

In this stage, the selected β will be cloned in proportion to the $\beta_{affinity}$ calculated. In other words, β with the highest R_β undergo the cloning process by duplicating all g which resulting to the new cloned β into the current population. The population clone size ($N_{\beta 3}$) of β is evaluated using Eq (43) below. Note that this approach was taken into into account to promote the exploration of other potential β with strong global optimization ability [24].

$$N_{\beta 3} = \delta_C(N_{\beta 2}), \quad (43)$$

where δ_C is the cloning rate of (0,1).

Stage 6: High-frequency mutation via somatic hypermutation

Similar to the somatic hypermutation mechanism proposed by Zamri et al. [12], all the β in this stage are subjected to high-frequency mutation. This type of mutation gives the proposed bCSA the ability of local search. Furthermore, it also maintains the diversity of the β in the population. Based on the conventional mutation in the study of GA, the possibility of which β to be mutated is set by roulette wheel selection or at random. Hence, there is a possibility whereby the mutation process never takes place. Unfortunately, this will steer the algorithm in the direction of trapping in local optimal solutions. Therefore, the number of β from $N_{\beta 3}$, ($N_{\beta 4}$) that will undergo somatic hypermutation is evaluated as in Eq (44).

$$N_{\beta 4} = \delta_M(N_{\beta 3}), \quad (44)$$

where δ_M is the mutation rate of (0,1). The proposed somatic hypermutation will ensure that β will undergo the mutation with at least one g is being mutated with respect to the objective function. Finally, all stored and improved β are being proceeded as the S_i^f of the U2SAT with respect to high fitness and diversity. Regeneration of β in Stage 1 are subjected to the generation of the local field computation. Therefore, Stages 1–6 are repeated until the termination conditions are met.

6. Proposed logic mining model

In an effort to enhance the effectiveness of the existing logic mining model by Kho et al. [11], Kasihmuddin et al. [15], and Jamaludin et al. [14], we have introduced a novel logic mining method, namely Weighted Systematic 2 Satisfiability based Reverse Analysis Method (U2SATRA). The primary purpose of our proposed logic mining is to extract knowledge from a dataset and represent it in the form of logical rules after considering significant attributes. In general, the effectiveness of logic

mining is dependent upon the efficiency of U2SAT and the Reverse Analysis method in processing the dataset and transforming it into induced logic. Specifically, U2SAT comprises three major phases: Pre-processing phase, learning phase, and retrieval phase. The steps and explanations for each phase will be discussed in the following subsections respectively.

6.1. Pre-processing phase

Pre-processing phase should be initiated to learn the dataset in order to improve the quality of the data before implementing it into logic mining. This phase focuses on three major steps: The attribute selection through Topological Data Analysis (TDA) approach, data preparation, and data splitting. Hence, the steps involved in this phase will be explained accordingly. First, among all the variables in the raw dataset, only R optimal attributes is selected for the process of U2SAT. The attribute selection is utilized to filter all the important attributes with respect to the dataset. The implementation of TDA technique, specifically the mapper algorithm is utilized as an attribute selection method to filter attributes that significantly affect the structure of the dataset [25]. Notably, TDA is a dimensionality reduction technique that maps data from its original high-dimensional space to a low-dimensional space that it is easier to understand and visualize [26]. The Mapper technique is employed to analyze the dataset where the dataset is transforms to a simpler form. Generally, the Mapper technique starts with transforming original data into one-dimensional using filter function. Then, the number of hypercubes denoted as (hc) is created based on the transformed data and clustering method is applied for each hypercube. Specifically with a dataset Y , the steps in applying the Mapper technique are explained as follows with a point cloud consists of a collection of points (referring to the variables in the raw dataset, A_i). First, choose a filter function as in Eq (45) to map the dataset into Euclidean space, \mathbb{R}^d .

$$f: Y \rightarrow \mathbb{R}^d. \quad (45)$$

Then, the range of values for the image of the function, $f(Y)$ is partitioned into collection of open sets, $(\Gamma_i)_{i \in I}$ whereby I is a finite indexing set. The partitioned number is set by parameter hc and are overlap with each other with respect overlapping percentage, denoted as hp . Next, for each interval Γ_i , the points in the preimage $f^{-1}(\Gamma_i)$ is clustered by utilizing clustering algorithm and it will form a set of m clusters, K_m [27]. Following to that, each cluster, K_m is corresponds to a node and each node consists of attribute A_i with similar behavior. This study only chooses one attribute from each node as it is similar with other attributes in same node. By applying this rule, in the end, a collection of significant attributes is obtained and are implemented as the selected attributes in application of U2SAT. Hence, only one A_i is chosen in each node. Worth mentioning that, TDA helps in filtering all the A_i that have the similar behavior so that all the selected A_i insert in U2SAT give significant effect to the dataset. The selection of N optimal attributes that will be used in the U2SAT is important to maintain the quality of induced logic by excluding non-optimal attributes [16].

After R attributes has been selected, the entries from the raw data, X_{ij}^i in the selected attributes will be changed into bipolar representation by using k -means clustering technique [14]. Before X_{ij}^i can be changed to bipolar, the mean value for each attribute is calculated. Each mean obtained in each attribute is used as an indicator to change the X_{ij}^i in each attribute into bipolar representation. This

method is important to ensure the convergence of the final neuron state in U2SAT [1]. Therefore, the formulation of k -mean clustering technique is formulated in Eq (46).

$$S_i = \begin{cases} -1, & \text{if } X_{ij}^i < \overline{A_i} \\ 1, & \text{otherwise} \end{cases} \quad (46)$$

After all the X_{ij}^i has been changed to bipolar, the dataset will be split into learning and retrieval data with the ratio of $a : b$. The learning and retrieval data is denoted as P_{learn} and P_{test} , respectively. This ratio has a good agreement with majority of the existing logic mining models. Additionally, k -fold cross validation is considered in the proposed logic mining model. Note that, the average of all folds will be taken as the final result for all metrics considered.

6.2. Learning phase

In the context of learning phase, the proposed U2SATRA introduce an alternative computation of best logic that maximizes true and false classification of the learning data. This will proffer the proposed logic mining in learning important structure of $rS2SAT$ that is able to capture the classification patterns in the learning data. The steps involve in the learning phase will be discussed in detail. The first step in learning phase is P_{rS2SAT}^b will be generated for all values of $r = [0.1, 0.9]$. The maximum number of unique P_{rS2SAT} , $N_{P_{rS2SAT}}$ can be evaluated by considering the combination of n and the defined λ as in Eq (47).

$$N_{P_{rS2SAT}} = {}^n C_{\lambda} \quad (47)$$

Note that, only unique structures of P_{rS2SAT} is generated to avoid learning redundant logical rule. Next, the outcome of P_{rS2SAT}^b will be compared with the outcome from P_{learn} to obtain the classification of confusion matrix as in Figure 2. Then, P_{rS2SAT}^b will be evaluated based on five performance metrics which is Accuracy (ACC), Sensitivity (SNS), Specificity (SPC), Negative Predictive Value (NPV), and Mathews Correlation Coefficient (MCC). For each performance metric, q number of P_{rS2SAT}^b with the highest value of the metric are selected as the best logic. Note that, each performance metric is treated independently. Therefore, each DHNN is corresponds to the best logic produced based on each performance metric and it is defined as multi-unit DHNN. Next, the selected P_{rS2SAT}^b will be learned using EA in consideration of achieving objective function in learning phase of U2SAT.

6.3. Retrieval phase

Retrieval phase is the last stage in the U2SATRA. Notably, the proposed best logic produced multi-unit DHNN whereby each DHNN corresponds to each performance metric. Therefore, the retrieval phase will be executed for five number of times whereby each retrieval phase corresponds to each performance metric. The step in retrieval phase is start with capitalizing the values of the synaptic weight for each P_{rS2SAT}^b in Eq (16) and the final neuron states will be retrieved after the implementation of bCSA. Note that, the consideration of multi-objective in Eq (21) is initiated in the

proposed logic mining model. Then, all possible induced logic, P_{rS2SAT}^i will be produced after U2SAT retrieved S_i^{nf} . Based on the RA method proposed by Sathasivam and Abdullah [10], the retrieved final neuron states will be transformed into SAT logical rule as the induced logic. Therefore, the possible P_{rS2SAT}^i is produced based on Eq (48).

$$S_i^{induced} = \begin{cases} A_i, & S_i^{nf} = 1 \\ \neg A_i, & S_i^{nf} = -1 \end{cases} \quad (48)$$

After that, the outcome from P_{rS2SAT}^i will be compared with the outcome from P_{test} in order to assess the classification of TP , TN , FP , and FN based on Figure 2. The best P_{rS2SAT}^i will be selected based on the highest performance metric attained from the entries of P_{test} . Worth noting that, the best P_{rS2SAT}^i will represent each performance metric used in finding the best logic. The overall implementation of U2SATRA from the pre-processing phase to the retrieval phase is presented in Figure 3. Worth mentioning that, the green, orange, and blue boxes in Figure 3 represents pre-processing, learning, and retrieval phase, respectively. In addition, Algorithm 2 shows the Pseudocode of the proposed U2SATRA.

