mathematics

MDPI

Article

Supervised Learning Perspective in Logic Mining

Mohd Shareduwan Mohd Kasihmuddin [1], Siti Zulaikha Mohd Jamaludin [1], Mohd. Asyraf Mansor [2,*], Habibah A. Wahab [3] and Siti Maisharah Sheikh Ghadzi [3]

[1] School of Mathematical Sciences, Universiti Sains Malaysia, George Town 11800, Malaysia; shareduwan@usm.my (M.S.M.K.); szulaikha.szmj@usm.my (S.Z.M.J.)
[2] School of Distance Education, Universiti Sains Malaysia, George Town 11800, Malaysia
[3] School of Pharmaceutical Sciences, Universiti Sains Malaysia, George Town 11800, Malaysia; habibahw@usm.my (H.A.W.); maisharah@usm.my (S.M.S.G.)
* Correspondence: asyrafman@usm.my; Tel.: +60-4-6533-935

Abstract: Creating optimal logic mining is strongly dependent on how the learning data are structured. Without optimal data structure, intelligence systems integrated into logic mining, such as an artificial neural network, tend to converge to suboptimal solution. This paper proposed a novel logic mining that integrates supervised learning via association analysis to identify the most optimal arrangement with respect to the given logical rule. By utilizing Hopfield neural network as an associative memory to store information of the logical rule, the optimal logical rule from the correlation analysis will be learned and the corresponding optimal induced logical rule can be obtained. In other words, the optimal logical rule increases the chances for the logic mining to locate the optimal induced logic that generalize the datasets. The proposed work is extensively tested on a variety of benchmark datasets with various performance metrics. Based on the experimental results, the proposed supervised logic mining demonstrated superiority and the least competitiveness compared to the existing method.

Keywords: supervised learning; Hopfield neural network; logic mining; artificial neural network

MSC: 68T07

Citation: Kasihmuddin, M.S.M.; Jamaludin, S.Z.M.; Mansor, M.A.; Wahab, H.A.; Ghadzi, S.M.S. Supervised Learning Perspective in Logic Mining. *Mathematics* **2022**, *10*, 915. https://doi.org/10.3390/math10060915

Academic Editor: Liangxiao Jiang

Received: 15 January 2022
Accepted: 30 January 2022
Published: 13 March 2022

1. Introduction

In the area of artificial intelligence (AI), two important perspectives stand out. The first is the applied rule that represents the given problem. The applied rule is vital in decision making in order to explain the nature of the problem. The second perspective is the automation process based on the rule which leads to neuro symbolic integration. These two perspectives rely heavily on the practicality of the symbolic rule that governs the AI system. The use of a satisfiability (SAT) perspective in software and hardware system theories is currently one of the most effective methods in bridging the two perspectives. SAT offers the promise, and often even the reality, that the model checks efforts with feasible industrial application. There were several practical applications of SAT that can be mentioned in this section. Ref. [1] utilized Boolean SAT by integrating satisfiability modulo theories (SMT) in tackling the scheduling problem. The proposed SMT method was reported to outperform other existing methods. Ref. [2] discovered vesicle traffic network by model checking that incorporates Boolean SAT. The proposed SAT model established a connection between vesicle transport graph connectedness and underlying rules of SNARE protein. In another development, [3] developed several SAT formulations to deal with the resource-constrained project scheduling problem (RCPSP). The proposed method is reported to solve various benchmark instances and outperform the existing work in terms of computation time and optimality. SAT formulation is a dynamic language that can be used in representing problem in hand. Ref. [4] proposed a special SAT in modelling the

circuit. The proposed method reconstructed the accurate circuit configuration up to 90%. The application of SAT in very-large-scale integration (VLSI) inspires the authors to extend the application of SAT into pattern reconstruction [5] where they used the variable in SAT as a building block of the desired pattern. The practicality of SAT motivates researchers to implement SAT in navigating the structure in an artificial neural network (ANN).

Logic programming in ANN has been initially proposed by [6]. In his work, logic programming can be embedded into the Hopfield neural network (HNN) by minimizing the logical inconsistencies. This is also a pioneer to the Wan Abdullah method which obtains the synaptic weight by comparing cost function with Lyapunov energy function. Ref. [7] further developed the idea of the logic programming in HNN by implementing Horn satisfiability (HornSAT) as a logical structure of HNN. The proposed network achieved more than 80% global minima ratio but high computation time due to the complexity of the learning phase. Since then, logic programming in ANN was extended to another type of ANN. Ref. [8] initially proposed logic programming in radial basis function neural network (RBFNN) by calculating the centre and width of the hidden neurons that corresponds to the logical rule. In the proposed method, the dimensionality of the logical rule from input to output can be reduced by implementing Gaussian activation function. The further development of logic programming in RBFNN were proposed in [9] where the centre and the width of the RBFNN are systematically calculated. In another development, [10] proposed a systematic logical rule by implementing a 2-satisfiability logical rule (2SAT) in HNN. The proposed hybrid network is incorporated with effective learning methods, such as genetic algorithm [11] and artificial bee colony [12]. The proposed network managed to achieve more than 95% of global minima ratio and can sustain a high number of neurons. In another development, [13] proposed the higher order non-systematic logical rule, namely random k satisfiability (RANkSAT) that consists of random first-, second-, and third-order logical rule. The proposed works run a critical comparison among a combination of RANkSAT and demonstrate the capability of non-systematic logical rule to achieve optimal final state. The practicality of the SAT in HNN was explored in pattern satisfiability [5] and circuit satisfiability [4] where the user can capture the visual interpretation of logic programming in HNN. However, up to this point, the choice of SAT structure in HNN has received very little research attention, despite its practical importance.

Current data mining were reported to achieve good accuracy but the interpretability of the output is poorly understood due to emphasize of the black box model. In other words, the output makes sense for the AI but not for the user. One of the most useful applications of logic programming in HNN is logic mining. Logic mining is a relatively new perspective in extracting the behaviour of the dataset via logical rule. This method is a pioneer work of [14]. In this work, the proposed RA extracted individual logical rule that represents the performance of the students. The logical rule extracted from the datasets is based on the number of induced Horn logics produced by HNN. Thus, there is very limited effort to identify the "best" induced logical rule that represent the datasets. To complement the limitation of the previous RA, several studies include specific SAT logical rules to be embedded into HNN. Ref. [15] introduced 3-satisfiability (3SAT) as a logical rule in HNN, thus creating the first systematic logic mining technique, i.e., the k satisfiability reverse analysis method (kSATRA). The proposed hybrid logic mining is used to extract logical rule in several fields of studies, such as social media analysis [15] and cardiovascular disease [16]. In another development, different types of logical rule (2SAT) have been implemented by [17]. They proposed 2SATRA by incorporating the 2SAT logical rule in extracting a diabetes dataset [17] and student's performance dataset [18]. Ref. [19] utilized 2SATRA by extracting logical rule for football datasets in several established football league in the world. Pursuing that, the ability of 2SATRA is further tested when the proposed method is implemented in e-games. The 2SATRA has been proposed to extract the logical rule that explains the simulation game of the League of Legend (LOL) [20]. The proposed method achieved an acceptable range of logical accuracy. The application of logic mining was extended to several prominent areas, such as extracting the price information from

commodities [21]. Another interesting development for *k*SATRA is by incorporating energy in induced logic. Ref. [22] proposed an energy-based 2-satisfiability-based reverse analysis method (E2SATRA) for e-recruitment. The proposed method reduced the suboptimal induced logic and increased the classification accuracy of the network. Despite the increase in application in data mining, the existing logic mining endured a significant drawback. The induced logic produced by the proposed method suffers from a limited amount of search space. This is due to the positioning of the neurons in *k*SAT formulation which affects the classification ability of 2SATRA. In this case, the optimal choice of the neuron pair in the *k*SAT clause in logic mining is crucial to avoid possible overfitting.

There were various studies that implemented regression analysis in ANN. Standalone regression analysis was prone to data overfitting [23], easily affected by outlier [24], and mostly limited to a linear relationship [25]. Due to the above weaknesses, regression analysis was implemented to complement the intelligent system. In most studies, regression analysis will be utilized in the pre-processing layer before it can be processed by the ANN. Ref. [26] proposed a combination of regression analysis with a RBFNN. The proposed method formed a prediction model for national economic data. Ref. [27] proposed an ANN that combines with regression analysis via a mean impact value. The proposed hybrid network identifies and extracts input variables that deal with irregularity and vitality of Beijing International Airport's passenger flow dataset. In [28], ANN is used to predict the water turbidity level by using optical tomography. The proposed ANN utilized the regression analysis value as an objective function of the network. Ref. [29] fully utilized logistic regression to identify significant microseismic parameters. The significant parameters will be trained by a simple neural network which results in the highly accurate seismic model. By nature, ANN is purely unsupervised learning and logistic regression analysis displays a major improvement to the overall performance. Although there were many studies conducted to confirm the benefit logistic regression analysis in classification and prediction paradigm, regression analysis has never been implemented in classifying the SAT logical rule. Regression analysis has the ability to restructure the logical rule based on the strength of relationship for each k variables in the *k*SAT clause. In that regard, the ANN will learn the correct logical structure and the probability to achieve highly accurate induced logical rule will increase dramatically. In that regard, relatively few studies have examined the effectiveness of regression in analysing data features that correspond to the *k*SAT. The choice of variable pair in the 2SAT clause can be made optimally by implementing regression analysis without interrupting the value of the cost function.

Unfortunately, there is no recent effort to discover the optimal choice that leads to the true outcome of the *k*SAT. The closest work that addresses this issue is shown by [30]. This work [30] utilized neuron permutation to obtain the most accurate induced logical rule by considering $n(n-1)!$ neuron arrangement in *k*SAT. Hence, the aim of this paper is to effectively explore the various possible logical structures in 2SATRA. The proposed logic mining model identifies the optimal neuron pair for 2SAT clause forming a new logical formula. Pearson chi-square association analysis will be conducted to examine the connectedness of the neuron with respect to the outcome. By doing so, the new 2SAT formula learned by HNN as an input logic and the new induced logical rule can be obtained. Thus, the contributions of this paper are:

(a) To formulate a novel supervised learning that capitalize correlation filter among variables in the logical rule with respect to the logical outcome;
(b) To implement the obtained supervised logical rule into HNN by minimizing the cost function which minimizes the final energy;
(c) To develop a novel logic mining based on the hybrid HNN integrated with the 2-satisfiability logical rule;
(d) To construct the extensive analysis for the proposed logic mining in doing various datasets. The proposed logic mining will be compared to the existing state of the art logic mining.

An effective 2SATRA model incorporating a new supervised model will be compared with the existing 2SATRA model for several established datasets. In Section 2, we describe satisfiability programming in HNN in detail. In Section 3, we describe some simulation of HNN by using simulated result. Discussion follows in Section 4. The concluding remarks in Section 5 complete the paper.

2. Motivation

2.1. Optimal Attribute Selection Strategy

Optimal attribute selection is vital to ensure HNN learn the correct logical rule during the learning phase. Ref. [30] proposed logic mining that capitalize random attribute combination that leads to creation of 2SAT logic. In this study, the synaptic weight connection obtained from 2SAT is purely based on the most frequent logical incidence in the datasets. The main question to ask: what happen if the 2SAT logical rule selected the wrong attribute? Hence, there is a huge possibility of the logic mining to learn the wrong synaptic which leads to suboptimal induced logic. A similar observation was made in the study by [31] which proposed 3SAT for induced logic, with a heavy focus on the random attribute selection. It is agreeable that the induced logic might produce accurate induced logic, but this issue leads logic mining to choose the random attributes that reduce the interpretability of induced logic. To solve this issue, the latest study by [30] proposed permutation operator to optimize the random selection proposed by [20]. The permutation operator will increase the accuracy of the induced logic when we change the attribute in the logical formula. Despite the increase in the accuracy and other metrics, the interpretability issue remains unsolvable. This is due to the random selection that contributes to a lack of interpretability of the learned logic in HNN. In this paper, we capitalize the work of [20,30] by constructing the dataset in the form of 2SAT logical rule and permutation operator. By selecting the optimal attribute combination of 2SAT, we can obtain more search space which leads to optimal induced logic.

2.2. Energy Optimization Strategy

Energy optimization in HNN is vital to ensure that every induced logic produced during retrieval phase is always achieved by global minimum energy. This creates an important question is: why HNN must achieve global minimum energy? Global minimum energy indicates a good agreement between the learned logic during pre-processing stage with the induced logic during retrieval phase. Induced logic that achieved global minimum energy can be interpreted. In contrast, induced logic that can achieve local minimum energy might achieve good accuracy, but this is difficult to interpret. In [22], the proposed logic mining is mainly the focus on the energy stability. The main issue when the induced logic is solely focusing on global minimum energy is limit on the possible search space of the HNN. The proposed HNN tends to overfit and produce more redundant induced logic. This will worsen when the proposed HNN selects the wrong attribute to learn. Non-optimal induced logic obtained a lack of interpretability and generalization during the retrieval phase. We tend to achieve similar induced logic which will lead to lower accuracy. Another factor that might affect overfitting of the induced logic structure is the monotonous behaviour of HNN that always converges to the nearest minimum energy. Hence, the feature of energy optimization with the optimal attributes selection will lead to a result that is optimal and easy to interpret.

2.3. Lack of Effective Metric to Assess the Performance of Logic Mining

Effective metric in logic mining is crucial to ensure the actual performance of the induced logic in doing clustering and classification. According to the previous studies, the point of assessment and type of metric are still shallow and do not represent the performance of the logic mining. For instance, the work of [21] reported the error analysis learning phase of HNN but a failure to provide metrics that are related to the contingency table. As a result, the actual performance of the induced logic is still not well understood.

Similar limitation reported in [14] where only metric of global minima ratio is used to demonstrate the connection between neurons. The local minimum solution signifies the induced logic rule does not correspond to the learned logic which contribute to the lack of generalization capability. In this case, if the measurement is solely based on the energy metric, then quantifying each element, in terms of confusion metric, is necessary so that the induced logic can carry out the classification task. In addition, the building block that leads to intermediate logics is solely based on the obtained synaptic weight. In this context, without synaptic weight analysis, the connection of the induced logic is poorly understood. For instance, logic mining [20] does not report the result of the strength of connection between variables in the induced logic. As a result, there is no method to assess the logical pattern stored in the content addressable memory (CAM). In this paper, comprehensive analysis, such as error analysis, synaptic weight analysis, and statistical analysis will be employed to get an overall view on the actual performance of all the logic mining models.

3. Satisfiability Representation

SAT is a representation of determining the interpretation that satisfies the given Boolean formula. According to [32], SAT is proven to be an NP-complete problem and is included to cover wide range of optimization problem. Extensive research on SAT leads to the creation of variant SAT which is 2SAT. In this paper, the choice of $k = 2$ is due to the two-dimensional decision-making system. Generally, 2SAT consist of the following properties [19]:

(a) A set of defined x variables, $q_1, q_2, q_3, \ldots, q_x$ where $q_i \in \{-1, 1\}$ that exemplify false and true, respectively.
(b) A set of literals. A literal can be variable or the negation of variable such that $q_i \in \{q_i, \neg q_i\}$.
(c) A set of x definite clauses, $C_1, C_2, C_3, \ldots, C_y$. Every consecutive C_i is connected to logical AND $(\wedge)$. Each two literals in (b) are connected by logical OR $(\vee)$.

By taking property (a) into account until (c), one can define the explicit definition of Q_{2SAT} as follows:

$$Q_{2SAT} = \bigwedge_{i=1}^{y} C_i \tag{1}$$

where C_i is a list of clause with two variables each

$$C_i = \bigvee_{i=1}^{x} (m_i, n_i) \tag{2}$$

By considering the Equations (1) and (2), a simple example of Q_{2SAT} can be written as

$$Q_{2SAT} = (A \vee \neg B) \wedge (\neg M \vee D) \wedge (\neg E \vee \neg F) \tag{3}$$

where the clauses in Equation (3) are $C_1 = (A \vee \neg B)$, $C_2 = (\neg M \vee D)$, and $C_3 = (\neg E \vee \neg F)$. Note that each clauses mentioned above must be satisfied with specific interpretations [10]. For example, if the interpretation reads $(M, D) = (1, -1)$, Q_{2SAT} will evaluate false or -1. Since Q_{2SAT} contains an information storage mechanism and is easy to classify, we implemented Q_{2SAT} into ANN as a logical system.

4. Satisfiability in Discrete Hopfield Neural Network

HNN [33] consists of interconnected neurons without a hidden layer. Each neuron in HNN is defined in bipolar state $S_i \in \{1, -1\}$ that represents true and false, respectively. An interesting feature about HNN is the ability to restructure the neuron state until the network reached its minimum state. Hence, the proposed HNN achieved the optimal final

state if the collection of neurons in the network reached the lowest value of the minimum energy. The general definition of HNN with the i-th activation is given as follows

$$S_i = \begin{cases} 1 \ , & if \ \sum_{i=0}^{N} W_{ij} S_j \geq \theta \\ -1 \ , & otherwise \end{cases} \tag{4}$$

where θ and W_{ij} represent a threshold and synaptic weight of the network, respectively. Without compromising the generality of HNN, some study used $\theta = 0$ as the threshold value. Note that N is the number of 2SAT variables. W_{ij} is also defined as the connection between neuron S_i and S_j. The idea of implementing Q_{2SAT} in HNN (HNN-2SAT) is due to the need of some symbolic rule that can govern the output of the network. The cost function $E_{Q_{2SAT}}$ of the proposed Q_{2SAT} in HNN is given as follows:

$$E_{Q_{2SAT}} = \sum_{i=1}^{NC} \prod_{j=1}^{2} M_{ij} \tag{5}$$

where NC is the number of $E_{Q_{2SAT}}$ clause. The definition of the clause M_{ij} is given as follows [9]

$$M_{ij} = \begin{cases} \frac{1}{2}(1 - S_y), & if \ \neg y \\ \frac{1}{2}(1 + S_y), & otherwise \end{cases} \tag{6}$$

where y is the negation of literal in Q_{2SAT}. It is also worth mentioning that $E_{Q_{2SAT}} = 0$ if the $\frac{1}{4}(1 \pm S_y) = 0$ is because the neuron state S_y associated to Q_{2SAT} is fully satisfied. Each variable inside a particular M_{ij} will be connected by W_{ij}. Structurally, the synaptic weight of Q_{2SAT} is always symmetrical for both the second- and third-order logical rule:

$$W_{AB}^{(2)} = W_{BA}^{(2)} \tag{7}$$

with no self-connection between neurons:

$$W_{AA}^{(2)} = W_{BB}^{(2)} = 0 \tag{8}$$

Note that Equations (5)–(8) only account for a non-redundant logical rule because the logical redundancies will result in the diminishing effect of the synaptic weight. The goal of the learning in HNN is to minimize the logical inconsistency that leads to $Q_{2SAT} = -1$ or $\neg Q_{2SAT} = 1$. Although synaptic weight of the HNN can be properly trained by using conventional method, such as Hebbian learning [33], ref. [14] demonstrated that the Wan Abdullah method can obtain the optimal synaptic weight with minimal neuron oscillation compared to Hebbian learning. For example, if the embedded logical clause is $C_1 = (A \vee \neg B)$, the synaptic weights will read $(W_A, W_{B,} W_{AB}) = (0.25, 0.25, -0.25)$. During retrieval phase of HNN-2SAT, the neuron state will be updated asynchronously based on the following equation.

$$S_i = \begin{cases} 1 \ , & \sum_{j=1, i \neq j}^{N} W_{ij}^{(2)} S_j + W_i^{(1)} \geq \xi \\ -1 \ , & \sum_{j=1, i \neq j}^{N} W_{ij}^{(2)} S_j + W_i^{(1)} < \xi \end{cases} \tag{9}$$

where S_i is a final neuron state with pre-defined threshold ξ. In terms of output squashing, the Sigmoid function can be used to provide non-linearity effects during neuron classification. Potentially, the final state of the neuron must contain information that lead to

$E_{Q_{2SAT}} = 0$, and the quality of the obtained state can be computed by using Lyapunov energy function:

$$H_{Q_{2SAT}} = -\frac{1}{2}\sum_{i=0,i\neq j}^{N}\sum_{j=0,j\neq i}^{N} W_{ij}^{(2)} S_i S_j - \sum_{i=0}^{N} W_i^{(1)} S_i \tag{10}$$

According to [33], the symmetry of the synaptic weight is sufficient condition for the existence of the Lyapunov function. Hence, the value of $H_{Q_{2SAT}}$ in Equation (10) decreases monotonically with network. The absolute minimum energy $H_{Q_{2SAT}}^{\min}$ can pre-determined by substituting interpretation that leads to $E_{Q_{2SAT}} = 0$. In this case, if the obtained neuron state can satisfy $\left|H_{Q_{2SAT}} - H_{Q_{2SAT}}^{\min}\right| \leq Tol$, the final neuron state achieved global minimum energy. Note that the current conventions of $S_i \in \{1, -1\}$ can be converted to binary by implementing different a Lyapunov function coined by [6].

5. Proposed Method

2SATRA is a logic mining method that can extract a logical rule from the dataset. The philosophy of the 2SATRA is to find the most optimal logical rule of Equation (1), which corresponds to the dynamic system of Equation (9). In the conventional 2SATRA proposed by [20], the choice of variable in 2SATRA will be determined randomly which leads to poor quality of the induced logic. The choices of the neurons are arranged randomly before the learning of HNN can take place. In this section, chi-square analysis will be used during the pre-processing stage. The aim of the association method is to assign the two best neurons/clauses that correspond to the outcome Q_{2SAT}. These neurons will take part during the learning phase of HNN-2SAT which leads to better induced logic. In other words, the additional optimization layer is added to reduce the pre-training effort for 2SATRA to find the best logical rule.

Let N the number of neurons represent the attribute of the datasets $S_i = (S_1, S_2, S_3, \ldots, S_N)$ where each neuron is converted into bipolar interpretation $S_i = \{-1, 1\}$. Necessarily, 2SATRA is required to select d neurons that will be learned by HNN-2SAT. In this case, the number of possible neuron permutation after considering the learning logic Q_i^l structure is $\frac{N!}{2(N-d)!}$. By considering the relationship between Q_i^l and neuron S_i, we can optimally select the pair of S_i for each clause C_i. The S_i selection for each C_i is given as follows:

$$Q_i^l = \bigwedge_{i=0,i\neq j}^{NC}\left(S_i^{\min|P_i|} \vee S_i^{\min|P_j|}\right), \quad i \neq j, \quad 0 \leq P_i \leq \alpha, \quad 0 \leq P_j \leq \alpha \tag{11}$$

where P_i is the P value between Q_i^l and the neuron S_i. $\min|P_i|$ signifies the minimized value of P_i recorded between Q_i^l and S_i, and the value of α is pre-defined by the network. Note that $i \neq j$ does not significy a self-connection between the same neurons. By considering the best- and worst-case scenario, the neuron will be chosen at random if $\min|P_i| = \min\left|P_j\right|$. If the examined neurons do not achieve the pre-determined association, HNN-2SAT will reset the search space, which fulfils the threshold association value. Hence, by using Equation (11), the proposed 2SATRA is able to learn the early feature of the dataset. After obtaining the right set of neurons for Q_i^l, the dataset will be converted into bipolar representation:

$$S_i = \begin{cases} 1\ , & S_i = 1 \\ -1\ , & otherwise \end{cases} \tag{12}$$

Note that we only consider the second-order clause or $C_i^{(2)}$ for each clause in Q_i^l. Hence, the collection of S_i that leads to positive outcome of the learning data or $Q_i^l = 1$ will be segregated. By calculating the collection of $C_i^{(2)}$ that leads to $Q_i^l = 1$, the optimum logic Q_{best} is given as follows:

$$Q_{best} = \max\left[n\left(C_i^{(2)}\right)\right], \quad Q_i^l = 1 \tag{13}$$

where $n\left(C_i^{(2)}\right)$ is the number of Q_i^l that leads to $Q_i^l = 1$. Hence, the logical feature of the Q_{best} can be learned by obtaining the synaptic weight of the HNN. In this case, the cost function in Equation (11) which corresponds to Q_{best} will be compared to Equation (5). By using Equation (9), we obtain the final neuron state S_i^B.

$$S_i^{induced} = \begin{cases} S_i & , \quad S_i^B = 1 \\ \neg S_i & , \quad S_i^B = -1 \end{cases} \tag{14}$$

Since the proposed HNN-2SAT only allows an optimal final neuron state, the quality of the S_i^B will be verified by using $\left|H_{Q_{2SAT}} - H_{Q_{2SAT}}^{\min}\right| \leq Tol$. In this case, S_i^B that leads to local minima will not be considered. Hence, the classification of the induced Q_i^B is as follows:

$$Q_i^B = \begin{cases} Q_i^B & , \quad \left|H_{Q_i^B} - H_{Q_i^B}^{\min}\right| \leq \partial \\ 0 & , \quad otherwise \end{cases} \tag{15}$$

where $H_{Q_i^B}^{\min}$ can be obtained from Equation (10). It is worth mentioning that if the two neurons do not have the strong association, the neurons will not be considered. Thus, if the association value for all neurons does not achieve the threshold variable $0 \leq \rho_i \leq \alpha$, the proposed network will be reduced to conventional *k*SATRA proposed by [21,31]. Figure 1 shows the implementation of the proposed supervised logic mining or (S2SATRA). Algorithm 1 shows Pseudo code of the Proposed S2SATRA.

Algorithm 1. Pseudo code of the Proposed S2SATRA.

Input: Set all attributes $A_1,\ A_2, A_3, \ldots, A_N$ with respect to Q_{learn}.
Output: The best induced logic Q_i^B.

```
Begin
Initialize algorithm parameters;
Define the Attribute for A_1, A_2, A_3, ..., A_N with respect to Q_i^l;
Find the correlation value between A_i with Q_i^l;
for (Q_i^l ≤ Q_{N_data}^l) do
    if Equation (11) is satisfied then
        Assign A_i as S_i, and continue;
    while (i ≤ Per) do
        Using the found attributes, find Q_best using Equation (13);
        Check the clause satisfaction for Q_best;
        Compute H_{Q_best}^min using Equation (10);
        Compute the synaptic weight associated with Q_best using the WA method;
        Initialize the neuron state;
        for (g ≤ trial)
            Compute h_i using Equation (9);
            Convert S_i^B to the logical form using Equation (14);
            Evaluate the H_{Q_i^B} by using Equation (10);
            If Condition (15) is satisfied then
                Convert to induced logic Q_i^B;
                Compare the outcome of the Q_i^B with Q_test and continue;
            g ← g + 1;
        end for
        i ← i + 1;
    end while
end for
End
```

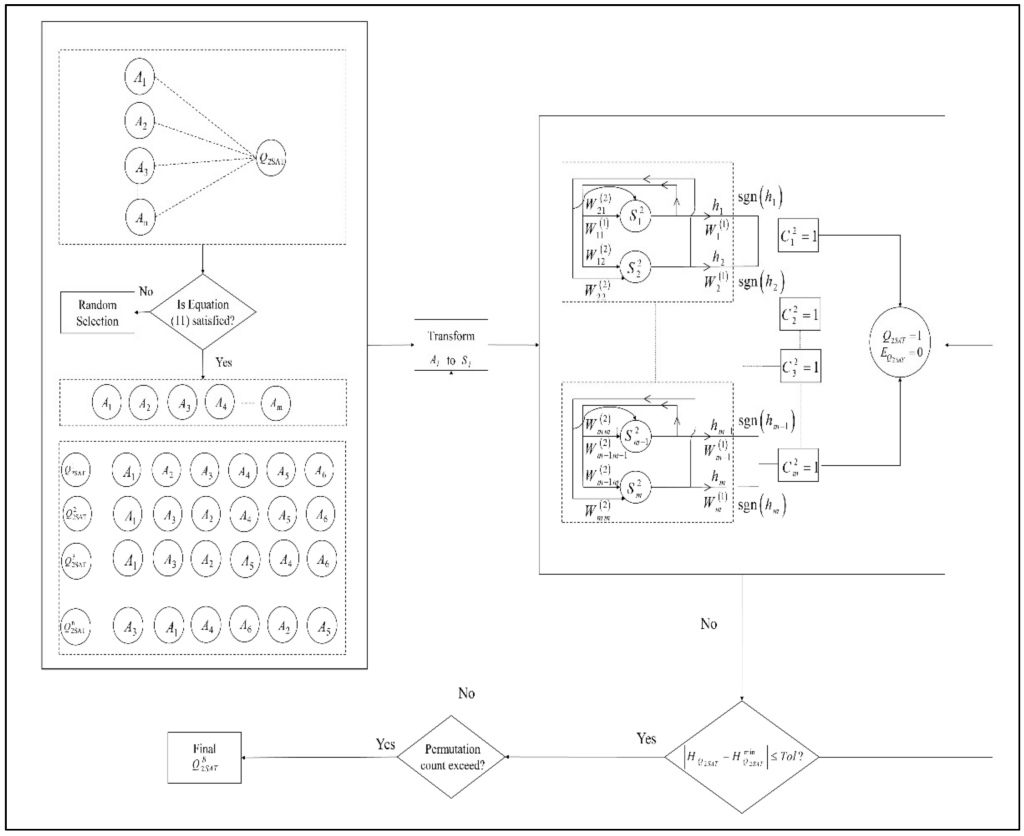


Figure 1. The implementation of the proposed S2SATRA.

6. Experiment and Discussion

6.1. Experiment Setup

In this section, we describe the components of the experiments carried out here. The purpose of this experiment is to elucidate the different logic mining mechanism that leads to Q_{best} before it can be learned by HNN. To guarantee the reproducibility of the experiment, we set up our experiment as follows.

6.1.1. Benchmark Datasets

In this experiment, 12 publicly available datasets are obtained from UCI repository https://archive.ics.uci.edu/mL/datasets.php (accessed on 10 December 2021). These datasets are widely used in the classification field and are representative of practical classification problem. The details of the datasets are summarized in Table 1.

Table 1. List of datasets.

ID	Data	Instances	Attributes	Area	Outcome $Q_{2SAT}^{k_i}$
F1	Pageblocks	5473	10	Computer	Class
F2	Australian	690	14	Financial	Class
F3	Zoo	101	17	Life	Class
F4	Wisconsin	569	32	Life	Class
F5	Speaker	329	12	Social	Language
F6	Shuttle	58,000	9	Physical	Class
F7	Facebook	500	19	Business	Status
F8	Wine	178	13	Physical	Class
F9	Computer	209	9	Computer	ERP
F10	Energy Y1	768	8	Computer	Heating Load
F11	Ionosphere	351	34	Physical	Class
F12	Energy Y2	768	8	Computer	Cooling Load

To avoid possible field bias, the area of interest in the dataset varies from science to social datasets. The choice of datasets is based on two aspects. First, we only select a dataset that contains more than 100 instances to preserve the statistical property of a distribution. For example, we avoid choosing balloon datasets because the number of instances is statistically too small to assess the capability learning phase of the proposed model. Second, we only select a dataset that contains more than six attributes. The choice of having more than six attributes is to check the effectiveness of the proposed model in adapting the concept of an optimal attribute selection. In other words, this experiment is unable to assess the effectiveness of the proposed model using association analysis and permutation if the number of attributes is low. Note that the state of the data will be stored in neuron by using bipolar representation $S_i \in \{-1, 1\}$ and each state can represent the behaviour of the dataset with respect to Q_{best}. In terms of data normalization, *k*-mean clustering [34] will be used to normalize the continuous datasets into 1 and -1. For a dataset that contains categorical data, the proposed model and the existing model will randomly select $Q_{2SAT}^{k_i}$. Since the number of missing values for all datasets is very small and negligible, we replaced the missing value with a random neuron state. The experiment employs a train-split method where 60% of the dataset will be trained and 40% of the dataset will be tested [31]. Note that multi-fold validation was not implemented in this paper because we wanted to ensure that Q_{best} learned by HNN has a similar starting point for all logic mining models. A multi-fold validation method will eliminate the original point of assessment during the training phase of logic mining. Hence, the comparison among logic mining is not possible.