		Outcome from the $P_{rS2SAT}^b / P_{rS2SAT}^i$	
		Positive (1)	Negative (-1)
Outcome from the P_{learn} / P_{test}	Positive (1)	True Positive (TP)	False Negative (FN)
	Negative (-1)	False Positive (FP)	True Negative (TN)

Figure 2. The classification of confusion matrix used in U2SATRA.

Algorithm 2. The Pseudocode of U2SATRA

```

1  Input: Set all attributes of  $A_1, A_2, A_3, \dots, A_n$  with respect to  $P_{learn}, P$ , and  $NT$ .
2  Begin
3  Initialize algorithm parameters;
4  | Pre-processing Phase
5  | Do TDA;
6  | for ..( $n(Y_i) = R$ )..do
7  |   | if [ $(A_i \in K_m) \cap (|K_m| = 1)$ ]
8  |   |   Assign  $A_i$  as  $Y_i$ ;
9  |   |   else
10 |   |   Not selected as  $Y_i$ ;
11 |   |   end if
12 | end for
13 | while ( $i \leq P$ ) do
14 |   | Training Phase
15 |   | Generate  $P_{rS2SAT}^b$ ;
16 |   | for ( $i \leq combmax$ ) do
17 |   |   Compare the outcome from the  $P_{learn}$  and  $P_{rS2SAT}^b$ ;
18 |   |   Select best  $P_{rS2SAT}^b$  that achieve highest value of correspond metric;
19 |   |   end for
20 |   | Check the clause satisfaction for best  $P_{rS2SAT}^b$ ;
21 |   | Compute the synaptic weight associated with best  $P_{rS2SAT}^b$  using WA method;
22 |   | Store the synaptic weight and best  $P_{rS2SAT}^b$  in CAM;
23 |   | Retrieval Phase
24 |   | Initialize the final neuron state;
25 |   | for ( $t \leq NT$ ) do
26 |   |   Compute  $h_i$  using Equation (16);
27 |   |   Convert  $S_i^{nf}$  to bipolar state using Equation (17);
28 |   |   Transform  $S_i^{nf}$  into logical form using Equation (48) with respect to the
29 |   |   attributes;
30 |   |   Combine  $S_i^{nf}$  to generate induced logic  $P_{rS2SAT}^i$ ;
31 |   |   Compare the outcome of the  $P_{rS2SAT}^i$  with  $P_{test}$ ;
32 |   |    $t \leftarrow t + 1$ ;
33 |   |   end for
34 |   |  $i \leftarrow i + 1$ ;
35 | end while
36 Output: The best  $P_{rS2SAT}^i$  that obtain highest value of correspond metric.

```

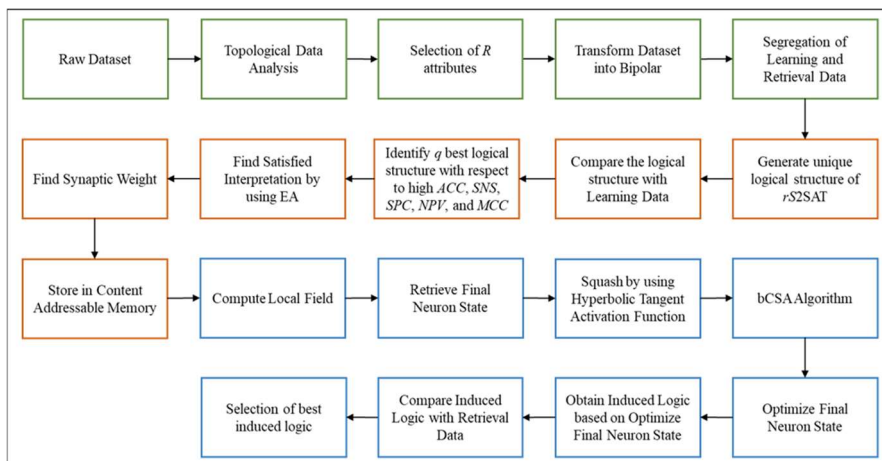


Figure 3. The overall process of U2SATRA.

7. Experimental framework

The simulation was conducted to evaluate the performance of U2SATRA in generalizing the best induced logic that represents the behavior of the dataset. To ensure the robustness of the experiment, the following setup must be employed as follows:

7.1. Datasets description

This research employs 20 data sets accessible through the UCI repository at <https://archive.ics.uci.edu/ml/index.php> and the Kaggle machine learning repository at <https://www.kaggle.com/datasets>. Detailed information about each data set is provided in Table 2. These datasets cover a range of fields of study, including but not limited to health finance, and social sciences domains.

Table 2. List of data sets employed in the experiment.

Code Name	Data Set	Attributes	Instances	Missing Value	Type of Data	Field/ Area
D1	Autistic Disorder	20	292	Yes	Mixed	Health
D2	Primary Tumor	17	339	Yes	Mixed	Health
D3	Bone Marrow Transplant	36	187	Yes	Mixed	Health
D4	Breast Cancer Wisconsin	31	569	No	Mixed	Health
D5	Real Estate Dataset	13	511	Yes	Mixed	Economics
D6	Hungarian Chickenpox	20	522	No	Quantitative	Health
D7	Red Wine	13	178	No	Quantitative	Physic and Chemistry

Continued on next page

Code Name	Data Set	Attributes	Instances	Missing Value	Type of Data	Field/ Area
D8	Adult	14	32561	Yes	Mixed	Social Sciences
D9	Airline Passenger Satisfaction	20	1004	Yes	Mixed	Social Sciences
D10	Australian Credit Approval	14	690	No	Quantitative	Finance
D11	Bank Marketing	20	4119	No	Mixed	Finance
D12	Cylinder Bands	36	541	Yes	Mixed	Physic and Chemistry
D13	Behavior of the Urban Traffic of Sao Paulo, Brazil	17	135	No	Mixed	Traffic
D14	Diabetes Classification	14	390	No	Mixed	Health
D15	Dermatology	34	366	Yes	Mixed	Health
D16	IBM HR Analytics Employee Attrition	34	1470	No	Mixed	Business
D17	Water Quality	21	2001	No	Quantitative	Biology
D18	Forest Type Mapping	26	523	No	Mixed	Biology
D19	Statlog (Heart)	12	270	No	Mixed	Health
D20	Body Fat Prediction	14	252	No	Quantitative	Health

There are several criteria that need to be considered when selecting a dataset. First, the dataset should encompass more than 100 instances. This is crucial, as a lower number of instances poses a higher risk of the network being learned on a limited set of instances during the learning phase. This limitation may result in the retrieval of non-optimal induced logic [16]. This concern aligns with the findings by [28] who observed that the number of instances in the dataset significantly influences the accuracy of the classification task. Insufficient instances may indeed lead to a decrease in accuracy. Second, each dataset must comprise more than 10 attributes. This criterion is based on several reasons, including the fact that the proposed logic consists of a minimum of 10 literals. Additionally, this choice is made to assess the effectiveness of the proposed model in incorporating the concept of optimal attribute selection [15]. Besides that, before all the data sets can be used to be learned by the network, all the data sets must be in bipolar state of 1 and -1 . This is due to the WA method in finding the synaptic weight during the learning phase which can only be significant to bipolar value. Consequently, the normalization of data will be executed by employing a k -means clustering approach to convert the data into bipolar form [12]. In addition, it is crucial to acknowledge that when dealing with benchmark or real-life datasets, two common issues that cannot be avoided are imbalanced data sets and data sets with missing values. Table 3 shows the characterization of all the datasets with regards to the imbalance ratio and missing value rate. The formulation of imbalance ratio is given in Eq (49) [29].

$$\text{Imbalanced Ratio}(\%) = \frac{\text{Frequency of minority class}}{nn} \times 100\% . \quad (49)$$

Note that nn is the total number of entries in the dataset whereby it is equal to the total frequency of minority and majority class. Therefore, in addressing imbalanced datasets, this study adopts the k -fold cross validation technique. Regarding missing values, a strategy involves replacing them with a random bipolar state of 1 and -1 , as proposed by [17]. Lastly, to maintain comparability among the logic mining models, all datasets adhere to the train-split method, where 60% is designated for learning data and 40% for retrieval data, following the approach outlined by [30].

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

Code Name	Missing Value Rate (%)	Frequency of Minority Class	Frequency of Majority Class	Imbalanced Ratio (%)	Majority Class
D1	1.696	142	150	48.630	1
D2	3.687	169	170	49.853	1
D3	1.098	85	102	45.455	1
D4	0.000	212	357	37.258	1
D5	0.070	208	303	40.705	-1
D6	0.000	192	330	36.782	-1
D7	0.000	59	119	33.146	1
D8	0.873	7841	24720	24.081	-1
D9	0.005	459	545	45.717	-1
D10	0.000	307	383	44.493	-1
D11	0.000	451	3668	10.949	-1
D12	5.231	229	312	42.329	-1
D13	0.000	56	79	41.481	-1
D14	0.000	60	330	15.385	-1
D15	0.062	161	205	43.989	-1
D16	0.000	237	1233	16.122	-1
D17	0.000	700	1301	34.983	-1
D18	0.000	242	281	46.272	1
D19	0.000	120	150	44.444	-1
D20	0.000	124	128	49.206	1

7.2. Evaluation criteria

As per Sen and Deokar [31], a confusion matrix for discrete classification is a 2x2 table that assesses the occurrences of the four potential outcomes of a discrete classifier. In a binary classification scenario, there seems to be a focus mention of a 2x2 table. The confusion matrix typically consists of four cells which is True Positive (TP) indicates the number of instances that were correctly predicted as positive by the model, False Positive (FP) is the number of instances that were incorrectly predicted as positive by the model, True Negative (TN) that the number of instances that were correctly predicted as negative by the model and the False Negative (FN) refers to the number of instances that were incorrectly predicted as negative by the model. These outcomes are evaluated during both the learning

and retrieval phases in the U2SATRA model. Hasija et al. [32] suggests examining the learning accuracy (ACC) provides insights into the model performance. If the model demonstrates an increase in ACC , resulting in higher retrieval phase ACC , it is deemed effective for the proposed models. The ACC value can be measured using Eq (50) as shown as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \quad (50)$$

The Sensitivity (SNS) evaluates the accuracy of correctly predicting positive instances in a specific scenario, as noted by [33]. The formulation of SNS is as follows:

$$SNS = \frac{TP}{TP + FN}. \quad (51)$$

Specificity (SPC) is defined as the ratio of the number of samples correctly classified as negative to the total number of actual negative samples [34]. The SPC can be identified as per Eq (52) shown below by Luque et al. [33]:

$$SPC = \frac{TN}{TN + FP}. \quad (52)$$

Negative Predictive Value (NPV) is the ratio of true negatives and overall negatives. The NPV provides correctly identifies instances that do not belong to a particular class. It also helps in assessing the proposed model performance in accurately predicting negative outcomes. The NPV formula is represented as per shown in Eq (53) [35].

$$NPV = \frac{TN}{TN + FN}. \quad (53)$$

The Matthews correlation coefficient (MCC) has attracted increasing attention logic mining because of its strong and dependable performance evaluation capabilities as a classifier, particularly in binary settings [36]. However, its definition can naturally be expanded to accommodate multi-class scenarios. The efficiency of the logic mining process will be assessed using the MCC , which takes into account all elements of a confusion matrix. According to Chicco and Jurman [37], MCC is a valid indicator to evaluate the quality of global model and may applied in various sizes of classes. The MCC is calculated as per Eq (54).

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \quad (54)$$

7.3. Existing baseline models

Given that the primary focus of this study is to assess the performance of the induced logic generated by U2SATRA, our comparison is restricted to methods that specifically produce induced logic. The details for each model are outlined as follows:

- (a) **RA** [10]: This is the pioneer work of logic mining which HornSAT to extract knowledge of the data. Several improvements have been made to ensure this model is comparable with our proposed method. First, to avoid any dimensionality issue, this study only employed

HornSAT with two literals per clause. Second, instead of assigning neuron for each instance, each neuron will be assigned to the attributes.

- (b) **2SATRA** [11]: This model is the first approach on the logic mining that uses logic containing 2 literal per clause or P_{2SAT} . During learning phase, the best logic is formulated by considering the most frequency of $C_i^{(2)}$ that obtained $P_{learn} = 1$. However, the formulation of P_{2SAT} is based on randomized attribute selection. Subsequently, P_{2SAT}^i is generated in the retrieval phase to generalize the information of the dataset.
- (c) **P2SATRA** [14]: This model is the improved conventional 2SATRA with permutation operator that considers possible attribute arrangement in $C_i^{(2)}$. P2SATRA uses P_{2SAT} as a logical rule to represent the relationship of the dataset. In this context, P2SATRA capitalizes on expanding the search space in finding P_{2SAT}^i by rearranging the selected attributes to increase the possibility of obtaining the best induced logic.
- (d) **E2SATRA** [13]: This model is the energy-based logic mining that consider P_{2SAT}^i with global minima energy. E2SATRA capitalizes on $H_{P_{2SAT}}^{\min}$ to ensure the generated P_{2SAT}^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 P_{2SAT}^i . This process can ensure that the P_{2SAT}^i retrieved by E2SATRA is always achieve global minimum energy.
- (e) **3SATRA** [12]: This model is the higher order logic mining approach that capitalize logic contains 3 literals per clause or P_{3SAT} . During learning phase, the best logic is formulated by considering the most frequency of $C_i^{(3)}$ that obtained $P_{learn} = 1$. However, the formulation of P_{3SAT} is based on randomized attribute selection. Regardless of no attribute selection was considered, 3SATRA able to generate P_{3SAT}^i that represent the dataset.
- (f) **S2SATRA** [15]: This model utilized supervised attribute selection using correlation analysis and implement permutation operator to expand the search space of finding optimal P_{2SAT}^i . Interestingly, S2SATRA emphasizes on the attribute selection method and flexible attribute arrangement in the $C_i^{(2)}$ that leads to the higher accuracy value. On the other hand, the final neuron state of S2SATRA is scaled based on Eq (20).

7.4. Experimental design

The compiler is designed to input datasets randomly. In the DHNN models, neurons are represented by bipolar values $(-1, 1)$, which are considered suitable components for the neural network. All experimentation, including the use of real-life datasets, was carried out using the open-source software Dev C++ (Version 5.11). The simulations were performed on a single personal computer to ensure unbiased interpretation of results. To maintain consistency, experiments should use same compiler settings and be conducted on devices with similar processing capabilities. Tables 4–6 show the list of parameters involved in the U2SAT, bCSA, and TDA, respectively. Moreover, Tables 7–13 show the important parameters for all the logic mining models.

Table 4. List of parameters for U2SAT.

Parameter	Parameter Value
Synaptic weight method	Wan Abdullah method [2]
Number of learning (NH)	100
Learning Algorithm	EA [38]
Number of Trials (NT)	100
Tolerance value (tol)	0.01 [39]
Retrieval Algorithm	bCSA
Threshold for D_r (ω)	0.6
Threshold for R_s (γ)	0.3

Table 5. List of parameters for bCSA.

Parameter	Parameter Value
Selection rate (δ_S)	1
Clone rate (δ_C)	0.5 [12]
Mutation rate (δ_M)	1
Number of generations	100

Table 6. List of parameters for TDA.

Parameter	Parameter Value
Number of Hypercube (hc)	1
Overlapping Percentage (hp)	0.3 [25]
Number of clusters (m)	10
Number of selected attributes (R)	10

Table 7. List of parameters for U2SATRA.

Parameter	Parameter Value
Number of neuron (n)	10
Number of clauses (y)	5
Maximum combination	[10, 252]
Attribute Selection	TDA
Number of best logic (q)	5
Learning Algorithm	EA
Number of Trials (NT)	100 [7]
Logical Permutation (P)	10
Retrieval Algorithm	bCSA
Energy analysis	-

Table 8. List of parameters for S2SATRA.

Parameter	Parameter Value
Number of neuron (n)	6
Number of clauses (y)	3
Maximum Combination	-
Attribute Selection	Correlation
Number of best logic (q)	1
Learning Algorithm	ES
Number of Trials (NT)	100 [7]
Logical Permutation (P)	100 [14]
Retrieval Algorithm	-
Energy analysis	HTAF [40]

Table 9. List of parameters for P2SATRA.

Parameter	Parameter Value
Number of neuron (n)	6
Number of clauses (y)	3
Maximum combination	-
Attribute Selection	Random
Number of best logic (q)	1
Learning Algorithm	ES
Number of Trials (NT)	100 [7]
Logical Permutation (P)	100
Retrieval Algorithm	-
Energy analysis	-

Table 10. List of parameters for E2SATRA.

Parameter	Parameter Value
Number of neuron (n)	6
Number of clauses (y)	3
Maximum combination	-
Attribute Selection	Random
Number of best logic (q)	1
Learning Algorithm	ES
Number of Trials (NT)	100 [7]
Logical Permutation (P)	-
Retrieval Algorithm	-
Energy analysis	HTAF [40]

Table 11. List of parameters for 2SATRA.

Parameter	Parameter Value
Number of neuron (n)	6
Number of clauses (y)	3
Maximum combination	-
Attribute Selection	Random
Number of best logic (q)	1
Learning Algorithm	ES
Number of Trials (NT)	100 [7]
Logical Permutation (P)	-
Retrieval Algorithm	-
Energy analysis	-

Table 12. List of parameters for 3SATRA.

Parameter	Parameter Value
Number of neuron (n)	9
Number of clauses (y)	3
Maximum combination	-
Attribute Selection	Random
Number of best logic (q)	1
Learning Algorithm	ES
Number of Trials (NT)	100 [7]
Logical Permutation (P)	-
Retrieval Algorithm	-
Energy analysis	-

Table 13. List of parameters for RA.