6.1.2. Performance Metrics

In terms of metric evaluation performance, several performance metrics were selected to measure the robustness of the proposed method compared to the other existing work. We divided performance metrics into a few parts. Error evaluations consist of a standard error metric, such as a root mean square error (*RMSE*) and a mean absolute error (*MAE*). The formulation for both errors are as follows:

$$RMSE = \sum_{i=1}^{n} \frac{1}{n} \left(Q_i^{test} - Q_i^B \right)^2 \tag{16}$$

$$MAE = \sum_{i=1}^{n} \frac{1}{n} \left| Q_i^{test} - Q_i^B \right| \tag{17}$$

where Q_i^{test} is the state of the data $Q_i^{test} \in \{-1, 1\}$. In detail, the best logic mining model will produce the Q_i^B with the lowest error evaluation. Next, standard classification metrics, such as accuracy, *F-score*, precision, and sensitivity will be utilized in the experiment. According to [35], the sensitivity metric *Se* analyses how well a case correctly produces a positive result for an instance that has a specific condition. Note that, TP (true positive) is the number of positive instances that correctly classified, FN (false negative) is the number of positive instances that incorrectly classified, TN (true negative) is the number of negative instances that correctly classified, and FP (false positive) is the number of incorrectly classified positive instances.

$$Se = \frac{TP}{TP + FN} \tag{18}$$

Meanwhile, precision is utilized to measure the algorithm's predictive ability. Precision refers to how precise the prediction is from those positively predicted with how many of them are actually positive. The calculation for precision (*Pr*) is defined as follows:

$$Pr = \frac{TP}{TP + FP} \tag{19}$$

Accuracy (*Acc*) is generally the common metric for determining the performance of the classification. This metric measures the percentage of instances categorized correctly:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

As stated by [36], *F-score* is a significant necessity that reflects the highest probability of correct result, explicitly representing the ability of the algorithm. Additionally, F1-score is described as the harmonic mean of precision and sensitivity. Next, the Matthews correlation coefficient (*MCC*) will be used to examine the performance of the logic mining based on the eight major derived ratios from the combination of all components of a confusion matrix. *MCC* is regarded as a good metric that represents the global model quality and can be used for classes of a different size [37].

$$F\ Score = \frac{2TP}{2TP + FP + FN} \tag{21}$$

$$MCC = \frac{TP\ \ TN\ - FP\ \ FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{22}$$

It is worth mentioning that this is our first encounter to approach logic mining with various performance metrics. In [20,22], the only metric used is only accuracy and testing error.

6.1.3. Baseline Methods

Since the main focus of this paper is to examine the performance of the induced logic produced by S2SATRA, we limit our comparison to only method that produce induced logic. Despite the fact that we respect the capability of the existing model in classifying the dataset, we will not compare S2SATRA with the existing classification model, such as random forest, decision tree, etc., because these models do not produce any logical rule that classifies the dataset. For consistency purposes, all the experiments will employ the same type of logical rule, i.e., Q_{2SAT}. For comparison purposes, the proposed S2SATRA will be compared with all the existing logic mining models, such as 2SATRA [20], the energy-based 2-satisfiability reverse analysis method (E2SATRA) [22], the 2-satisfiability reverse analysis method with permutation element (P2SATRA) [30], and the state-of-the-art reverse analysis method (RA) [14]. This section will discuss the implementation of each logic mining models.

(a) The conventional 2SATRA model proposed by [20] utilizes Q_{2SAT} integrated with the Wan Abdullah method. The determination of Q_{best} follows the Equation (13) and the selected attributes are randomized. During the retrieval phase, HNN-2SAT will retrieve the optimal S_i^B that leads to optimal induced logic which then leads to the potential generalization of the datasets. There is no layer of verification around whether the final state S_i^B produced is the global minimum energy.

(b) In E2SATRA [22], Lyapunov energy function in Equation (10) will be used to verify the Q_i^B. The final state of the HNN will converge to the nearest minimum solution. In this case, Q_i^B that achieve local minimum energy will be filtered out during retrieval phase of HNN-2SAT. The dataset generalization of E2SATRA does not consider the optimal attribute selection.

(c) In P2SATRA [30], the permutation operator will be used to permutate the attribute in $C_i^{(2)}$. The permutation operator will explore the possibility of search space related to the chosen attributes. Note that redundant permutation will not be considered during the attribute selection. The retrieval property of the P2SATRA will have the same property as conventional 2SATRA.

(d) As for RA proposed by [14], we introduced RA that can only produce HornSAT property [7] while still maintaining the two attributes per $C_i^{(2)}$. To make the proposed RA

comparable with our proposed method, calibration is required. The main calibration from the previous RA is the number of Q_i^B produced by the datasets. Instead of assigning neuron for each instance, we assign each neuron with attributes. The neuron redundancy is also introduced to avoid the net-zero effect of the synaptic weight.

During the learning phase, learning optimization is implemented to ensure that the synaptic weight obtained is purely due to the HNN. Note that the effective synaptic weight management will change the final state of HNN, leading to different Q_i^B. Since the HNN has a recurrent learning property [33], the neuron will change states until $Q_i^{learn} = 1$ and until the learning threshold NH is reached. According to [14], if the learning of Q_{2SAT} exceeds the proposed NH, the HNN will use the current optimal synaptic weight for the retrieval phase. During the retrieval phase of HNN, the neuron state will be initially randomized to reduce the possible bias. Noise function is not added, such as in [22,31], because the main objective of this experiment is to investigate the type of attributes that retrieve the most optimal final Q_i^B. To obtain consistent results throughout all 2SATRA models, the only squashing function employed by the neurons in 2SATRA models is the hyperbolic activation function in [38]. By considering only one fixed learning rule, we can examine the effect of supervised learning towards the 2SATRA model. Tables 1–5 illustrate the list of parameters involved in the experiment.

Table 2. List of parameters in S2SATRA.

Parameter	Parameter Value
Neuron Combination	100
Number of Trial	100
Number of Learning (Ω)	100
P-Value (P)	0.05
Logical Rule	Q_{2SAT}
Tolerance Value (∂)	0.001
No_Neuron String	100
Maximum Permutation (Per)	100

Table 3. List of parameters in E2SATRA [22].

Parameter	Parameter Value
Neuron Combination	100
Attribute Selection	Random
Number of Learning (Ω)	100
Logical Rule	Q_{2SAT}
Tolerance Value (∂)	0.001
No_Neuron String	100
Selection_Rate	0.1
Neuron Combination	100

Table 4. List of parameters in 2SATRA [20].

Parameter	Parameter Value
Neuron Combination	100
Attribute Selection	Random
Number of Learning (Ω)	100
Logical Rule	Q_{2SAT}
No_Neuron String	100
Selection_Rate	0.1

Table 5. List of parameters in P2SATRA [30].

Parameter	Parameter Value
Neuron Combination	100
Attribute Selection	Random
Number of Learning (Ω)	100
Logical Rule	Q_{2SAT}
No_Neuron String	100
Selection_Rate	0.1
Maximum Permutation	100

6.1.4. Experimental Design

The simulations were all implemented using Dev C++ Version 5.11 (manufactured by Bloodshed Company from USA) for Windows 10 (Microsoft from USA) in 2 GB RAM with Intel Core I3 (Intel from USA) as a workstation. As for association analysis, Q_{best} will be obtained by using IBM SPSS Statistics Version 27 (manufactured by IBM from New York, NY, USA). All the experiments were implemented in the same device to avoid a possibly bad sector during the simulation. Each 2SATRA model will undergo 10 independent runs to reduce the impact of bias caused by the random initialization of a neuron state.

7. Results and Discussion

7.1. Synaptic Weight Analysis

Figure 1 demonstrates that the optimal 2SATRA model requires pre-processing structure for neurons before the Q_{best} can be learned by HNN. The currently available 2SATRA model specifically optimizes the logic extraction from the dataset without considering the optimal Q_{best}. Hence, the mechanism that optimizes the optimal neuron relationship before the learning can occur remains unclear. Identifying a specific pair of neurons for Q_{2SAT} will facilitate the logic mining to obtain the optimal induced logic.

Figures 2–13 demonstrate the synaptic weight for all logic mining models in extracting logical information for F1–F12. Note that $W_i^{(1)}$ and $W_{ij}^{(2)}$ represent the first- and second-order connection in the $C_i^{(2)}$ clause. In this section, we will check the optimality of the synaptic weight with respect to the obtained accuracy value. Several interesting points can be made from Figures 2–13.

(a) Despite different attribute selection for S2SATRA compared to the other logic mining model, the induced logic for S2SATRA shows more logical variation compared to other existing work. For instance, the synaptic weight for S2SATRA has a bias towards a positive literal for only four datasets while maintaining high accuracy.

(b) RA demonstrates logical rigidness because the synaptic weight must produce a final state with at least one positive literal. According to Figures 2–13, the induced logic tends to overfit with the datasets. The structure of the induced logic obtained in RA might exhibit some diversity compared to S2SATRA but remains suboptimal, leading to a lower accuracy value. Hence, great diversity with wrong attribute selection reduces the effectiveness of logic mining model.

(c) In terms of energy optimization strategy, the energy filter in S2SATRA is able to retrieve global induced logic that contains more negated neurons compared to E2SATRA. This shows that the choice of attribute will definitely influence the choice of synaptic weight learning. For example, E2SATRA managed to achieve 10 similar global induced logic as an optimal logic for F1, F2, F3, F4, F5, F6, F7, F9, F10, and F12 compared to S2SATRA which can only retrieve 4 similar induced logic for F3, F4, F8, and F9. Despite having similar global induced logic, S2SATRA can still obtain a high accuracy level.

(d) Another interesting insight is that permutation operators improve P2SATRA in learning optimal synaptic weight, but the improvement seems more obvious in S2SATRA. For instance, with the same synaptic weight for neuron A and D but a different

attribute representation, S2SATRA is able to achieve higher accuracy. A similar observation is made for other neurons from A to E. This implies the need of the optimal attribute selection before learning of HNN can take place.

7.2. Correlation Analysis for S2SATRA

Tables 6 and 7 demonstrate the correlation value between the attribute A_i for F1 until F12 with respect to $Q_{2SAT}^{k_i}$. For a clear illustration, H_0 signifies that there is no correlation between the attribute A_i with $Q_{2SAT}^{k_i}$. Hence, if the correlation exists between the attributes and the outcome, we will "reject" the decision of H_0 and the connotation of "Accept" means the otherwise [39]. In other words, the aim of this analysis is to verify which A_i will be chosen to represent the C_i in $Q_{2SAT}^{k_i}$. Based on Table 8, most of the attributes selected in S2SATRA have a high correlation with $Q_{2SAT}^{k_i}$. The non-correlated attributes will be disregarded in the right way before it can be introduced in the learning phase of HNN. The main concern in the conventional logic mining model is the possible choice of A_i that construct C_i purely based on the random selection. For example, in F12, the logic mining model without a supervised layer might choose A_6 and A_8 to construct C_i and will have to learn unnecessary attributes that lead to $Q_{2SAT}^{k_i} = 1$. In this context, HNN-2SAT will learn non-optimal $Q_{2SAT}^{k_i}$ that corresponds to the datasets which has no correlation with the final outcome. Hence, the effectiveness of knowledge extraction for logic mining will be reduced dramatically because one of the C_i is not correlated to the desired outcome. Based on the result, the correlation layer is vital to avoid S2SATRA from choosing the wrong attributes.

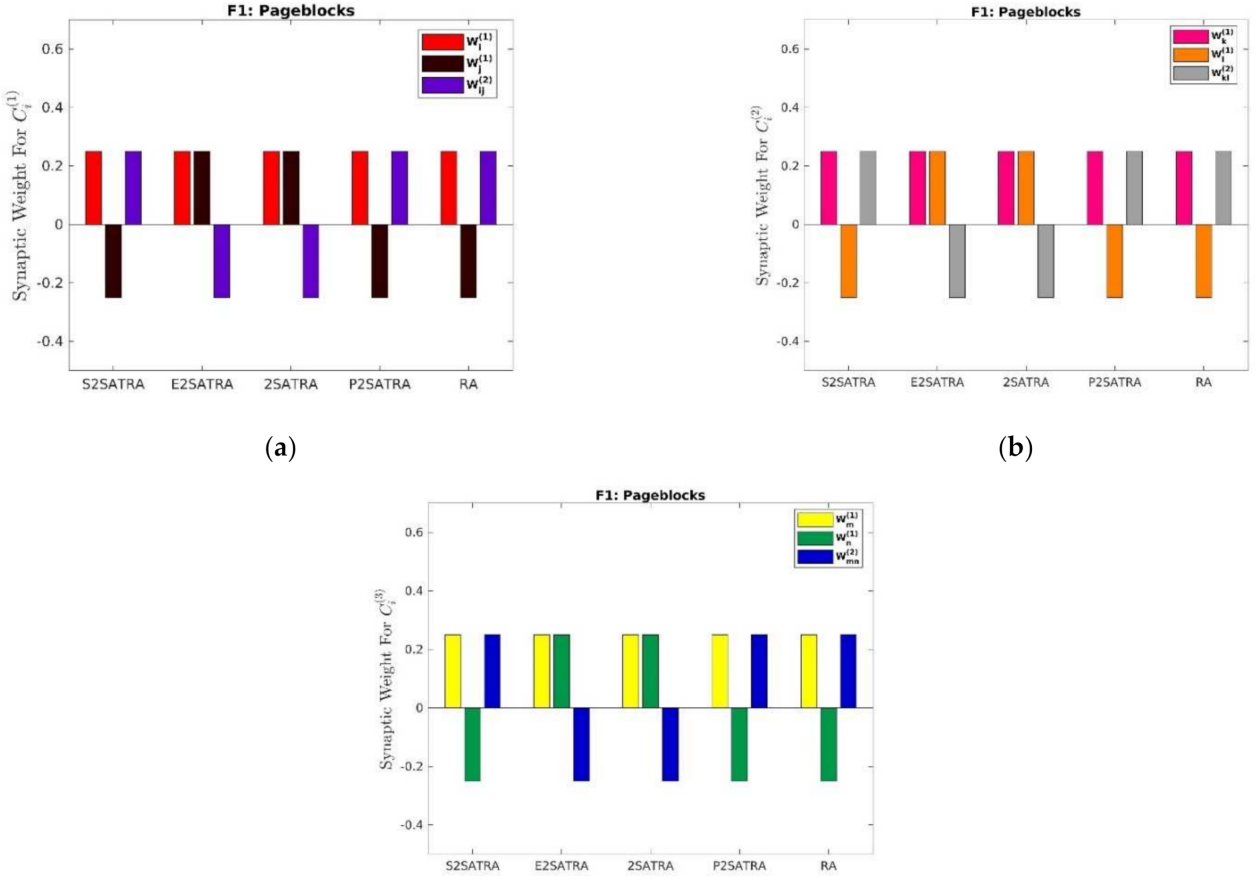


Figure 2. Synaptic weight analysis for F1: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.

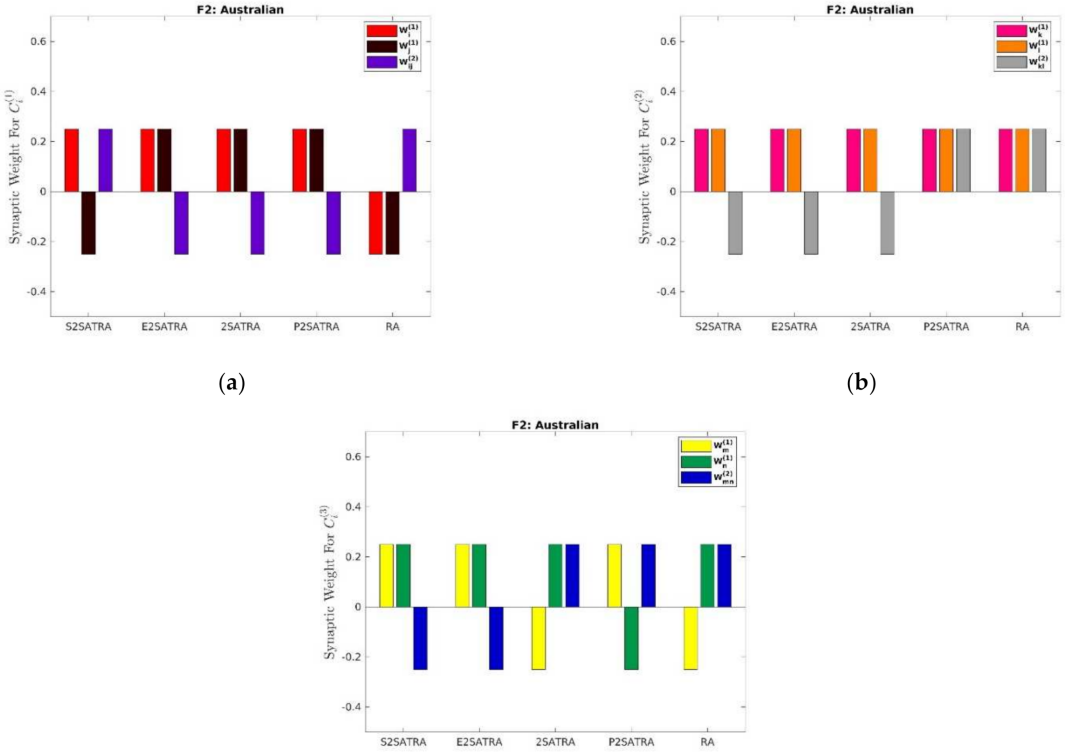


Figure 3. Synaptic weight analysis for F2: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.

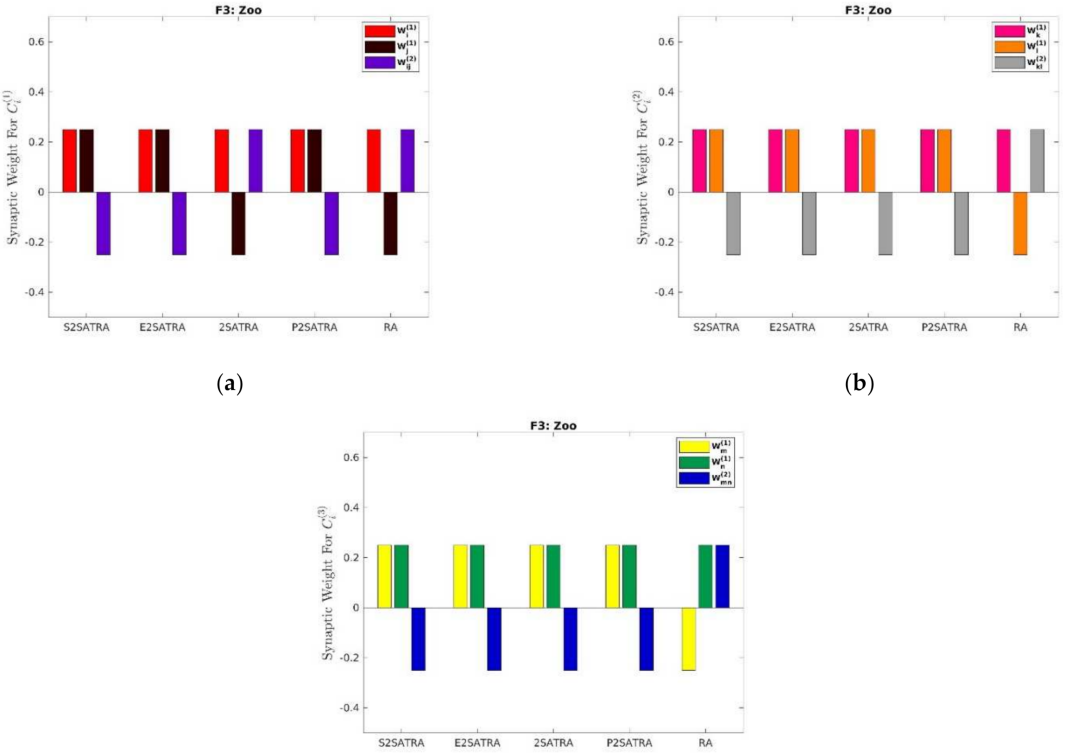


Figure 4. Synaptic weight analysis for F3: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.

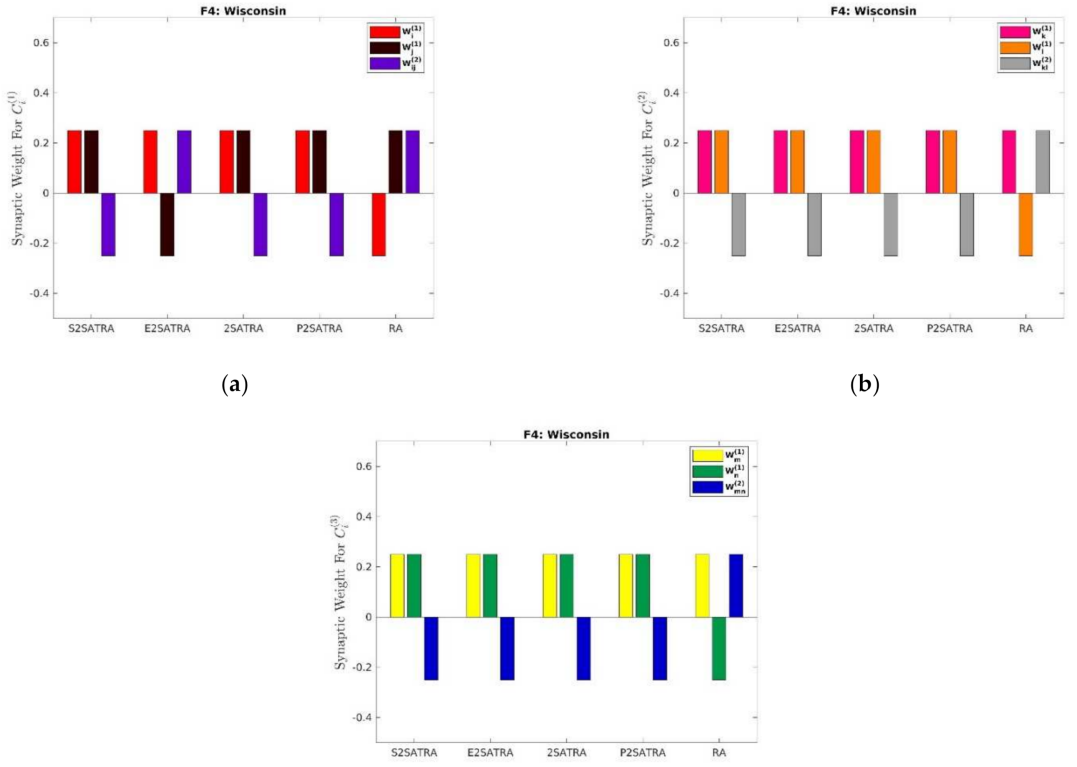


Figure 5. Synaptic weight analysis for F4: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.

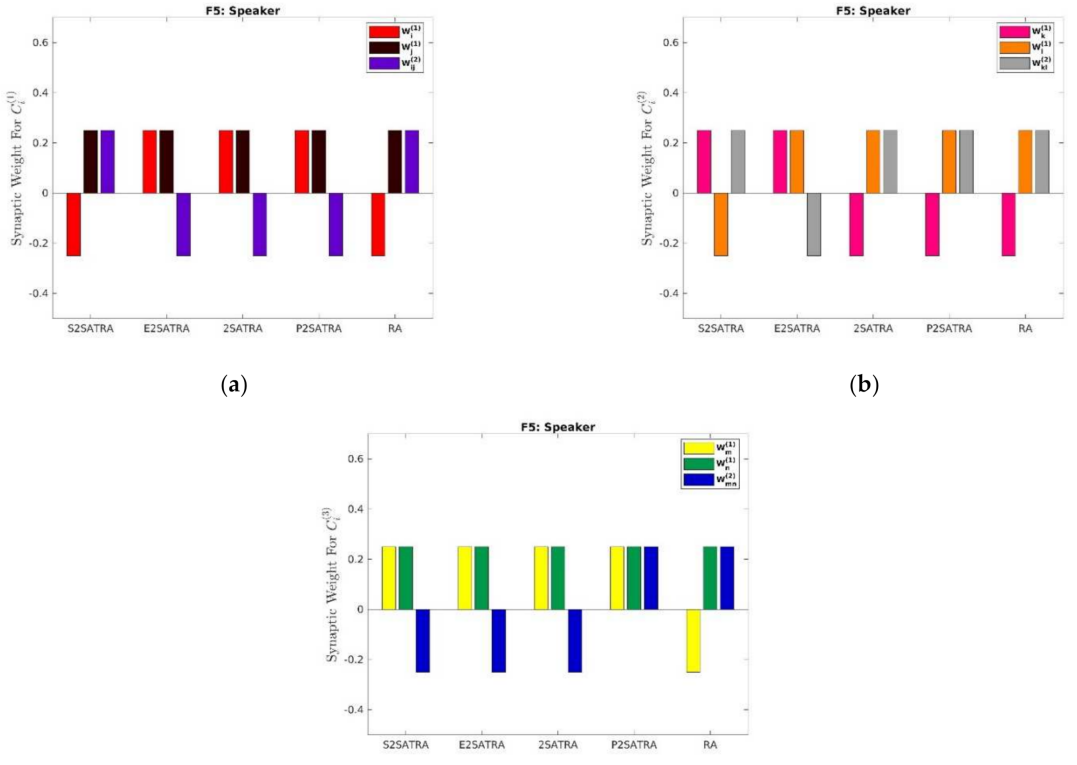


Figure 6. Synaptic weight analysis for F5: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.

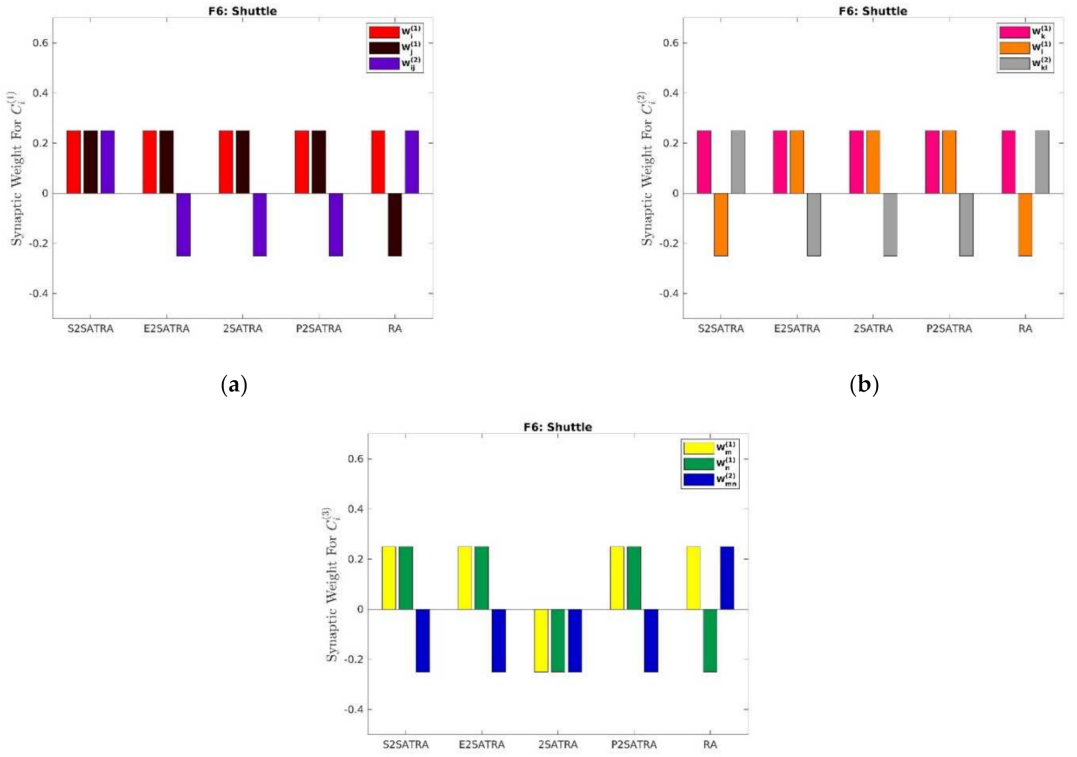


Figure 7. Synaptic weight analysis for F6: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.

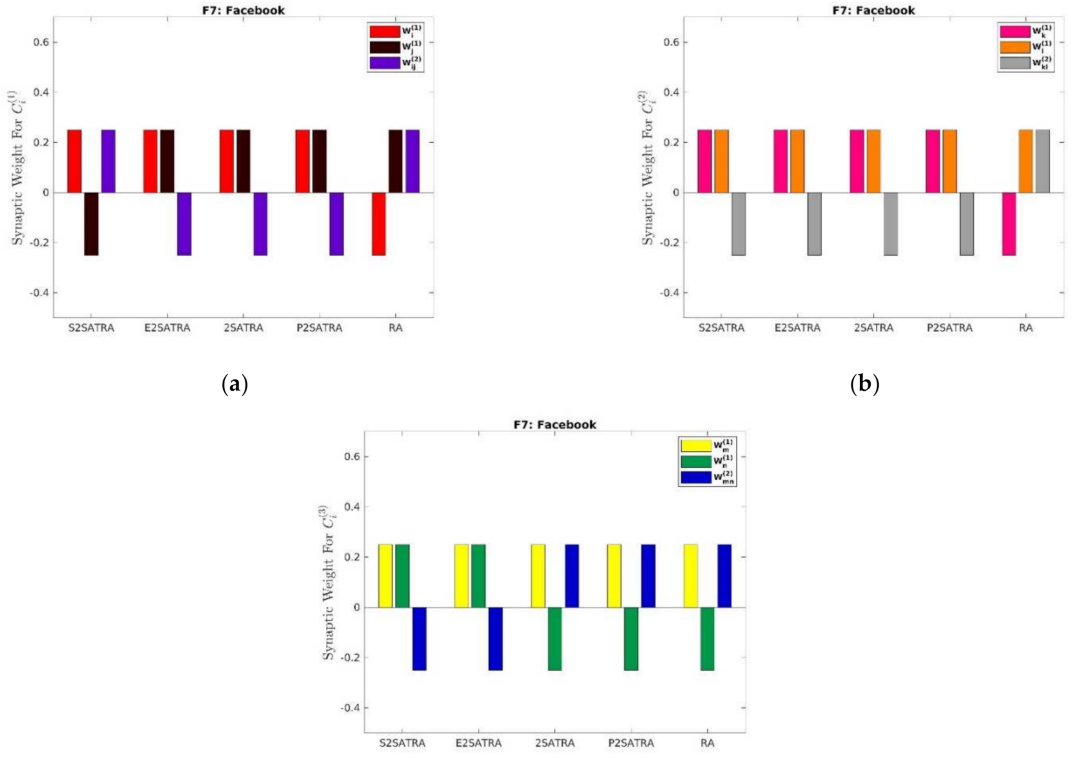
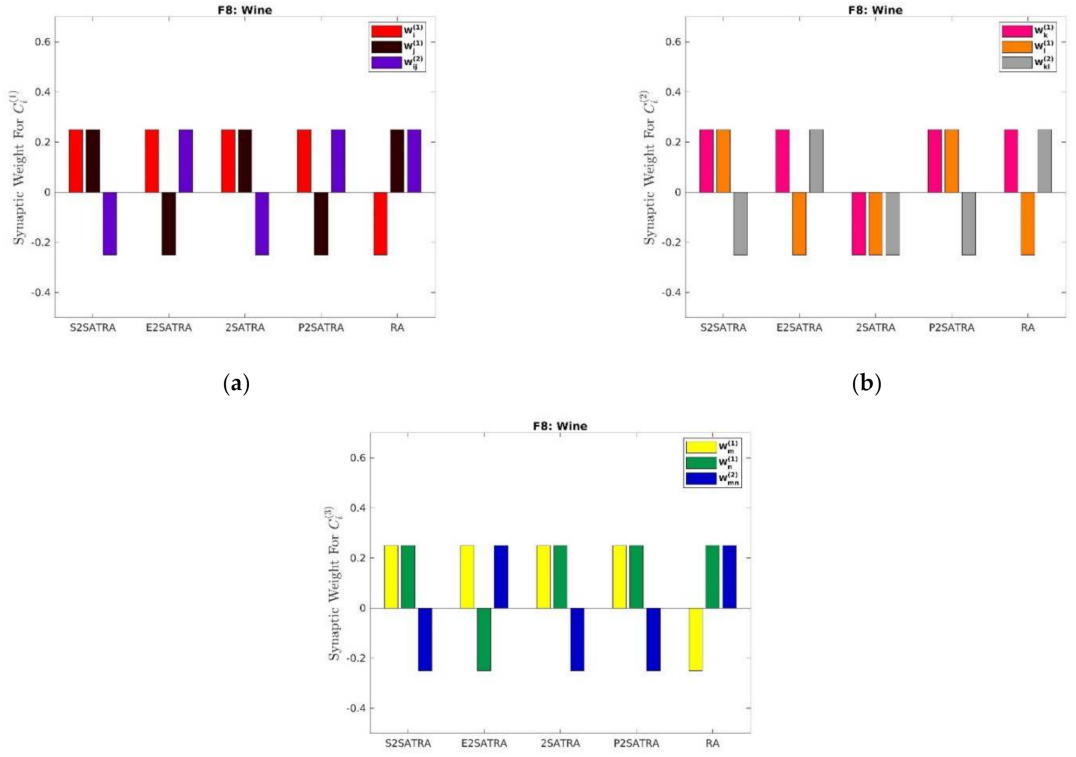