Parameter	Parameter Value
Number of neuron (n)	6
Number of clauses (y)	3
Maximum combination	-
Attribute Selection	Random
Number of best logic (q)	1
Learning Algorithm	ES
Number of Trials (NT)	100 [7]
Logical Permutation (P)	-
Retrieval Algorithm	-
Energy analysis	-

8. Findings and analysis

Our main purpose of this experiment is to analyze the performance of the logic mining when pre-processing structure is applied on selecting the attributes. This section presents all the result for each

performance metrics and the performance of the logic mining is indicated by (\uparrow) and (\downarrow) . Notably, (\uparrow) shows the higher value of the metric indicates the better performance of the logic mining model. Moreover, (\downarrow) shows the lower value of the metric indicates the better performance of the logic mining model.

8.1. Accuracy (ACC)

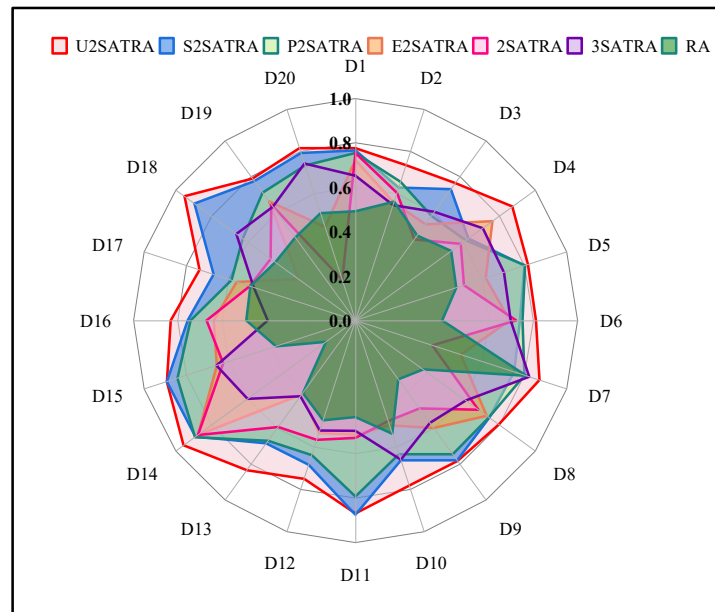


Figure 4. ACC values for all the logic mining models (\uparrow) .

Figure 4 illustrates the ACC values attained by all logic mining models for all 20 datasets. ACC highlights the ability of the logic mining models in retrieving optimal best induced logic that most suitable for the dataset analyzed. The ACC implies how well the retrieved best induced logic is able to represent the dataset with TP and TN outcomes. According to Iwendi et al. [41], high ACC shows the level of correctness of a model to retrieve truth outcomes. Based on the result of the logic mining models, U2SATRA obtained the highest average rank of 7.95. This finding indicates the superior performance of U2SATRA as one of the prominent logic mining models to date. The average ACC of the proposed model is 0.8221 which implies that, on average the distance between the predicted class to the actual class of all the datasets is only approximate to 17.79%. Thus, justification will be provided to emphasize several winning points of the U2SATRA. One of the possible reason why U2SATRA works well due to the mutual interaction of dynamic r in U2SATRA in DHNN. Dynamic values of r in r S2SAT provides better logical flexibility which resulting to better mapping of the SAT formula with the analyzed dataset that will minimize the FP and FN. This also highlights the ability of U2SATRA to generalize the datasets better than baseline logic mining models. Additionally, U2SATRA offers multiple search space dimensions by having multi-unit DHNNs in the learning phase. Hence, this creates a pathway to expand the search space in locating the best induced logic that will eventually maximize the values of TP and TN. Aligned to this finding, the multiple DHNNs in U2SATRA is proven to be effective. The performance of all the existing benchmark model far below

the U2SATRA model. This is because the systematic SAT formulas have random distribution of negative literals in respective logical rules. Zamri et al. [21] stated that random distribution of negative literals deteriorates the flexibility of the SAT formula to represent the dataset due to its structural rigidity. Thus, resulting to the failure to be generalized when dealing with real-life applications. Consequently, all the existing benchmark model attained suboptimal ACC values, especially in E2SATRA, 2SATRA, 3SATRA, and RA. This observation has a good agreement with the statement given by Zamri et. al [21]. Ultimately, this shed new light on how successful the proposed logic mining model in extracting optimal best induced logic with high ACC value.

Overall, all models except for E2SATRA performed 100% superior to RA due to the consideration of the attributes selection method. Random attributes selection method contributes to the lack of interpretability of the learned logical rule in DHNN. Thus, RA tends to misclassify the outcomes which leads to high FP and FN . This observation can also be seen in the results of 2SATRA. In comparison to S2SATRA, the attributes selected as based on the associations between variables in the dataset. Hence, DHNN is able to learn significant attributes which leads to higher ACC of the logic mining models. At one instance, U2SATRA is able to acquire almost 100% ACC for D18 with $ACC = 0.9522$ equivalent to 95.22%. This finding implies that the proposed model is able to correctly predict the testing data entries for majority TP and TN . Based on an observation in Figure 4, there is one dataset where U2SATRA model is not able to win over the ACC value. The dataset is D11 which consists of the higher percentage of missing values (5%) as referred to Table 3. Therefore, the TDA was not fully utilized because the behavior of the variables cannot be analyzed correctly due to inaccurate information. The exact mapping of attributes for D11 decreases the possibility of removing irrelevant features of the dataset. Thus, resulting in lower ACC values as compared to S2SATRA. Despite competitive performance, U2SATRA model remains superior with 19 wins and no dataset with ACC value less than 70%. Friedman test is conducted with the null hypothesis, H_0 is no performance differences between all logic mining models in terms of ACC . By observing the p -value, the H_0 is rejected. This implies that the values of ACC in Figure 4 are significant whereby all logic mining models do not have equal performance in attaining the values of TP and TN . The superiority of U2SATRA in terms of ACC is statistically proven.

8.2. Sensitivity (SNS)

SNS results are used to identify how sensitive all logic mining models are by minimizing FN values. The proposed U2SATRA showed superiority in Figure 5 with 17 wins and the highest average for all datasets. Overall, this observation exhibited the ability of the proposed model in maximizing SNS which indicates that the model is highly sensitive when detecting positive outcomes. Worth mentioning that, U2SATRA achieved high SNS for health (D1, D2, D3, D4, D9, D14, and D19) and financial (D5, D10, and D11) datasets. The classification of health and financial datasets should be able to minimize FN because they are contributing to type II error which is less tolerable or more costly [42]. There are several justifications to support these findings. In comparison to existing logic mining models, U2SATRA generates optimal best logic which resulting to an effective testing phase of U2SATRA. The best logic captured the overall trend of the learning data with consideration of TP and TN values, which positively impacted the quality of the best induced logic produced. By this approach, the positive outcomes of the learning data are carefully represented and preserved.

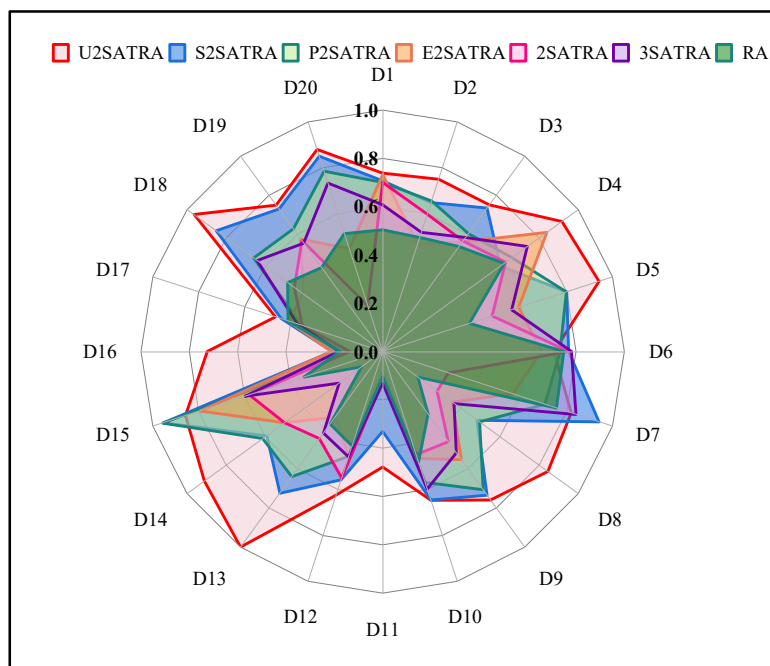


Figure 5. SNS values for all the logic mining models (\uparrow).

Figure 6 represents the ratio of r frequency based on the U2SATRA retrieved best induced logic for the 20 datasets. In other words, these figures showed the frequency of which r the best induced logic are selected from. As presented in the figure, the structure of r S2SAT with $r = 0.1$ has the highest r frequency ratio. This implies two observations: First, the majority of the mentioned datasets have high compatibility with the structure of r S2SAT when $r = 0.1$ in both learning and retrieval phase of the proposed U2SATRA. Second, the systematic 2SAT clauses in r S2SAT lead to higher satisfiable property. Conclusively, these observations lead to the increase of getting positive outcomes. This is beneficial in maximizing the values of TP thus reduces FN , especially when dealing with datasets that have more positive outcomes like D4. This also explains why the existing logic mining models such as 3SATRA is able to obtain slightly higher SNS values than U2SATRA for one dataset. This is because 3SATRA capitalized the systematic SAT logical rule of 3SAT which increases the probability of satisfied interpretations. The Friedman test is conducted with the null hypothesis, and H_0 is performance differences between all logic mining models in terms of SNS. Since the p -value attained is less than 0.05, then H_0 is rejected which showed that the performance of all logic mining models in terms of SNS is significant and not equal. Therefore, this shows the superiority of U2SATRA in terms of SNS being acknowledged.