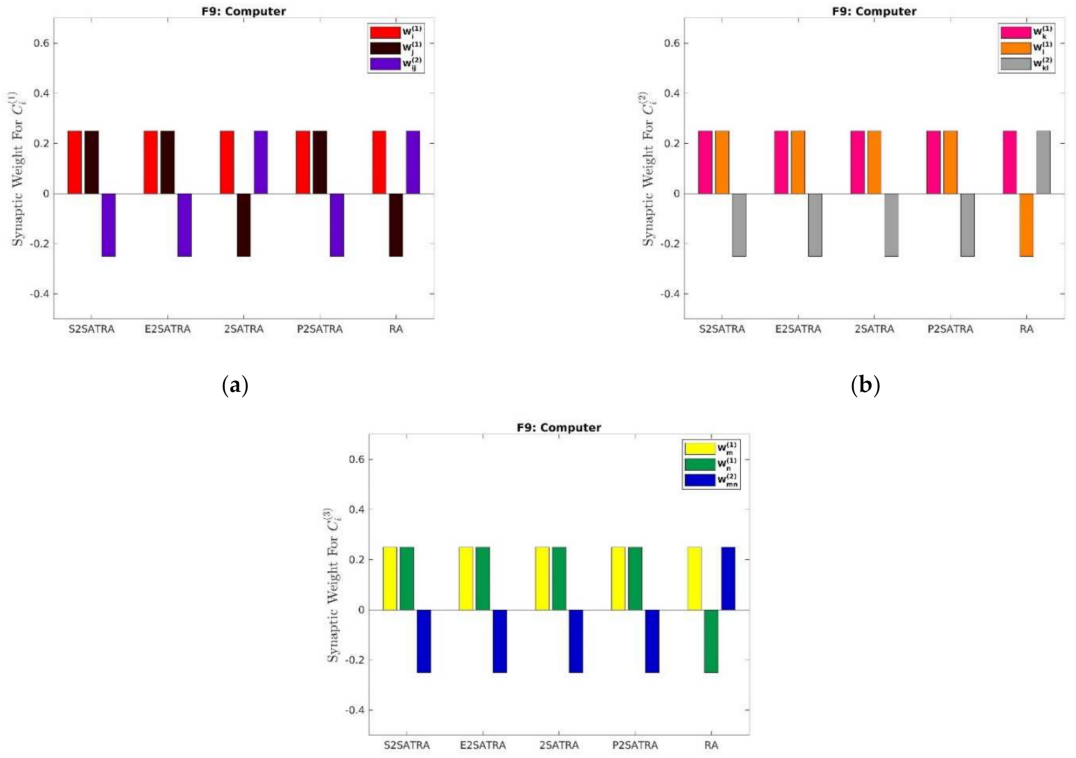
Figure 8. Synaptic weight analysis for F7: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.



(**a**) (**b**)

(**c**)

Figure 9. Synaptic weight analysis for F8: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.



(**a**) (**b**)

(**c**)

Figure 10. Synaptic weight analysis for F9: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.

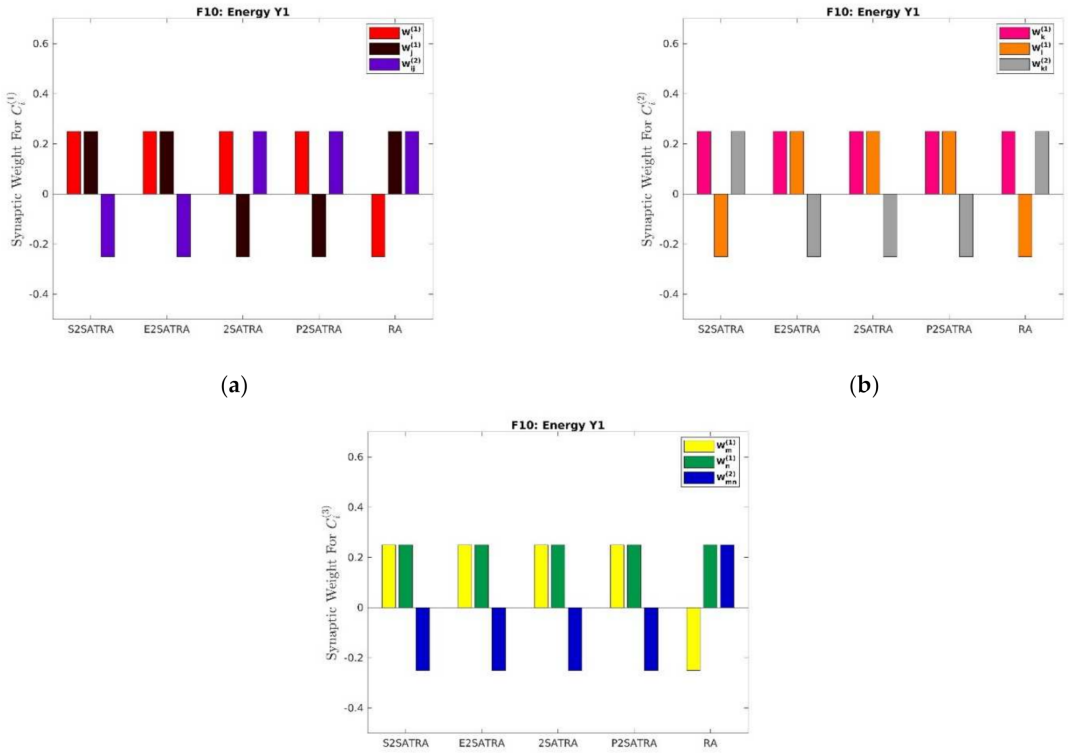


Figure 11. Synaptic weight analysis for F10: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.

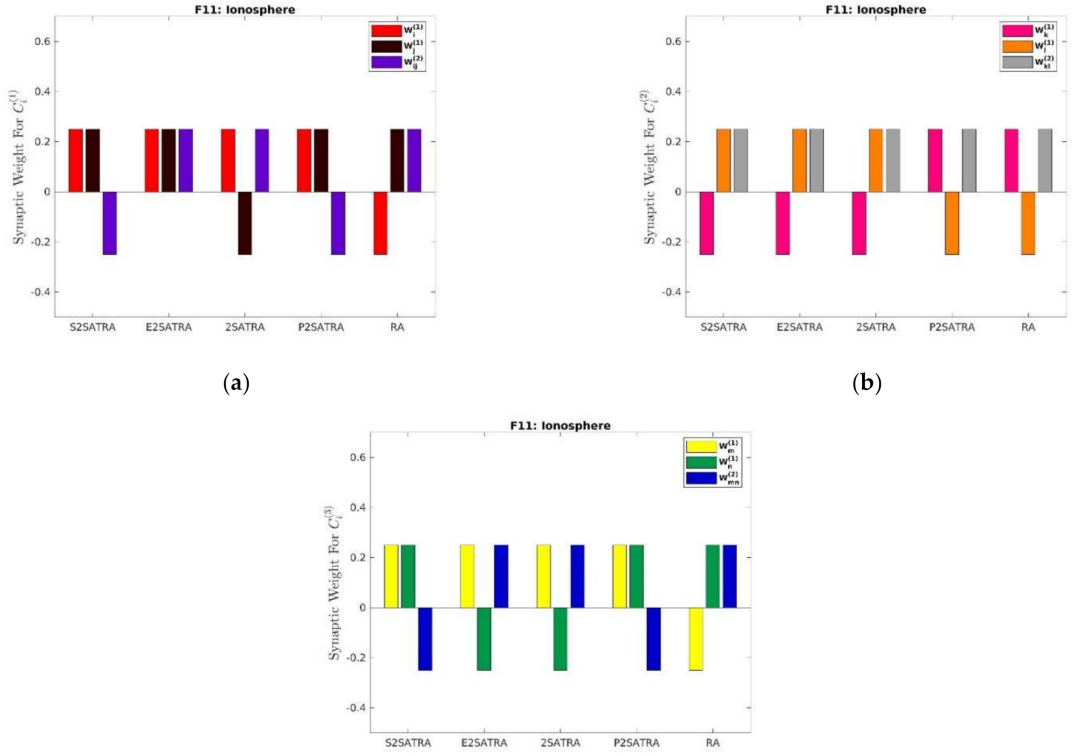


Figure 12. Synaptic weight analysis for F11: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.

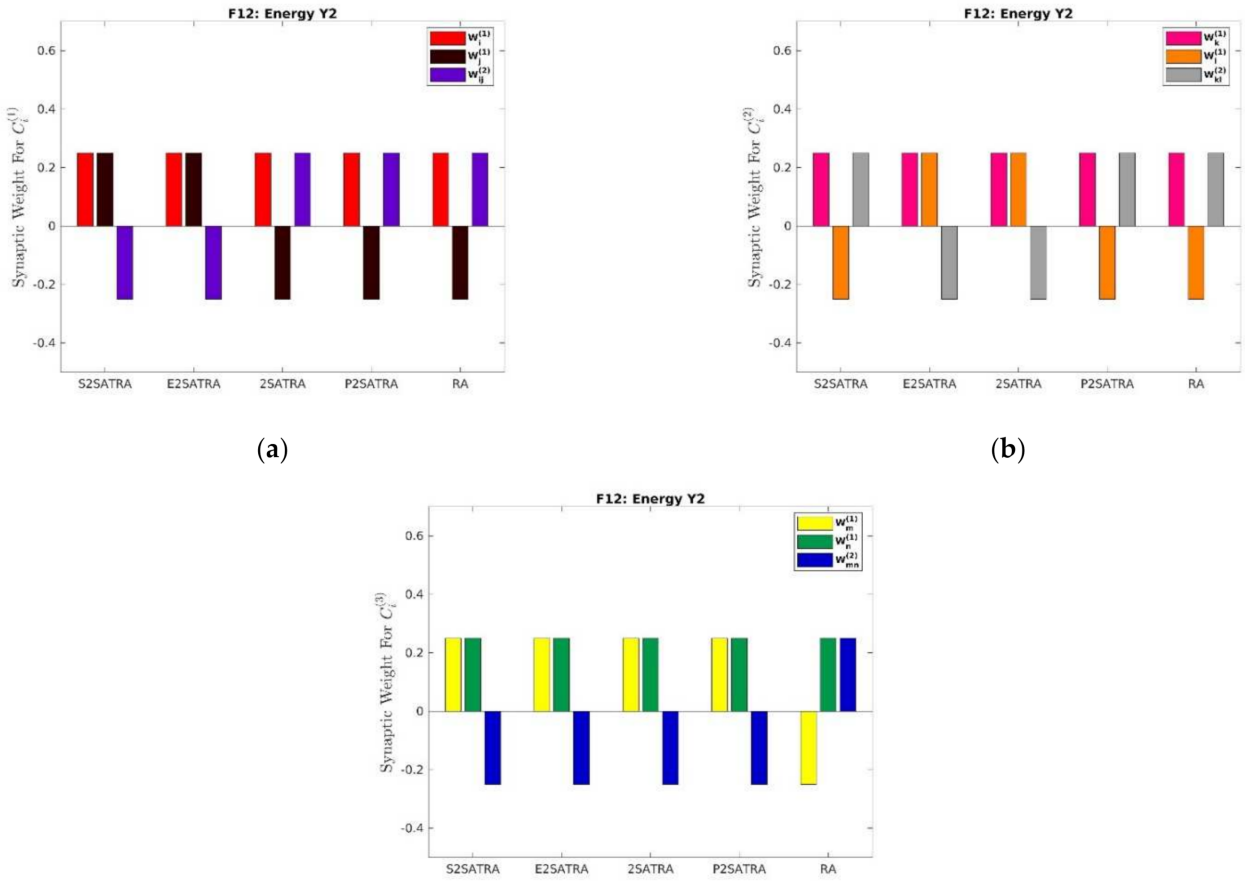


Figure 13. Synaptic weight analysis for F13: (**a**) $C_i^{(1)}$; (**b**) $C_i^{(2)}$ and (**c**) $C_i^{(3)}$.

(a) According to Tables 7 and 8, the worst performing correlation values which account for most of the weakly correlated values are F1, F5, F7, and F11. The weak correlation is determined after considering the absolute value of the correlation. Despite the low correlation value, S2SATRA is still able to avoid attributes with no correlation at all.

(b) The best performing correlation datasets are F4, F9, F10, and F12 where all the attributes of interest are selected for learning. The optimal selection by S2SATRA has a good agreement with high accuracy of the induced logic compared to the existing model.

(c) F6 and F8 are the only datasets that partially achieve the optimal number of attributes with a high correlation with $Q_{2SAT}^{k_i}$. These datasets are reported to be highly correlated and the results have slightly low accuracy in terms of induced logic.

(d) Overall, we can also conclude that S2SATRA does not require any randomized attribute selection because all correlation values agree with the association threshold value.

7.3. Error Analysis

Tables 9 and 10 demonstrate the error evaluation for all the logic mining models. The S2SATRA model outperforms all logic mining models in terms of *RMSE* and *MAE*. Note that the improvement ratio is considered by taking into account the differences between the error value divided with the error produced by logic mining.

Table 6. Correlation analysis (ρ) for 8 sampled attributes for F1–F6.

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
F1								
Correlation	**0.352**	−0.004	**0.335**	0.097	**0.211**	**−0.178**	**0.166**	**0.157**
P	$\mathbf{5.4 \times 10^{-159}}$	7.7×10^{-1}	$\mathbf{2.1 \times 10^{-69}}$	7.8×10^{-13}	$\mathbf{4.3 \times 10^{-56}}$	$\mathbf{2.7 \times 10^{-40}}$	$\mathbf{3.9 \times 10^{-35}}$	$\mathbf{1.6 \times 10^{-31}}$
Decision H_0	**Reject**	Accept	**Reject**	Reject	**Reject**	**Reject**	**Reject**	**Reject**
F2								
Correlation	−0.014	**0.374**	**0.247**	**0.720**	**0.458**	**0.406**	0.032	**0.115**
P	7.2×10^{-1}	$\mathbf{2.7 \times 10^{-24}}$	$\mathbf{5.1 \times 10^{-11}}$	$\mathbf{1.9 \times 10^{-111}}$	$\mathbf{3.9 \times 10^{-37}}$	$\mathbf{7.9 \times 10^{-29}}$	4.1×10^{-1}	$\mathbf{2.0 \times 10^{-3}}$
Decision H_0	Accept	**Reject**	**Reject**	**Reject**	**Reject**	**Reject**	Accept	**Reject**
F3								
Correlation	**0.366**	0.202	**0.344**	0.230	**0.376**	**0.581**	**−0.338**	**0.432**
P	$\mathbf{2.0 \times 10^{-3}}$	1.0×10^{-1}	$\mathbf{4.0 \times 10^{-3}}$	6.2×10^{-2}	$\mathbf{2.0 \times 10^{-3}}$	$\mathbf{2.4 \times 10^{-7}}$	$\mathbf{5.0 \times 10^{-3}}$	$\mathbf{3.0 \times 10^{-4}}$
Decision H_0	**Reject**	Accept	**Reject**	Accept	**Reject**	**Reject**	**Reject**	**Reject**
F4								
Correlation	**0.687**	**0.678**	**0.686**	0.580	**0.752**	**0.636**	0.604	0.284
P	$\mathbf{9.3 \times 10^{-99}}$	$\mathbf{4.2 \times 10^{-95}}$	$\mathbf{2.3 \times 10^{-98}}$	5.4×10^{-64}	$\mathbf{1 \times 10^{-127}}$	$\mathbf{1.4 \times 10^{-80}}$	1.1×10^{-70}	2.1×10^{-14}
Decision H_0	**Reject**	**Reject**	**Reject**	Reject	**Reject**	**Reject**	Reject	Reject
F5								
Correlation	0.081	**−0.278**	**0.250**	**0.269**	0.077	**0.189**	**−0.271**	**0.214**
P	1.4×10^{-1}	$\mathbf{2.8 \times 10^{-7}}$	$\mathbf{4.4 \times 10^{-6}}$	$\mathbf{7.5 \times 10^{-7}}$	1.6×10^{-1}	$\mathbf{5 \times 10^{-4}}$	$\mathbf{5.8 \times 10^{-7}}$	$\mathbf{0.0 \times 10^{-1}}$
Decision H_0	Accept	**Reject**	**Reject**	**Reject**	Accept	**Reject**	**Reject**	**Reject**
F6								
Correlation	**0.737**	**0.144**	−0.010	**−0.447**	−0.016	**−0.595**	**0.521**	**0.735**
P	$\mathbf{0.0 \times 10^{-1}}$	$\mathbf{8.8 \times 10^{-68}}$	2.3×10^{-1}	$\mathbf{0.0 \times 10^{-1}}$	5.5×10^{-2}	$\mathbf{0.0 \times 10^{-1}}$	$\mathbf{0.0 \times 10^{-1}}$	$\mathbf{0.0 \times 10^{-1}}$
Decision H_0	**Reject**	**Reject**	Accept	**Reject**	Accept	**Reject**	**Reject**	**Reject**

Table 7. Correlation analysis (ρ) for 8 sampled attributes for F7–F12.

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
F7								
Correlation	**−0.086**	−0.0324	**−0.397**	**0.393**	**−0.091**	**−0.180**	−0.072	**−0.133**
P	$\mathbf{4.1 \times 10^{-13}}$	7.9×10^{-172}	$\mathbf{2.0 \times 10^{-264}}$	$\mathbf{4.3 \times 10^{-259}}$	$\mathbf{2.7 \times 10^{-14}}$	$\mathbf{1.4 \times 10^{-52}}$	1.8×10^{-9}	$\mathbf{5.4 \times 10^{-29}}$
Decision H_0	**Reject**	Reject	**Reject**	**Reject**	**Reject**	**Reject**	Reject	**Reject**
F8								
Correlation	**0.518**	**−0.847**	0.489	**−0.499**	0.266	**−0.617**	**−0.788**	**−0.634**
P	$\mathbf{1.3 \times 10^{-13}}$	$\mathbf{2.7 \times 10^{-50}}$	4.3×10^{-12}	$\mathbf{1.3 \times 10^{-12}}$	3.0×10^{-4}	$\mathbf{4.4 \times 10^{-20}}$	$\mathbf{5.9 \times 10^{-39}}$	$\mathbf{2.2 \times 10^{-21}}$
Decision H_0	**Reject**	**Reject**	Reject	**Reject**	Reject	**Reject**	**Reject**	**Reject**
F9								
Correlation	0.178	0.009	**0.819**	**0.901**	**0.649**	**0.611**	**0.592**	**0.966**
P	1.0×10^{-2}	8.9×10^{-1}	$\mathbf{6.7 \times 10^{-52}}$	$\mathbf{4.2 \times 10^{-77}}$	$\mathbf{2.5 \times 10^{-26}}$	$\mathbf{9.7 \times 10^{-23}}$	$\mathbf{3.6 \times 10^{-21}}$	$\mathbf{3.4 \times 10^{-124}}$
Decision H_0	Reject	Accept	**Reject**	**Reject**	**Reject**	**Reject**	**Reject**	**Reject**
F10								
Correlation	**0.671**	**−0.704**	**0.473**	**−0.914**	**0.933**	**0.995**	0.156	−0.055
P	$\mathbf{4.4 \times 10^{-50}}$	$\mathbf{4.2 \times 10^{-57}}$	$\mathbf{3.3 \times 10^{-22}}$	$\mathbf{3.8 \times 10^{-147}}$	$\mathbf{2.2 \times 10^{-166}}$	$\mathbf{0.0 \times 10^{-1}}$	3.0×10^{-3}	2.9×10^{-1}
Decision H_0	**Reject**	**Reject**	**Reject**	**Reject**	**Reject**	**Reject**	Reject	Reject
F11								
Correlation	0.011	0.072	**0.310**	**0.315**	**0.345**	**0.581**	**0.336**	**0.306**
P	8.4×10^{-1}	1.8×10^{-1}	$\mathbf{3.0 \times 10^{-9}}$	$\mathbf{1.6 \times 10^{-9}}$	$\mathbf{3.1 \times 10^{-11}}$	$\mathbf{5.0 \times 10^{-33}}$	$\mathbf{9.9 \times 10^{-11}}$	$\mathbf{4.9 \times 10^{-9}}$
Decision H_0	Accept	Accept	**Reject**	**Reject**	**Reject**	**Reject**	**Reject**	**Reject**
F12								
Correlation	**0.674**	**−0.710**	**0.435**	**−0.900**	**0.924**	0.022	**0.136**	−0.051
P	$\mathbf{8.4 \times 10^{-51}}$	$\mathbf{2.1 \times 10^{-58}}$	$\mathbf{1.2 \times 10^{-18}}$	$\mathbf{3.7 \times 10^{-136}}$	$\mathbf{6.2 \times 10^{-157}}$	6.8×10^{-1}	$\mathbf{9.0 \times 10^{-3}}$	3.3×10^{-1}
Decision H_0	**Reject**	**Reject**	**Reject**	**Reject**	**Reject**	Accept	**Reject**	Accept

Table 8. Improved RA [14].

Parameter	Parameter Value
Neuron Combination	100
Number of Learning (Ω)	100
Logical Rule	Q_{2SAT}
No_Neuron String	100
Selection_Rate	0.1

Table 9. *RMSE* for all logic mining models. The bracket indicates the ratio of improvement and * indicates division by zero. A negative ratio implies the method outperform the proposed method. *P* is obtained from the paired Wilcoxon rank test and ** indicates the model with significant inferiority compared to the superiority model.

Dataset	S2SATRA	E2SATRA	2SATRA	P2SATRA	RA
F1	**18.125**	**18.125 (0)**	31.034 (0.416)	31.034 (0.416)	676.174 (0.973)
F2	**5.417**	15.289 (0.646)	17.215 (0.685)	15.891 (0.659)	76.601 (0.929)
F3	**0.000**	0.920 (1.000)	3.849 (1.000)	0.770 (1.000)	49.267 (1.000)
F4	**0.569**	3.695 (0.846)	1.563 (0.636)	**0.569 (0.000)**	1.563 (0.636)
F5	**1.572**	11.708 (0.866)	14.329 (0.890)	7.514 (0.791)	20.794 (0.9244)
F6	**25.134**	32.199 (0.219)	106.945 (0.765)	30.124 (0.166)	958.121 (0.974)
F7	**18.839**	34.874 (0.460)	46.760 (0.597)	20.745 (0.092)	98.791 (0.809)
F8	**1.179**	10.371 (0.886)	11.078 (0.894)	1.414 (0.166)	10.371 (0.886)
F9	**0.655**	2.619 (0.749)	8.510 (0.923)	**0.655 (0.000)**	12.001 (0.945)
F10	**0.000**	3.932 (1.000)	24.413 (1.000)	**0.000 (*)**	54.367 (1.000)
F11	**2.556**	5.112 (0.500)	19.369 (0.868)	2.695 (0.052)	59.337 (0.957)
F12	**0.000**	0.000 (*)	10.650 (1.000)	**0.000 (*)**	7.865(1.000)
(+/=/−)	-	11/1/0	12/0/0	8/4/0	12/0/0
Avg	**6.170**	11.814	24.643	9.284	168.761
Std	**9.039**	11.528	28.760	12.008	4.662
min	**0.000**	0.000	1.563	0.000	1.563
max	**18.839**	34.874	106.945	31.034	958.121
Avg Rank	**1.250**	2.917	4.083	2.083	4.667
P		0.005 **	0.002 **	0.012 **	0.002 **

Table 10. *MAE* for all logic mining models. The bracket indicates the ratio of improvement and * indicates division by zero. A negative ratio implies the method outperform the proposed method. *P* is obtained from the paired Wilcoxon rank test and ** indicates the model with significant inferiority compared to the superiority model.

Dataset	S2SATRA	E2SATRA	2SATRA	P2SATRA	RA
F1	**0.387**	**0.387** (0.000)	0.663 (0.416)	0.663 (0.416)	12.452 (0.973)
F2	**0.326**	0.920 (0.646)	1.109 (0.706)	0.957 (0.659)	4.601 (0.929)
F3	**0.000**	0.741 (1.000)	0.741 (1.000)	**0.148** (1.000)	9.481 (1.000)
F4	**0.040**	0.263 (0.848)	0.111 (0.640)	0.040 (0.000)	0.111 (0.640)
F5	**0.137**	1.023 (0.866)	1.252 (0.891)	0.656 (0.791)	1.817 (0.925)
F6	**0.330**	0.423 (0.220)	1.404 (0.765)	0.396 (0.167)	12.582 (0.974)
F7	**0.352**	0.652 (0.460)	0.874 (0.597)	0.388 (0.093)	1.846 (0.809)
F8	**0.139**	1.222 (0.886)	1.306 (0.894)	0.167 (0.168)	1.222 (0.886)
F9	**0.071**	0.286 (0.752)	0.929 (0.924)	**0.071** (0.000)	1.310 (0.946)
F10	**0.000**	0.322 (1.000)	2.000 (1.000)	**0.000** (*)	4.456 (1.000)
F11	0.233	0.467 (0.501)	1.631 (0.857)	**0.227** (−0.026)	5.417 (0.957)
F12	**0.000**	**0.000** (*)	0.872 (1.000)	**0.000** (*)	0.644 (1.000)
(+/=/−)	-	10/2/0	12/0/0	7/4/1	12/0/0
Avg	**0.168**	0.559	1.074	0.309	310.235
Std	**0.151**	0.359	0.493	0.309	4.505
min	**0.000**	**0.000**	0.111	**0.000**	0.111
max	**0.387**	1.222	2.000	0.957	12.582
Avg Rank	**1.333**	2.958	4.041	2.000	4.667
P		0.002 **	0.002 **	0.003 **	0.003 **

A high value of *RMSE* demonstrates the high deviation of the error compared with the $Q_{2SAT}^{k_i}$. S2SATRA ranks first on 12 datasets. The "+", "−", and "=" in the results column indicate that S2SATRA is superior, inferior, and equal to the comparison algorithm, respectively. The "Avg" indicates the corresponding algorithm's average of the Friedman test for 12 datasets. The rank represents the ranking of the "Avg Rank". Although the value S2SATRA is the lowest compared to other logic mining model, the *RMSE* value is

high, which shows that the error is deviated from the mean of the error for the whole $Q_{2SAT}^{k_i}$. According to Tables 9 and 10, there are several winning points for S2SATRA, which are as follows.

(a) In terms of individual *RMSE* and *MAE*, S2SATRA outperforms all the existing logic mining models which extract the logical rule from the datasets.
(b) There were several datasets that recorded zero error, such as in F3 and F10. In terms of *MAE*, S2SATRA achieved less than 0.5 for all the datasets, resulting in a lower mean *MAE* (0.168).
(c) Despite showing the best performance compared to all existing methods, the *RMSE* value for S2SATRA is still high for several datasets, such as in F1, F6, and F7. Although a high value of *RMSE* is recorded, the value is much lower compared to the other existing work.
(d) The Friedman test rank is conducted for all the datasets with $\alpha = 0.05$ and a degree of freedom of $df = 4$. The P for both *RMSE* and *MAE* are 1.27×10^{-7} $\left(\chi^2 = 37.33\right)$ and 2.09×10^{-7} $\left(\chi^2 = 36.68\right)$, respectively. Hence, the null hypothesis of equal performance for all the logic mining models is rejected. According to Tables 9 and 10 for all the datasets, S2SATRA has an average rank of 1.25 and 1.333 for *RMSE*, respectively, which is highest compared to other existing methods. The closest method that competes with S2SATRA is P2SATRA with an average rank of 2.083 and 2.000, respectively.
(e) Overall, the average *RMSE* and *MAE* for S2SATRA shows an improvement by 83.9% compared to the second best method which is P2SATRA. In this case, the optimal attribute selection contributes towards a lower value of *RMSE* and *MAE*.
(f) In addition, the Wilcoxon rank test is conducted to statistically validate the superiority of S2SATRA [40]. From the table, we observe that S2SATRA is the top-ranked logic mining model in terms of error analysis followed by P2SATRA, E2SATRA, 2SATRA, and RA.

P2SATRA is observed to achieve a competitive result where the 5 out of 12 datasets have the same error during the retrieval phase. This indicates that the conventional 2SATRA model can be further improved with a permutation operator. Despite the high permutation value (up to 1000 permutation/run) implemented in each dataset, most of the attributes in the P2SATRA are insignificant with respect to the final output. Hence, the accumulated testing error will be higher than the proposed S2SATRA. It is also worth noting that implementation of the permutation operator from P2SATRA benefits S2SATRA in terms of search space. In another perspective, an energy-based approach, E2SATRA, is able to obtain $Q_{2SAT}^{k_i}$ which can achieve the global minima energy but tends to get trapped in suboptimal solution. According to Tables 9 and 10, E2SATRA showed improvement in terms of error compared to the conventional 2SATRA but the induced logic only explores a limited search space. For example, the high accumulation error in F2–F8 were due to small number of $Q_{2SAT}^{k_i}$ produced by E2SATRA. The only advantage for E2SATRA compared to RA is the stability of the $Q_{2SAT}^{k_i}$ in finding the correction dataset generalisation. E2SATRA is reported to be slightly worse compared to P2SATRA, except for F8 and F10 where the error difference is 86.3% and 47.2%, respectively. Conventional 2SATRA and RA were reported to produce $Q_{2SAT}^{k_i}$ with the worst quality due to the wrong choice of attribute selection. Another interesting insight is that the modified RA from [14] tends to overlearn, which results in an accumulation of error. For instance, RA accumulates a large *RMSE* value in F1, F6, and F7, due to the rigid structure of $Q_{2SAT}^{k_i}$ during the learning phase and the testing phase of RA. Additionally, the rigid structure for $Q_{2SAT}^{k_i}$ in RA does not contribute to effective attribute representation. Overall, it can be seen that, compared with each comparison algorithm, S2SATRA has the greatest advantages on more than 10 datasets in terms of *RMSE* and *MAE*.

7.4. Accuracy, Precision, Sensitivity, F1-Score, and MCC

Figures 14 and 15 demonstrate the result for *F-score* and *Acc* for all the logic mining models. There are several winning points for S2SATRA according to both figures, which are as follows.

(a) In terms of *Acc*, S2SATRA achieved the highest *Acc* value in 11 out of 12 datasets. The closest model that competes with S2SATRA is P2SATRA. A similar observation in *F-score* is that S2SATRA achieves the highest value in 8 out of 12 datasets, while the closest model that competes with S2SATRA is P2SATRA.

(b) There were three datasets (F3, F10, and F12) that achieve $Acc = 1$, which means that S2SATRA can correctly predict the $Q_{test} = 1$ for all values of TP and TN. For the *F-score* value, there were three datasets that achieved $F = 1$ value, meaning that S2SATRA can correctly produce TP during the retrieval phase of HNN. In this context, $F = 1$ indicates the perfect precision and recall.

(c) There is no value for *F-score* for F5 for all the logic mining models because there is no TP in the testing data.

(d) According to the Figures 14 and 15, no value for $Acc < 0.8$ is reported and only F11 reports the lowest value of *F-score*. No *F-score* value in F5 indicates that there is no value of TP during the testing data. This justifies the superiority of the S2SATRA in differentiating TP and TN cases which is very crucial in logic mining.

(e) S2SATRA shows an average improvement in the *Acc* value ranging from 27.1% to 97.9%. This shows that the clustering capability of S2SATRA significantly improved while the error value remains low (refer Table 7 (A)). A similar observation is reported in *F-score*. S2SATRA shows an average improvement ranging from 30.1% until 75.7%. This also shows that the clustering capability of S2SATRA significantly improved while the error value remains low.