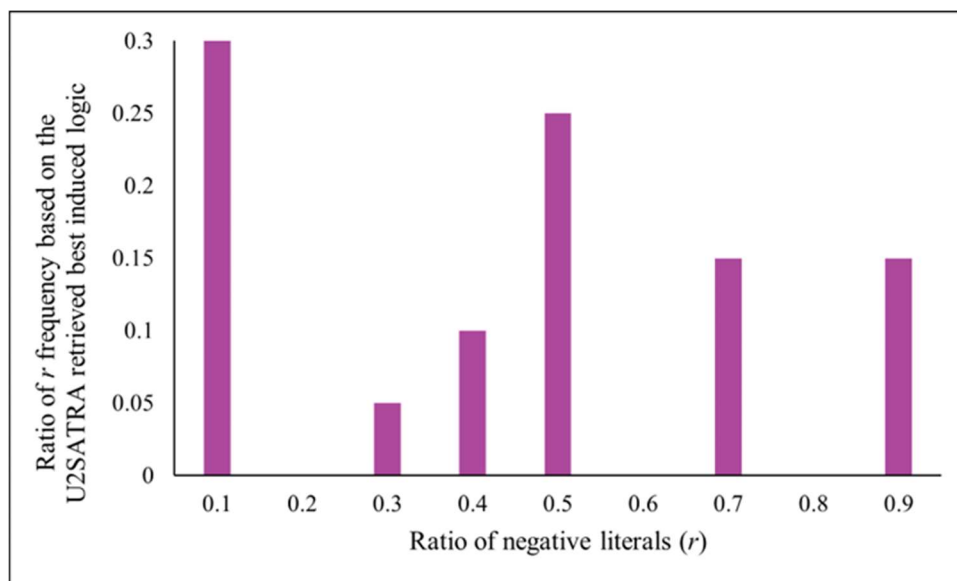


Figure 6. Ratio of r frequency based on the U2SATRA retrieved best induced logic for the 20 datasets.

8.3. Specificity (SPC)

The result of SPC for all logic mining models in analyzing 20 datasets is visualized in Figure 7. According to an existing work by Nouredin et al. [43], the values of SPC indicates the correctly identified TN by the model. Note that, 13 of the datasets utilized in this paper consists of majority class with -1 entries than 1 as shown in Table 3. Thus, high values of TN are preferable. Although not all datasets are imbalanced towards -1 , the values of SPC are significant to exhibit the ability of the logic mining model in retrieving TN , rather than misclassified a lot of negative outcomes as positive. The proposed U2SATRA obtained 19 wins, whereby 50% of the datasets attained perfect values of $SPC=1$. Generally, the structural component of a SAT formula in DHNN is crucial because the relationship between each literal represents the neuron connections in the network. Hence, the synaptic weights of the DHNN highly impacted the quality of the induced logic produced from the local field computation. In terms of maximizing SPC , the CAM feature of DHNN should have the consistent magnitude with the variation of negativity. These qualities of CAM are demanded to ensure the production of optimal best induced logic to effectively captures the negative entries of the dataset resulting to the negative outcomes. Subsequently, the model will ascertain high value of TN which correspond to maximization of SPC . One of the reasons why U2SATRA remains superior in terms of SPC as compared to other logic mining models is due to the structural components of $rS2SAT$. The structure of $rS2SAT$ offered consistent magnitude and negative variations in CAM by having systematic 2SAT clauses with dynamic r . The systematic 2SAT clauses in SAT is able to offer same satisfiable property and resulting to the consistent synaptic weight values. This will prevent disruption during the retrieval phase of the DHNN, thereby increasing the probability of identifying the best induced logic. This observation can be seen through the results of SPC for the following datasets of D2, D3, D4, D5, D7, D13, D15, D16, D18, and D20. These datasets achieved the maximum SPC values for U2SATRA model. The combination of systematic 2SAT clauses and dynamic r provided better logical representation in capturing the entries with negative outcome.

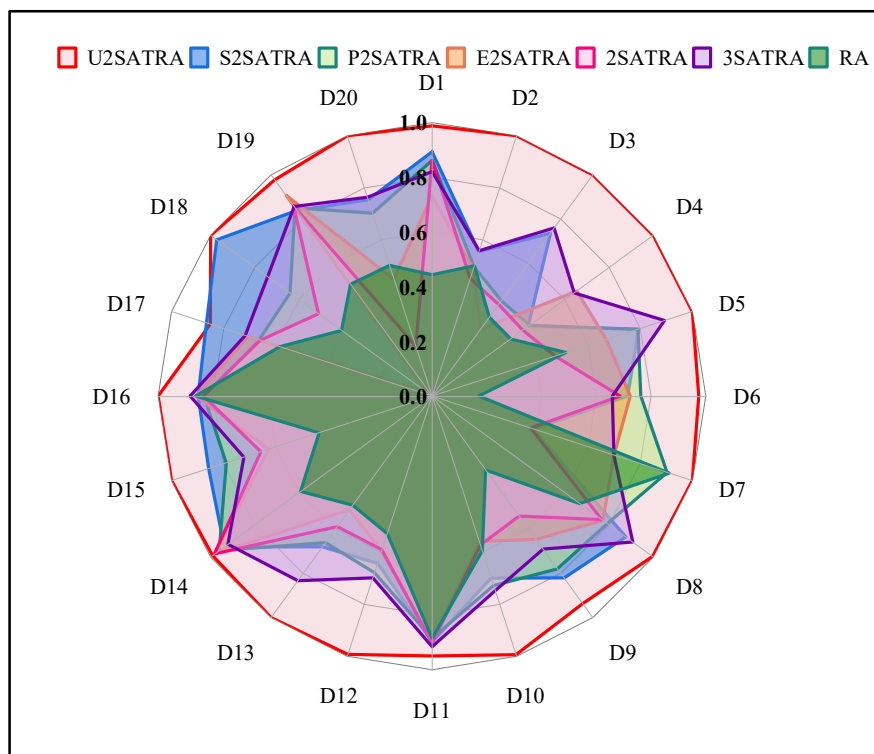


Figure 7. *SPC* values for all the logic mining models (\uparrow).

However, the component of a logical structure is insignificant without an optimizer in the retrieval phase of DHNN. The proposed bCSA in U2SATRA exhibited good performance in offering diversification of the final neuron state which directly impacted to the production of best induced logic. This can be seen through the superiority of U2SATRA in terms of *SPC* values more than 90%. According to Ong and Zainuddin [44], the main issue with the conventional DHNN is that the network may result in biased local field because the nature of DHNN is directly memorized in the final neuron states without generating a new state. Even when the synaptic weight varies, the tendency of repetitive induced logic cannot be avoided. To make things worse, if the logical structure only consist of positive clauses, the local field computation will be biased towards only the positive entries of the dataset. Thus, this creates an unfavourable situation to produce best induced logic that maps to the negative outcomes of the dataset. In the retrieval phase of U2SATRA, bCSA generated diversified final neuron state to counter this problem. In light of this, bCSA is able to produce optimal best induced logic with high *SPC*. Overall, the values of *SPC* are highly influenced by the structural components of the SAT logical rule and the diversification of the final neuron state. The Friedman test is conducted with the null hypothesis, and H_0 has no performance differences between all logic mining models in terms of *SPC*. The obtained *p*-value is less than 0.05, so the H_0 is rejected. This implies that values of *SPC* in Figure 7 are significant, as the logic mining models do not have equal performance in maximizing the values of *TN*, which leads to high *SPC*. Hence, the superiority of the proposed logic mining model is acknowledged as compared to other existing logic mining models.

8.4. Negative Predictive Value (NPV)

Based on Figure 8, the proposed U2SATRA model attained most wins as compared to other baseline models with 65% wins out of all datasets. An observation by [35], NPV values guide researchers in visualizing the reliability of a model in classifying the model as negative. In this case, high NPV implies the proposed model ability to locate correct negative outcome of each dataset. With the most wins, U2SATRA outweighs other models by obtaining an average of 0.8591 or 85.91% and one dataset (D13) that is able to obtain $NPV = 1$. When NPV reached its maximum values, this indicates that U2SATRA is able to minimize the values of FN with no positive entries that is incorrectly classified. One of the possible reasons is due to the flexibility of the negative state in the systematic structure of $rS2SAT$. The flexibility of the negative state is advantageous as compared to other SATs in other logic mining models because $rS2SAT$ is able to capture or make sense of the dataset due to the flexibility of the negative state. The dynamic ratio of r provides distinct outcome that will affect the values of TP and TN . When $NPV = 1$, the retrieved best induced logic by U2SATRA increases its expressivity and interpretability power via the ratio of r in $rS2SAT$. This approach is beneficial towards datasets with more negative outcomes. This observation can be seen through D13, which achieved maximum NPV whereby the majority class consists of -1 entries. This approach provides wider search space for the proposed logic mining model in searching for the most optimal best induced logic with maximum values of TP and TN . On the contrary, all existing logic mining models considered randomized distribution of positive and negative literals in respective logic mining models. This observation can be seen from the comparison of NPV and ACC values. When NPV is high, high values of TP is ascertained. Notably, U2SATRA showed the best performance in terms of both NPV and ACC for datasets D3, D4, D6, D7, D9, D10, D13, D14, D16, D17, D18, D19, and D20. This implies the proposed model ability to generate best induced logic with prediction close to the actual outcome without having to trade-off positive or negative outcomes.

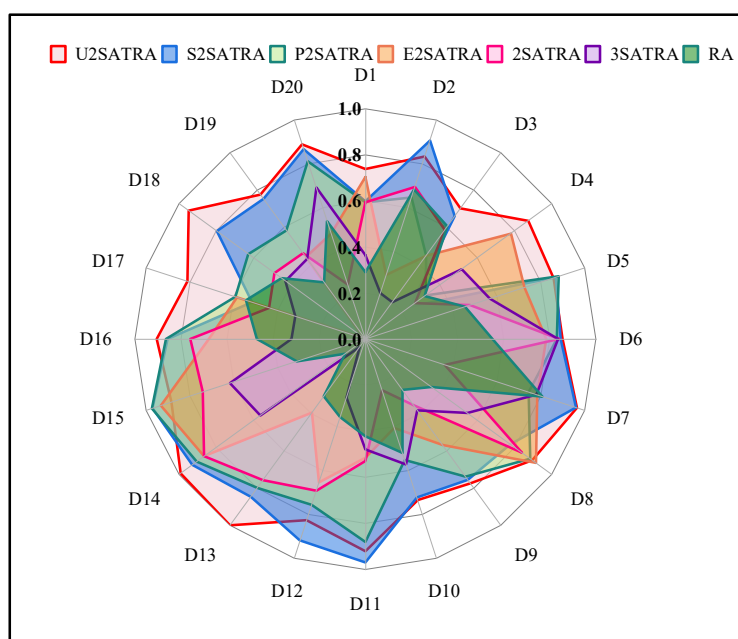


Figure 8. NPV values for all the logic mining models (\uparrow).

The proposed U2SATRA attained the highest average of NPV of 0.8591 which magnifies the capability of U2SATRA not only to be successful in minimizing FN but also attain high values of TN . Despite the nature of r in $rS2SAT$ neglects the values of $r=1$ (all negative literals), the proposed TDA in the pre-processing phase of U2SATRA works efficiently in preserving negative entries that contributed to the negative outcome. The mutual interaction between TDA and $rS2SAT$ in U2SATRA exhibited a harmony combination in reducing FP and FN . Significantly, RA achieved the lowest average (0.4336) of NPV which highlighted the fact that the model not able to attain high values of TN . This describes the low competency of the random attribute selection and distribution of negative literals less contributed to the negative outcome of the dataset. Overall, the U2SATRA model outperforms all baseline models in terms of NPV with no result of $TN=0$. The Friedman test has been conducted with the null hypothesis, and H_o has no performance differences between all logic mining models in terms of NPV . The achieved p -value of NPV is less than 0.05. As a result, the H_o is rejected, which concludes that all logic mining models do not have equal performances in terms of NPV . The values of NPV achieved for all datasets are statistically significant. In light with this finding, the superiority of U2SATRA is acknowledged in minimizing FN as compared to the existing logic mining models.

8.5. Mathews Correlation Coefficient (MCC)

Figure 9 illustrates the results of MCC values attained by all logic mining models for all 20 datasets. Notably, U2SATRA achieved the highest number of wins with 18 and 2 losses. The average MCC values gained by U2SATRA for all datasets utilized is the highest among other benchmark models. This resulting to U2SATRA in achieving the highest rank with 7.85, which exhibited obvious superiority in comparison to all existing logic mining models. The closest performance in terms of MCC after U2SATRA is S2SATRA with the second-best average of 0.4142. Moreover, the worst model is RA with the lowest average of 0.1353. The motive of MCC analysis is to identify whether the performance of the logic mining model is similar, worse, or better than the random classifier model. According to Chicco and Jurman [37], the MCC values for any model with equal or lower than 0.14 is considered having the same or worse performance than a random classifier. From Figure 9, the number of datasets that achieved MCC values with equal or lower than 0.14 for U2SATRA is zero, S2SATRA is 2, followed by 3SATRA is five, P2SATRA is seven, and E2SATRA is nine. Based on this, 2SATRA and RA are categorized as poor-performed logic mining models with majority of the datasets performed equal or less than a random classifier model with 13 and 15 datasets, respectively. One of the visible reasons is the dynamic ratio of the negative literals in the SAT formula considered in the U2SATRA. As mentioned, MCC quantifies the right balance between the values of TP and TN . Thus, the retrieved best induced logic must be able to represent both positive and negative outcomes of the dataset. In other words, the best induced logic must effectively capture the retrieval data entries that leads to both outcomes. The logical structure utilized in U2SATRA have different distribution of negative literals with regards to the r . Hence, during the retrieval phase, the best induced logic with $r=0.1$ has a higher tendency to produce more positive outcomes, leading to high values of TP . Correspondingly, with a higher value of r , the logic exhibits a greater tendency to generate more negative outcomes, which correlates with high values of TN . This magnified the ability of the proposed U2SATRA to maximize both values of TP and TN which then later resulting to high MCC . However, all existing logic mining models utilize a random distribution of negative literals, which allows only for the same pattern of CAM. This uniform pattern of CAM promotes inflexibility in capturing the

entries of the dataset, which results in producing repetitive induced logic.

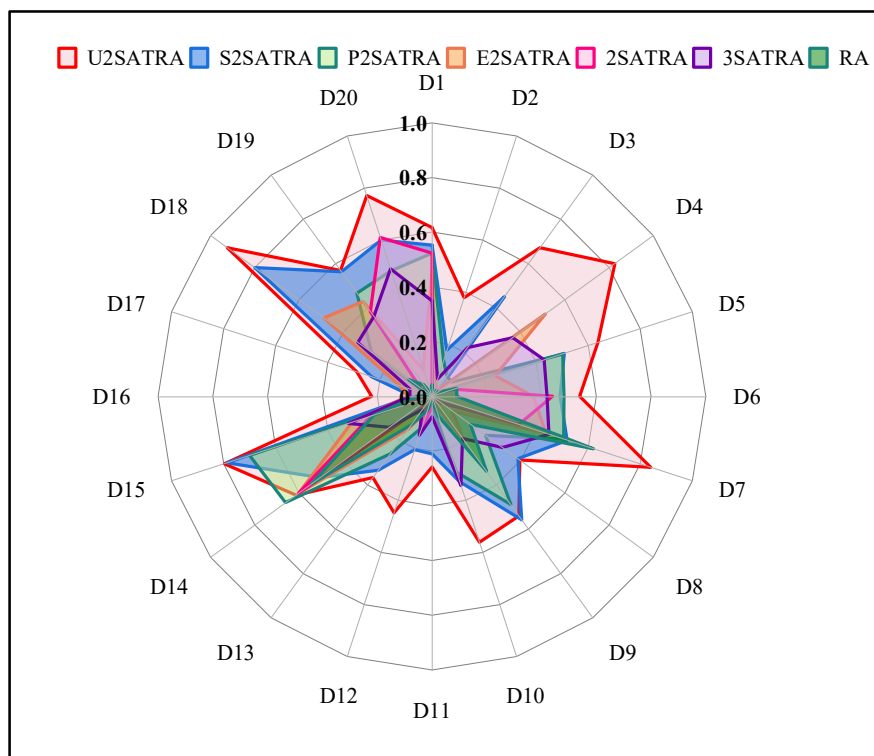


Figure 9. *MCC* values for all the logic mining models (\uparrow).