(f) The Friedman test rank is conducted for all the datasets with $\alpha = 0.05$ and a degree of freedom of $df = 4$. The P both for *Acc* and *F-score* are 4.26×10^{-7} $\left(\chi^2 = 35.18\right)$ and 8.00×10^{-6} $\left(\chi^2 = 29.03\right)$, respectively. Hence, the null hypothesis of equal performance for all the logic mining models is rejected. S2SATRA has an average rank of 1.375 which is the highest compared to other existing method for *Acc*. The closest method that competes with S2SATRA is P2SATRA with an average rank of 2.083. On the other hand, S2SATRA has an average rank of 1.458 which is the highest compared to other existing logic mining models for *F-score.* The closest method that competes with S2SATRA is P2SATRA with an average rank of 2.333. Both results statistically validate the superiority of S2SATRA compared to the existing work.

(g) In addition, the paired Wilcoxon rank test is conducted to statistically validate the superiority of the S2SATRA. From the table, we observed that S2SATRA is the top-ranked logic mining model in terms of *Acc* and *F-score* followed by P2SATRA, E2SATRA, 2SATRA, and RA.

Tables 11 and 12 demonstrate the result for *Pr* and *Se* for all the 2SATRA models. According to Table 7 (A), there are several winning points for S2SATRA, which are as follows.

(a) In terms of *Pr*, S2SATRA outperforms other logic mining model in 6 out of 12 datasets. The closest model that competes with S2SATRA is P2SATRA. For Se, S2SATRA outperforms other 2SATRA models in 7 out of 12 datasets. Similar to the *Pr* value, the closest model that competes with S2SATRA is P2SATRA.

(b) There were three datasets that achieve *Pr* = 1 value, which means that S2SATRA can correctly predict the $Q_{test} = 1$ in comparison with all the positive outcomes. For the *Se* value, four datasets achieved an $Se = 1$ value, which means that S2SATRA can correctly produce a positive result during the retrieval phase of HNN.

(c) No value for both *Pr* and *Se* is reported for F5 because there is no positive outcome for these datasets.

(d) The only datasets that achieved *Pr* < 0.8 were F8 and F11. This shows that 2SATRA has good capability in differentiating a positive result with a negative result. A similar

observation is reported in *Se* where the only datasets that achieved $Se < 0.8$ were F8 and F11. Hence, S2SATRA has a competitive capability to produce a positive result $Q_{test} = 1$ compared to other existing 2SATRA model.

(e) S2SATRA shows an average improvement in the *Pr* value, ranging from 12.3% to 61.2%. This shows that the clustering capability of S2SATRA significantly improved while the error value remained low (refer Table 11). A similar observation is reported in the *Se* result. S2SATRA shows an average improvement ranging from 1.8% to 63.9%. This also shows that the clustering capability of S2SATRA significantly improved while the error value remained low.

(f) According to the Friedman test rank for all the datasets, S2SATRA has an average rank of 1.458 which is the highest compared to other existing methods for *Pr*. The closest method that competes with S2SATRA is P2SATRA, with an average rank of 2.333. On the other hand, S2SATRA has an average rank of 1.375 which is the highest compared to other existing method for *Se*. The closest method that competes with S2SATRA is P2SATRA, with an average rank of 2.083. Both results statistically validate the superiority of S2SATRA compared to the other logic mining.

(g) In addition, the paired Wilcoxon rank test is conducted to statistically validate the superiority of S2SATRA. From the table, we observed that S2SATRA is the top-ranked logic mining model in terms of *Pr* and *Se*, as compared to most of the existing work.

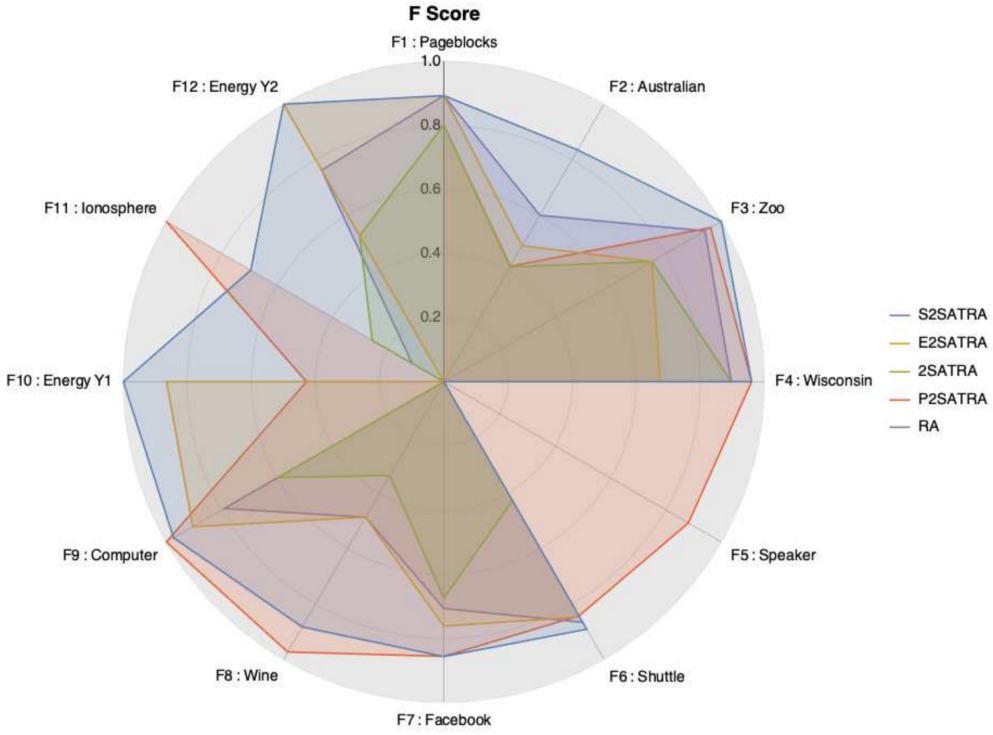


Figure 14. *F-score* for all logic mining models.

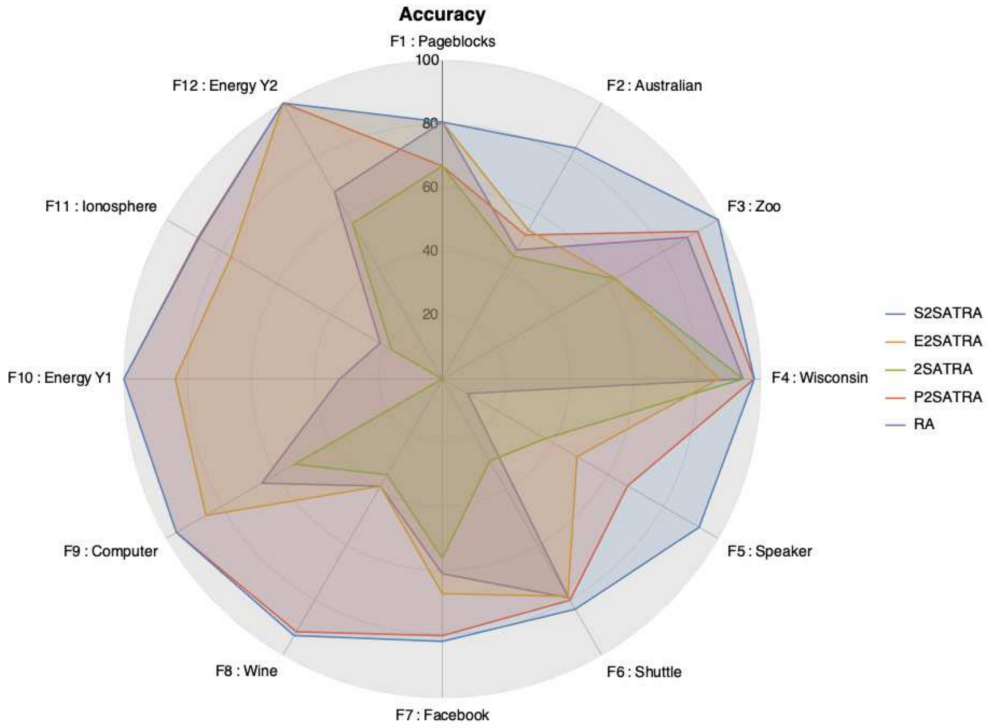


Figure 15. Accuracy for all the logic mining models.

Table 11. Precision (*Pr*) for all models. The bracket indicates the ratio of improvement and * indicates division by zero. A negative ratio implies the method outperform the proposed method. *P* is obtained from the paired Wilcoxon rank test and ** indicates the model with significant inferiority compared to the superiority model.

Dataset	S2SATRA	E2SATRA	2SATRA	P2SATRA	RA
F1	**0.826**	**0.826 (0)**	0.685 (0.205)	0.685 (0.206)	**0.826 (0)**
F2	0.934	**0.984** (−0.051)	0.443 (1.108)	0.385 (1.426)	0.902 (0.035)
F3	**1.000**	0.600 (0.667)	0.6 (0.667)	**1.000** (0)	0.960 (0.042)
F4	**0.942**	0.519 (0.815)	0.923 (0.021)	**0.942** (0)	0.923 (0.021)
F5	-	-	-	-	-
F6	0.854	0.737 (0.159)	0.330 (1.588)	**0.922 (−0.074)**	0.875 (−0.024)
F7	**0.992**	0.980 (0.012)	0.850 (0.167)	0.979 (0.013)	0.880 (0.127)
F8	0.792	0.875 (−0.095)	0.500 (0.584)	0.750 (0.056)	**0.875 (0.095)**
F9	0.966	**0.983 (−0.017)**	0.500 (0.932)	0.966 (0)	0.948 (0.019)
F10	**1.000**	**1.000 (0)**	0.000 (*)	**1.000 (0)**	0.000 (*)
F11	0.696	0.000 (-)	**0.909 (−0.2343)**	0.273 (1.549)	0.261 (1.667)
F12	**1.000**	**1.000** (0)	0.468 (1.137)	**1.000 (0)**	**1.000 (0)**
(+/=/−)	-	6/5/2	10/1/1	5/6/1	7/4/1
Avg	**0.909**	0.773 (0.175)	0.564 (0.612)	0.809 (0.123)	0.765 (0.188)
Std	**0.103**	0.307	0.274	0.261	0.324
Min	**0.696**	0.000	0.000	0.273	0.000
Max	**1.000**	**1.000**	0.923	1.000	**1.000**
Avg Rank	**1.458**	3.417	4.417	2.333	3.375
P		0.003 **	0.003 **	0.003 **	0.003 **

Table 12. Sensitivity (*Se*) for all logic mining models. The bracket indicates the ratio of improvement and * indicates division by zero. A negative ratio implies the method that outperforms the proposed method. ** due to no positive outcome in the dataset. *P* is obtained from the paired Wilcoxon rank test and ** indicates a model with significant inferiority compared to the superiority model.

Dataset	S2SATRA	E2SATRA	2SATRA	P2SATRA	RA
F1	**0.971**	**0.971 (0)**	0.966 (0.0052)	0.966 (0.005)	**0.971 (0)**
F2	**0.755**	0.490 (0.541)	0.388 (0.946)	0.452 (0.670)	0.449 (−0.682)
F3	**1.000**	**1.000 (0)**	**1.000 (0)**	0.926 (0.080)	0.923 (0.083)
F4	**0.980**	0.964 (0.017)	0.8723 (0.123)	**0.980 (0)**	0.873 (0.123)
F5	0.000 **	0.000 (**)	0.000 (**)	0.000 (**)	0.000 (**)
F6	0.934	**0.997 (−0.063)**	0.611 (0.528)	0.844 (0.107)	0.867 (0.078)
F7	**0.755**	0.624 (0.210)	0.560 (0.348)	0.741 (0.019)	0.592 (0.275)
F8	**1.000**	0.339 (1.950)	0.255 (2.922)	**1.000 (0)**	0.339 (1.950)
F9	**0.982**	0.838 (0.171)	0.744 (0.320)	0.982 (0)	0.679 (0.446)
F10	**1.000**	0.762 (0.312)	0.000 (*)	**1.000** (0)	0.000 (*)
F11	0.696	0.000 (*)	0.150 (3.64)	**1.000 (−0.304)**	0.073 (8.534)
F12	**1.000**	**1.000** (0)	0.600 (0.667)	**1.000** (0)	0.616 (0.6234)
(+/=/−)		7/4/1	10/2/0	5/6/1	10/2/0
Avg	**0.839**	0.666	0.512	0.824	0.532
Std	**0.287**	0.379	0.355	0.306	0.360
Min	**0.000 ****	**0.000 ****	**0.000 ****	**0.000 ****	**0.000 ****
Max	**1.000**	**1.000**	**1.000**	**1.000**	0.923
Avg Rank	**1.375**	3.167	4.708	2.083	3.667
P		0.612	0.086	0.003 **	0.084

Table 13 demonstrates MCC analysis for all logic mining models. According to Table 13, several winning points for S2SATRA are as follows.

(a) In terms of MCC, S2SATRA achieved the most optimal MCC value for 7 out of 12 datasets. The closest model that competes with S2SATRA is P2SATRA. On average, the logic mining model is reported to obtain the worst result where the MCC value approaches zero.

(b) There were three datasets (F3, F10, and F12) that achieve an $MCC = 1$ value which means that S2SATRA which produced Q_{test} represents perfect prediction.

(c) No value for MCC is reported for F5 because there is no positive outcome for this dataset.

(d) The only dataset that approaches zero MCC is F1. This shows that S2SATRA has good capability in differentiating all domain of the confusion matrix (TP, FP, TN, and FN).

(e) By taking into account the absolute value of MCC, S2SATRA shows an average improvement in the MCC value ranging from 35.9% until 3839%. This shows that the clustering capability of S2SATRA significantly improved while the error value remained low (refer Table 13).

(f) The Friedman test rank is conducted for all the datasets with $\alpha = 0.05$ and a degree of freedom of $df = 4$. The P for MCC is $1.09 \times 10^{-11} (\chi^2 = 57.26)$. Hence, the null hypothesis of equal performance for all the logic mining models was rejected. S2SATRA has an average rank of 1.363 which is the highest compared to other existing logic mining for MCC. The closest method that competes with S2SATRA is P2SATRA with an average rank of 2.955. This result statistically validates the superiority of S2SATRA compared to the existing work.

(g) In addition, the paired Wilcoxon rank test is conducted to statistically validate the superiority of S2SATRA. From the table, we observed that S2SATRA is the top-ranked logic mining model in terms of MCC compared to most existing work.

Table 13. *MCC* for all logic mining models. *P* is obtained from the paired Wilcoxon rank test and ** indicates the models with significant inferiority compared to the superiority model.

Dataset	S2SATRA	E2SATRA	2SATRA	P2SATRA	RA
F1	−0.070	−0.071	**−0.104**	**−0.104**	−0.071
F2	**0.693**	0.270	−0.109	0.015	0.039
F3	**1.000**	0.316	0.316	-	−0.055
F4	**0.948**	0.647	0.860	0.948	0.859
F5	-	-	-	-	-
F6	0.556	**0.595**	−0.406	0.301	0.489
F7	**0.679**	0.411	0.106	0.642	0.226
F8	**0.847**	0.028	0.365	0.816	0.028
F9	**0.918**	0.659	0.107	0.918	−0.129
F10	**1.000**	0.713	**−1.000**	**1.000**	−0.453
F11	**0.623**	−0.101	−0.064	0.490	−0.442
F12	**1.000**	**1.000**	0.137	**1.000**	0.453
Avg Rank	**1.363**	3.045	3.818	2.955	3.818
Mean	**0.745**	0.406	0.019	0.548	0.086
Std	**0.316**	0.355	0.469	0.412	0.397
(+/=/−)		9/2/1	10/1/1	6/5/1	10/1/1
P		0.011 **	0.003 **	0.018 **	0.005 **

7.5. McNemar's Statistical Test

To evaluate whether there is any significant difference between the performance of the two logic mining models, McNemar's test is performed. According to [38], McNemar is the only test that has acceptable Type 1 error and can validate the performance of the 2SATRA model. The normal test statistics are as follows:

$$Z_{ij} = \frac{f_{ij} - f_{ji}}{\sqrt{f_{ij} + f_{ji}}} \tag{23}$$

where Z_{ij} is a measure of significance of the accuracy obtained by model i and j, while f_{ij} is the number of cases where logic mining is correctly classified by model i but incorrectly classified by model j. A similar description is given for the notation f_{ij}. In this experiment, a 5% level of significance is used. The null hypothesis dictates a pair from the logic mining model with no difference in disagreement. The performance of classification accuracy is said to differ significantly if $|Z_{ij}| > 1.96$. Note that, a positive value of Z_{ij} means the model i performs better than model j. Tables 14 and 15 presents the result of the McNemar's test for all the logic mining models. Several winning points for S2SATRA are discussed below.