The superiority of U2SATRA in terms of *MCC* can also be explained through the existence of the best logic in the learning phase of the model. First, all possible structures of r S2SAT will be generated with respect to r and the 2SAT clauses with a range of [10, 252]. Each logical structure of r S2SAT has the potential to become the best logic by evaluating the values of TP , TN , FP , and FN when embedding the entries of the learning data into each logical rule. Second, the extraction of the best logic is based on the highest *MCC* values attained for each logical structure. The 5 best logics with the highest *MCC* for each r will be generated for each dataset. The reason why the value of *MCC* is being considered is to ensure that the best logic is able to represent entries in the retrieval data that leads to the high *MCC* values. These features in U2SATRA create various pathways to expand the search space in locating the best induced logic that will eventually maximize the values of *MCC*. Conjointly, this leads to higher *MCC* and attains better performance than a random classifier. On the contrary, all existing logic mining models capitalize only one logic in the learning phase through the generation of single best logic to represent the learning data. There is a higher tendency to produce repetitive induced logic by offering only one logic. Thus, the quality of the best induced logic produced will be affected because there are no variations of logic in the learning phase. Additionally, the extraction of best logic disregards the negative outcomes because it is solely based on the highest frequency of satisfied clauses from the learning data entries that only leads to positive outcomes. Hence, this explained why most of the existing logic mining models obtained low *MCC* values due to the retrieved best induced logic has poor interpretability property. The Friedman test is conducted with the null hypothesis, H_0 has no performance differences between all logic mining models in terms of *MCC*, and the p -value attained

is less than 0.05. Thus, the H_0 is rejected, which exhibited that the performance of all logic mining models in terms of MCC is significant and not equal. Overall, the performance of the U2SATRA in retrieving induced logic with the right balance of TP and TN is recognized as compared to other existing logic mining models.

8.6. Qualitative analysis

The performance evaluation of the U2SATRA model across 20 datasets provides valuable insights into its effectiveness in handling various datasets. Table 14 presents a comprehensive overview of the U2SATRA performance across five performance metrics. Upon analysis of the results, it is evident that U2SATRA exhibits strong performance across several key metrics, including ACC , SNS , SPC , NPV , and MCC . Of the 20 datasets evaluated, U2SATRA emerges as the top performer in most of the datasets across these metrics. This finding highlights the robustness of U2SATRA in effectively learning and adapting to diverse datasets. By consistently outperforming other models across multiple metrics, U2SATRA demonstrates its capability to generalize and handle various classification tasks with high accuracy and reliability. The effectiveness of the U2SATRA process can be attributed to several factors. First, U2SATRA employs an adaptive learning mechanism that enables it to effectively learn from the dataset. The introduction of the new computation method for determining the best logic during the learning phase of U2SATRA has had a significant impact on the induced logic retrieved during the retrieval phase. This is because the best logic is learned based on the specific performance metrics. For example, when considering the ACC metric, the selected P_{rS2SAT}^b must achieve a high ACC during the learning phase to be designated as the best logic. Subsequently, the selected P_{rS2SAT}^b is influenced by the ACC factor, increasing the likelihood of U2SATRA retrieving induced logic with a high ACC because it has learned P_{rS2SAT}^b with a high ACC . This applies to each of the metrics, which is the reason why U2SATRA proposed multi-unit DHNNs. This principle applies to each metric, which is why U2SATRA proposes multi-unit DHNNs. In this context, U2SATRA is able to produce multiple sets of induced logic that correspond to the metrics. Second, U2SATRA utilizes an effective attribute selection method known as TDA in the pre-processing phase, which is crucial in capturing the underlying patterns and relationships within the data. TDA helps in mapping the structure of dataset by clustering the attributes that exhibit similar behavior. This prevents U2SATRA from selecting redundant attributes that have a similar impact on the dataset. By extracting informative attributes from the input data, U2SATRA enhances its ability to make accurate predictions for the dataset. Additionally, the multi-objective function in the retrieval phase enhances the diversification of the final solutions produced. This is achieved by promoting diversity among the negative states in each solution. Consequently, this ensures that the final solutions produced exhibit high dissimilarity. On the other hand, U2SATRA is able to achieve good performance due to its logical structure, $rS2SAT$. $rS2SAT$ has a dynamic ratio, denoted as r , where the distribution of negative literals is pre-determined. Therefore, this increases the search space of U2SATRA by allowing it to find the best logical structure that corresponds to the dataset. Overall, the qualitative analysis contributes to the capability of U2SATRA to generalize and handle various classification tasks with high accuracy and reliability, highlighting its potential for real-world applications in diverse domains.

Table 14. Overall performance of the datasets in U2SATRA for all the performance metrics.

Data Set	<i>ACC</i>	<i>SNS</i>	<i>SPC</i>	<i>NPV</i>	<i>MCC</i>
D1	/	/	/	X	/
D2	/	/	/	X	/
D3	/	/	/	/	/
D4	/	/	/	/	/
D5	/	/	/	X	/
D6	/	X	/	/	/
D7	/	X	/	/	/
D8	/	/	/	X	/
D9	/	/	/	/	X
D10	/	/	/	/	/
D11	X	/	/	X	/
D12	/	/	/	X	/
D13	/	/	/	/	/
D14	/	/	/	/	X
D15	/	X	/	X	/
D16	/	/	/	/	/
D17	/	/	X	/	/
D18	/	/	/	/	/
D19	/	/	/	/	/
D20	/	/	/	/	/

9. Conclusions

We have extended the work of existing logic mining by making significant improvements in the pre-processing, learning, and retrieval phases by considering various important factors. A major contribution of the proposed model with other logic mining method is for the logical structure and the computation of best logic. U2SATRA was the first logic mining model that embedded a weighted feature in the systematic SAT, which is r S2SAT in representing the neurons in DHNN. Interestingly, U2SATRA also modified the mechanism of best logic in the existing reverse analysis method. The modified best logic is the strong logic that have both true and false classifications as compared with the P_{learn} . The U2SATRA model was experimented with 20 repository real life datasets and compared with six states of the art logic mining models. The result shows the superiority of U2SATRA as compared to the other baseline methods. Subsequently, the proposed U2SATRA obtained optimal average values of *ACC*, *SNS*, *SPC*, *NPV*, and *MCC*. We also performed the Friedman test to ensure that there is significant difference for U2SATRA with all the logic mining models. Based on the findings, U2SATRA outperformed all the baseline methods by winning all the 5 metrics in the average

rank. The metrics is able to achieve highest average rank with *ACC* (7.95), *SNS* (7.55), *SPC* (7.93), *NPV* (7.50), and *MCC* (7.85). Based on the obtained results, it proves that U2SATRA exhibits significant performance across all tested performance metrics. Our research provides opportunities for researchers to extend the application of the proposed logic mining model in the fields of discrete neural networks such as the fuzzy neural network [45,46] and multidimensional neural network [47]. This can provide new insights in solving real life classification problems.

Author contributions

Nurul Atiqah Romli: Conceptualization, Project Administration, Writing – Original Draft; Nur Fariha Syaquina Zulkepli: Resources; Mohd Shareduwan Mohd Kasihmuddin: Funding Acquisition; Nur Ezlin Zamri: Validation; Nur ‘Afifah Rusdi: Methodology; Gaeithry Manoharam: Writing – Review & Editing; Mohd. Asyraf Mansor: Supervision; Siti Zulaikha Mohd Jamaludin: Formal Analysis, Investigation; Amierah Abdul Malik: Visualization. All authors have read and approved the final version of the manuscript for publication.

Use of AI tools declaration

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

Acknowledgments

This research is financially supported by the Ministry of Higher Education Malaysia for the financial support of the Fundamental Research Grant Scheme (FRGS) with Project Code: FRGS/1/2022/STG06/USM/02/6 and Universiti Sains Malaysia. All the authors gratefully acknowledge the support.

Conflict of interest

The authors declare no conflict of interest.

References

1. J. J. Hopfield, D. W. Tank, “Neural” computation of decisions in optimization problems, *Biol. Cybern.*, **52** (1985), 141–152. <https://doi.org/10.1007/BF00339943>
2. 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>
3. 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).
4. 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).
5. 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>

6. S. Sathasivam, M. A. Mansor, A. I. M. Ismail, S. Z. M. Jamaludin, M. S. M. Kasihmuddin, M. Mamat, Novel random k satisfiability for $k \leq 2$ in hopfield neural network, *Sains Malays.*, **49** (2020), 2847–2857. <https://doi.org/10.17576/jsm-2020-4911-23>
7. S. A. Karim, N. E. Zamri, A. Alway, M. S. M. Kasihmuddin, A. I. M. Ismail, M. A. Mansor, et al., Random satisfiability: A higher-order logical approach in discrete hopfield neural network, *IEEE Access*, **9** (2021), 50831–50845. <https://doi.org/10.1109/ACCESS.2021.3068998>
8. Y. Guo, M. S. M. Kasihmuddin, Y. Gao, M. A. Mansor, H. A. Wahab, N. E. Zamri, et al., YRAN2SAT: A novel flexible random satisfiability logical rule in discrete hopfield neural network, *Adv. Eng. Softw.*, **171** (2022), 103169. <https://doi.org/10.1016/j.advengsoft.2022.103169>
9. N. E. Zamri, S. A. Azhar, S. S. M. Sidik, M. A. Mansor, M. S. M. Kasihmuddin, S. P. A. Pakruddin, et al., Multi-discrete genetic algorithm in hopfield neural network with weighted random k satisfiability, *Neural Comput. Appl.*, **34** (2022) 19283–19311. <https://doi.org/10.1007/s00521-022-07541-6>
10. S. Sathasivam, W. A. T. Wan Abdullah, Logic mining in neural network: Reverse analysis method, *Computing*, **91** (2011), 119–133. <https://doi.org/10.1007/s00607-010-0117-9>
11. L. C. Kho, M. S. M. Kasihmuddin, M. A. Mansor, S. Sathasivam, Logic mining in league of legends, *Pertanika J. Sci. Technol.*, **28** (2020).
12. N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, A. Alway, S. Z. M. Jamaludin, S. A. Alzaemi, Amazon employees resources access data extraction via clonal selection algorithm and logic mining approach, *Entropy*, **22** (2020), 596. <https://doi.org/10.3390/E22060596>
13. 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). <https://doi.org/10.3390/e23010040>
14. S. Z. M. Jamaludin, M. A. Mansor, A. Baharum, M. S. M. Kasihmuddin, H. A. Wahab, M. F. Marsani, Modified 2 satisfiability reverse analysis method via logical permutation operator, *Comput. Mater. Contin.*, **74** (2023), 2853–2870. <https://doi.org/10.32604/cmc.2023.032654>
15. 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>
16. S. Z. M. Jamaludin, N. A. Romli, M. S. M. Kasihmuddin, A. Baharum, M. A. Mansor, M. F. Marsani, Novel logic mining incorporating log linear approach, *J. King Saud Univ.-Comput. Inf. Sci.*, **34** (2022), 9011–9027. <https://doi.org/10.1016/j.jksuci.2022.08.026>
17. N. A. Rusdi, M. S. M. Kasihmuddin, N. A. Romli, G. Manoharam, M. A. Mansor, Multi-unit discrete hopfield neural network for higher order supervised learning through logic mining: Optimal performance design and attribute selection, *J. King Saud Univ.-Com.*, **35** (2023), 101554. <https://doi.org/10.1016/j.jksuci.2023.101554>
18. G. Manoharam, M. S. M. Kasihmuddin, S. N. F. M. A. Antony, N. A. Romli, N. A. Rusdi, S. Abdeen, et al., Log-linear-based logic mining with multi-discrete hopfield neural network, *Mathematics*, **9** (2023), 2121. <https://doi.org/10.3390/math11092121>
19. A. Alway, N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, S. Z. M. Jamaludin, M. F. Marsani, A novel hybrid exhaustive search and data preparation technique with multi-objective discrete hopfield neural network, *Decis. Anal.*, **9** (2023), 100354. <https://doi.org/10.1016/j.dajour.2023.100354>

20. 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. 2024. <https://doi.org/10.1016/j.eswa.2023.122307>
21. N. E. Zamri, S. A. Azhar, M. A. Mansor, A. Alway, M. S. M. Kasihmuddin, Weighted random k satisfiability for k=1,2 (r2SAT) in discrete hopfield neural network, *Appl. Soft Comput.*, **126** (2022), 109312. <https://doi.org/10.1016/j.asoc.2022.109312>
22. D. Bajusz, A. Rácz, K. Héberger, Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J. Cheminformatics*, **7** (2015), 1–13. <https://doi.org/10.1186/s13321-015-0069-3>
23. L. N. De Castro, F. J. von Zuben, The clonal selection algorithm with engineering applications, *Gecco*, **2000** (2000), 36–39.
24. Y. Zhu, W. Li, T. Li, A hybrid artificial immune optimization for high-dimensional feature selection, *Knowl.-Based Syst.*, **260** (2023), 11011. <https://doi.org/10.1016/j.knsys.2022.110111>
25. G. Singh, F. Mémoli, G. Carlsson, *Topological methods for the analysis of high dimensional data sets and 3D object recognition*, In: 4th symposium on point based graphics, PBG@Eurographics 2007, 2007.
26. A. D. Smith, P. Dłotko, V. M. Zavala, Topological data analysis: Concepts, computation, and applications in chemical engineering, *Comput. Chem. Eng.*, **146** (2021), 107202. <https://doi.org/10.1016/j.compchemeng.2020.107202>
27. Y. Chen, I. Volic, Topological data analysis model for the spread of the coronavirus, *Plos One*, **16** (2021), e0255584. <https://doi.org/10.1371/journal.pone.0255584>
28. O. Kwon, J. M. Sim, Effects of data set features on the performances of classification algorithms, *Expert Syst. Appl.*, **40** (2013), 1847–1857. <https://doi.org/10.1016/j.eswa.2012.09.017>
29. J. Dou, Y. Song, G. Wei, Y. Zhang, Fuzzy information decomposition incorporated and weighted Relief-F feature selection: When imbalanced data meet incompleteness, *Inf. Sci.*, **584** (2022), 417–432. <https://doi.org/10.1016/j.ins.2021.10.057>
30. K. Jha, S. Saha, Incorporation of multimodal multiobjective optimization in designing a filter based feature selection technique, *Appl. Soft Comput.*, **98** (2021), 106823. <https://doi.org/10.1016/j.asoc.2020.106823>
31. S. Sen, A. V. Deokar, Toward understanding variations in price and billing in US healthcare services: A predictive analytics approach, *Expert Syst. Appl.*, **209** (2022), 118241. <https://doi.org/10.1016/j.eswa.2022.118241>
32. S. Hasija, P. Akash, M. B. Hemanth, A. Kumar, S. Sharma, A novel approach for detection of COVID-19 and Pneumonia using only binary classification from chest CT-scans, *Neurosci. Inform.*, **2** (2022), 100069. <https://doi.org/10.1016/j.neuri.2022.100069>
33. A. Luque, A. Carrasco, A. Martín, A. de las Heras, The impact of class imbalance in classification performance metrics based on the binary confusion matrix, *Pattern Recognit.*, **91** (2019), 216–231. <https://doi.org/10.1016/j.patcog.2019.02.023>
34. F. Amin, M. Mahmoud, Confusion matrix in binary classification problems: A step-by-step tutorial, *J. Eng. Res.*, **6** (2022). <https://doi.org/10.21608/erjeng.2022.274526>
35. M. Ohsaki, P. Wang, K. Matsuda, S. Katagiri, H. Watanabe, A. Ralescu, Confusion-matrix-based kernel logistic regression for imbalanced data classification, *IEEE Trans. Knowl. Data Eng.*, **29** (2017), 1806–1819. <https://doi.org/10.1109/TKDE.2017.2682249>

36. J. Gorodkin, Comparing two K-category assignments by a K-category correlation coefficient, *Comput. Biol. Chem.*, **28** (2004), 367–374. <https://doi.org/10.1016/j.compbiolchem.2004.09.006>
37. 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), 1–13. <https://doi.org/10.1186/s12864-019-6413-7>
38. V. Someetheram, M. F. Marsani, M. S. M. Kasihmuddin, N. E. Zamri, S. S. M. Sidik, S. Z. M. Jamaludin, et al., Random maximum 2 satisfiability logic in discrete hopfield neural network incorporating improved election algorithm, *Mathematics*, **10** (2022), 4734. <https://doi.org/10.3390/math10244734>
39. S. Abdeen, M. S. M. Kasihmuddin, N. E. Zamri, G. Manoharam, M. A. Mansor, N. Alshehri, S-type random k satisfiability logic in discrete hopfield neural network using probability distribution: Performance optimization and analysis, *Mathematics*, **11** (2023), 984. <https://doi.org/10.3390/math11040984>
40. 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>
41. C. Iwendi, K. Mahboob, Z. Khalid, A. R. Javed, M. Rizwan, U. Ghosh, Classification of COVID-19 individuals using adaptive neuro-fuzzy inference system, *Multimedia Syst.*, **2022**, 1–15. <https://doi.org/10.1007/s00530-021-00774-w>
42. M. Herland, R. A. Bauder, T. M. Khoshgoftaar, The effects of class rarity on the evaluation of supervised healthcare fraud detection models, *J. Big Data*, **6** (2019), 1–33. <https://doi.org/10.1186/s40537-019-0181-8>
43. M. Nouredin, E. Truong, J. A. Gornbein, R. Saouaf, M. Guindi, T. Todo, et al., MRI-based (MAST) score accurately identifies patients with NASH and significant fibrosis, *J. Hepatol.*, **76** (2022), 781–787. <https://doi.org/10.1016/j.jhep.2021.11.012>
44. P. Ong, Z. Zainuddin, Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction, *Appl. Soft Comput.*, **80** (2019), 374–386. <https://doi.org/10.1016/j.asoc.2019.04.016>
45. H. L. Li, J. Cao, C. Hu, H. Jiang, A. Alsaedi, Synchronization analysis of nabla fractional-order fuzzy neural networks with time delays via nonlinear feedback control, *Fuzzy Set. Syst.*, **475** (2024), 108750. <https://doi.org/10.1016/j.fss.2023.108750>
46. H. L. Li, J. Cao, C. Hu, L. Zhang, H. Jiang, Adaptive control-based synchronization of discrete-time fractional-order fuzzy neural networks with time-varying delays, *Neural Netw.*, **168** (2023), 59–73. <https://doi.org/10.1016/j.neunet.2023.09.019>
47. J. Cao, K. Udhayakumar, R. Rakkiyappan, X. Li, J. Lu, A comprehensive review of continuous-/discontinuous-time fractional-order multidimensional neural networks, *IEEE Trans. Neural Netw. Learn. Syst.*, **34** (2021), 5476–5496. <https://doi.org/10.1109/TNNLS.2021.3129829>