(a) S2SATRA is reported to be statistically significant (in bold) in more than half of the datasets. The only dataset that has no statistical significance is F4 where S2SATRA only significantly differs with E2SATRA.
(b) In terms of statistical performance, S2SATRA is shown to be significantly better compared to other logic mining model. For instance, there is no negative test regarding the statistics found for S2SATRA (refer row) in comparison to the other 2SATRA model. The lowest test statistics value for S2SATRA is zero.
(c) The best statistical performance for S2SATRA is in F2, F5, and F6 where all the existing methods are significantly different and worse (indicated in the positive value). The second best statistical performances are F7 and F8 where at least one logic mining model is statistically insignificant but with a statistically better result.
(d) In addition, results from the McNemar test indicates the superiority of S2SATRA in distinguishing both correct and incorrect outcomes compared to the existing method.

Table 14. McNemar's statistical test for F1–F5.

		S2SATRA	E2SATRA	2SATRA	P2SATRA	RA
F1	**S2SATRA**	-	0.000	**9.128**	**9.128**	0.000
	ES2SATRA		-	**9.128**	**9.128**	0.000
	2SATRA			-	0.000	**−9.128**
	P2SATRA				-	**−9.128**
	RA					-
F2	**S2SATRA**	-	**6.980**	**9.194**	**7.406**	**−23.263**
	ES2SATRA		-	**2.213**	0.426	**−26.567**
	2SATRA			-	−15.449	**78.536**
	P2SATRA				-	1.277
	RA					-
F3	**S2SATRA**	-	**3.051**	**2.722**	0.544	0.816
	ES2SATRA		-	−0.278	**−2.496**	**−2.219**
	2SATRA			-	**−2.177**	−1.905
	P2SATRA				-	0.272
	RA					-
F4	**S2SATRA**	-	**2.211**	0.704	0.000	0.704
	ES2SATRA		-	−1.508	**−2.211**	−1.508
	2SATRA			-	−0.704	0.000
	P2SATRA				-	0.704
	RA					-
F5	**S2SATRA**	-	**7.264**	**9.020**	**4.201**	**13.592**
	ES2SATRA		-	1.723	**−3.077**	**6.278**
	2SATRA			-	**−4.819**	**4.572**
	P2SATRA				-	**9.391**
	RA					-

Table 15. McNemar's statistical test for F6–F11.

		S2SATRA	E2SATRA	2SATRA	P2SATRA	RA
F6	**S2SATRA**	-	**4.996**	**57.849**	**3.529**	**4.457**
	ES2SATRA		-	**52.853**	−1.467	−0.539
	2SATRA			-	**−54.321**	**−53.392**
	P2SATRA				-	0.929
	RA					-
F7	**S2SATRA**	-	**11.339**	**19.744**	1.348	**16.070**
	ES2SATRA		-	**8.405**	**−9.991**	**4.731**
	2SATRA			-	**−18.396**	**−3.674**
	P2SATRA				-	**14.722**
	RA					-
F8	**S2SATRA**	-	**6.500**	**7.000**	0.167	**6.500**
	ES2SATRA		-	−0.500	**−6.333**	0.000
	2SATRA			-	**−6.833**	**−4.157**
	P2SATRA				-	**6.333**
	RA					-
F9	**S2SATRA**	-	1.389	**5.555**	0.000	**4.012**
	ES2SATRA		-	**4.166**	−1.389	**2.623**
	2SATRA			-	**−5.555**	**−1.543**
	P2SATRA				-	**4.012**
	RA					-
F10	**S2SATRA**	-	**2.781**	**17.263**	0.000	**11.702**
	ES2SATRA		**-**	**14.482**	**−2.781**	**8.921**
	2SATRA			**-**	**−17.263**	**−5.561**
	P2SATRA				**-**	**11.702**
	RA					-
F11	**S2SATRA**	-	**1.807**	**11.204**	−1.052	**10.199**
	ES2SATRA		-	**9.470**	**−2.785**	**8.391**
	2SATRA			-	**−11.791**	**−1.424**
	P2SATRA				-	**10.832**
	RA					-
F12	**S2SATRA**	-	0.000	**7.531**	0.000	**5.561**
	ES2SATRA		-	**7.531**	0.000	**5.561**
	2SATRA			-	**−7.531**	**−1.970**
	P2SATRA				-	**5.561**
	RA					-

8. Discussion

The optimal logic mining model requires pre-processing structures for neurons before the Q_{best} can be learned by HNN. Currently, the logic mining model specifically optimizes the logic extraction from the dataset without considering the optimal Q_{best}. The mechanism that optimizes the optimal neuron relationship before the learning can occur is remain unclear. In this sense, identifying a specific pair of neurons for Q_{best} will facilitate the dataset generalization and reduce computational burden.

As mentioned in the theory section, S2SATRA is not merely a modification of a conventional logic mining model, but rather it is a generalization that absorbs all the conventional models. Thus, S2SATRA not only inherits many properties from a conventional logic mining model but it adds supervised property that reduces the search space of the optimal Q_i^B. The question that we should ponder is: what is the optimal Q_{best} for the logic mining model? Therefore, it is important to discuss the properties of S2SATRA against the conventional logic mining model in extracting optimal logical rule from the dataset. According to the previous logic mining model, such as [20,21,31], the quality of attributes is not well assessed since the attributes were randomly assigned. For instance, [21] achieved high accuracy for specific combination of attributes but the quality of different combination of the attributes will result in low accuracy due to a high local minima solution. A similar neuron structure can be observed in E2SATRA, as proposed by [24], because the choice of neurons is similar during the learning phase. Practically speaking, this learning mechanism [20–22,31] is natural in real life because the neuron assignment is based on trial and error. However, the 2SATRA model needs to sacrifice the accuracy if there is no optimum neuron assignment. By adding permutation property, as carried out in Kasihmuddin et al. [30], P2SATRA is able to increase the search space of the model in the expense of higher computational complexity. To put things into perspective, 10 neurons require learning 18,900 of Q_{best} learning for each neuron combination before the model can arrive to the optimal result. Unlike our proposed model, S2SATRA can narrow down the search space by checking the degree of association among the neurons before permutation property can take place. Supervised features of S2SATRA recognized the pattern produced by the neurons and align it with the Q_{best} clause. Thus, the mutual interaction between association and permutation will optimally select the best neuron combination.

As reported in Tables 7 and 8, the number of associations for analysis required for n attributes to create optimal Q_{best} was reduced by $\frac{1}{n}{}^nC_6$. In other words, the probability of P2SATRA to extract optimal Q_{best} is lower compared to the S2SATRA. As the Q_{best} supplied to the network has changed, the retrieval property of the S2SATRA model will improve. The best logic mining model demonstrates a high value of TP and TP with a minimized value of FP and FN. P2SATRA is observed to outperform the conventional logic mining in terms of performance metrics because P2SATRA can utilize the permutation attributes. In this case, the higher the number of permutations, the higher probability for the P2SATRA to achieve correct TP and TN. Despite a robust permutation feature, P2SATRA failed to disregard the non-significant attributes which leads to $Q_i^{learn} = 1$. Despite achieving high accuracy, the retrieved final neuron state is not interpretable. E2SATRA is observed to outperform 2SATRA in terms of accuracy because the induced logic in E2SATRA is the only amount in the final state that reached global minimum energy. The dynamic of the induced logic in E2SATRA follows the convergence of the final state proposed in [22] where the final state will converge to the nearest minima. Although all the final state in E2SATRA is guaranteed to achieve global minimum energy, the choice of attribute that is embedded to the logic mining model is not well structured. Similar to 2SATRA and P2SATRA, the interpretation of the final attribute will be difficult to design. In another development, 2SATRA is observed to outperform the RA proposed by [14] in terms of all performance metric. Although the structure of RA is not similar to 2SATRA in creating the Q_i^{learn}, the induced logic Q_i^B has a general property of $Q_{HORNSAT}$. In this case, $Q_{HORNSAT}$ is observed to create a rigid induced logic (at most 1 positive literal) and can reduce the

possible solution space of the RA. In this case, we only consider the dataset that satisfies the $Q_{HORNSAT}$ that will lead to $Q^{test} = 1$.

In contrast, S2SATRA employs a flexible Q_{2SAT} logic which accounts for both positive and negative literal. This structure is the main advantage over the traditional RA proposed by [14]. S2ATRA is observed to outperform the rest of the logic mining model due the optimal choice of attributes. In terms of feature, S2SATRA can capitalize the energy feature of E2SATRA and the permutation feature of P2SATRA. Hence, the induced logic obtained will always achieve global minimum energy and only relevant attribute $\rho < \alpha$ will be chosen to be learned in HNN. Another way to explain the effectiveness of logic mining is the ability to consistently find the correct logical rule to be learned by HNN. Initially, all logic mining models begin with HNN which has too many ineffective synaptic weights due to suboptimal features. In this case, S2SATRA can reduce the inconsistent logical rule that leads to suboptimal synaptic weight.

S2SATRA is reported to outperform almost all the existing logic mining models in terms of all performance metrics. S2SATRA has the capability to differentiate between $TP(Q_i^B = 1)$ and $TP(Q_i^B = -1)$, which leads to high *Acc* and *F-score* values. Since S2SATRA is able to obtain more $TP(Q_i^B = 1)$, the *Pr* and *Sen* will increase compared to the other existing methods. In terms of *Pr* and *Sen*, S2SATRA is reported to succesfully predict $Q_i^B = 1$ during the retrieval phase. In other words, the existing 2SATRA model is less sensitive to the positive outcome which leads to a lower value of *Pr* and *Se*. It is worth mentioning that the overfitting nature of the retrieval phase will lead to Q_i^B which can only produce more positive neuron states. This phenomenon was obvious in the existing method where the HNN tends to converge to only a few final states. This result has a good agreement with the McNemar's test where the performance of S2SATRA is significantly different from the existing method. The optimal arrangement of the Q_i^B signifies the importance of the association among the attributes towards the retrieval capability of the S2SATRA. Without proper arrangement, the obtained Q_i^B tends to overfit which leads to a high classification error. S2SATRA can only utilize correlation analysis during the pre-processing stage because correlation analysis provides preliminary connection between the attribute and Q_i^{learn}.

It is worth noting that although there are many developments of the supervised learning method, such as a decision tree, a support vector machine, etc., none of these methods can provide the best approximation to the logical rule. Most of the mentioned methods are numerically compatible as an individual classification task, but not as a classification via a logical rule. For instance, a decision tree is effective in classifying the outcome of the dataset but S2SATRA is more effective in generalizing the datasets in the form of induced logics. The obtained induced logic can be utilized for a similar classification task. In term of parameter settings, S2SATRA is not dependent on any free parameter. The only parameter that can improve S2SATRA is the number of *Trial*. Increasing the number of trials will increase the number of the final state that corresponds to the Q_i^B. The main problem with this modification is that increasing the number of trials will lead to an unnecessary high computation time. Hence, in this experiment, the number of *Trial* still follows the conventional settings in [38]. It is worth noting that S2SATRA achieved the lowest accuracy for F1. This is due to imbalanced data, which leads to non-optimal induced logic. Correlation analysis cannot discriminate the correct relationship between variables and Q_i^{learn}. Generally, S2SATRA improved the pre-processing phase of the logic mining which leads to an improved learning phase due to the correct combination of Q_i^{best}. The correct combination of Q_i^{best} will lead to optimal Q_i^B which can generalize the dataset.

Finally, we would like to discuss the limitations of the study. The limitation of the S2SATRA is the computation time due to the complexity of the learning phase. Since all logic mining models utilized the same learning model to maximize the fitness of the solution, computation time is not considered as a significant factor. As the number of attribute or clausal noise increases, the learning error will exponentially increase. Hence, metaheuristics and accelerating algorithms, such as in [41], are required to effectively

minimize the cost function in Equation (5) within a shorter computational time. This phenomenon can be shown when the number of neurons $NN \geq 20$ in the logic mining model is trapped in a trial-and-error state. In terms of satisfiability, all the proposed 2SATRA models do not consider non-satisfiable logical structure or $E_{Q_{2SAT}} \neq 0$, such as maximum satisfiability [42] and minimum satisfiability [43]. This is due to the nature of 2SATRA that only consider data point that leads to positive outcome or $Q^{learn} = 1$. In terms of network, HNN is chosen compared to other ANN structures, such as feedforward because feedback to the input is compatible to the cost function $E_{Q_{2SAT}}$. Another problem that might arise for feedforward ANN, such as within the radial basis function neural network (RBFNN), is the training choice. For instance, the work of [9,44] can produce a single induced logic due to the RBFNN structure. This will reduce the accuracy of the S2SATRA model. A convolution neural network (CNN) is not favoured because propositional logic only deals with bipolar representation and multiple layers only increase the computational cost for the S2SATRA. In another perspective, weighted satisfiability that randomly assign the negation of the neuron will reduce the generality of the induced logic. In this case, 2SATRA model must add one additional layer during the retrieval phase to obtain which logical weight yields the best accuracy. Unlike several learning environments in HNN-2SAT [45], learning iteration will not be restricted and will be terminated when $f_i = f_{NC}$. A restricted value of the learning iteration will lead to more induced logic trapped in local minimum energy. As a worst-case scenario, a logic mining model, such as E2SATRA, will not produce any induced logic in restricted learning environment. Hence, all the 2SATRA models exhibit the same learning rule via the Wan Abdullah method [6]. In addition, all the logic mining models, except for RA and conventional logic mining, follow the condition of $\left| H_{P_{S_i^{induced}}} - H^{\min}_{P_{S_i^{induced}}} \right| \leq \partial$. In this case, only induced logic that can achieve global minimum energy will be considered during the retrieval phase. This is supported by [33] where the final state of neuron that represents the induced logic will always converge to the nearest minimum. By employing the Wan Abdullah method and HTAF [4], the number of solutions that corresponds to the local minimum solution will reduce dramatically. The neuron combination is limited to only $COMBAX = 100$ because the higher the value of $COMBAX$, the higher the learning error and HNN tends to be trapped in a trial-and-error state.

The experimental results presented above indicate that the S2SATRA improved the classification performance more than other existing logic mining model and created more solution variation. Another interesting phenomenon we discovered is that supervised learning features in S2SATRA reduce the permutation effort in finding the optimal Q_i^{learn}. As a result, HNN can retrieve the logical rule to do with acquiring higher accuracy. Additionally, we observed that when a number of clausal noise was added, S2SATRA shows a better result compared to the existing model. It is expected that our work can give inspiration to other logic mining models, such as [20,21], to extract the logical rule effectively. The robust architecture of S2SATRA provides a good platform for the application of real-life bioinformatics. For instance, the proposed S2SATRA can extract the best logical rule that classifies single-nucleotide polymorphisms (SNPs) inside known genes associated with Alzheimer's disease. This can lead to large-scale S2SATRA design, which has the ability to classify and predict.

9. Conclusions and Future Work

In this paper, we proposed a new perspective in obtaining the best induced logic from real-life datasets. As in a standard logic mining model, the attribute selection was chosen randomly which leads to non-essential attributes and reduces the capability of the HNN to represent the dataset. To address the issue of randomness, a novel supervised learning (S2SATRA) capitalized the correlation filter among variables in the logical rule with respect to the logical outcome. In this case, the only attribute that has the best association value will be chosen during the pre-processing stage of S2SATRA. After obtaining the optimal Q_{best}, HNN can obtain the synaptic weight associated with the Q_{best} which minimizes the

cost function of the network. During the retrieval phase, the best combination of Q_i^B will be generated, thus creating the best Q_i^B that generalizes the logical rule of the datasets. The effectiveness of the proposed S2SATRA is illustrated by extensive experimental analysis that compares S2SATRA with several state-of-the-art logic mining methods. Experimental results demonstrate that S2SATRA can effectively produce more optimal Q_{best} which leads to the improved Q_i^B. In this case, S2SATRA was reported to outperform all the existing logic mining models in most of the performance metrics. Given the simplicity and flexibility of the S2SATRA, it is also worth implim3n5int other logical dimensions. For instance, it will be interesting to investigate the implementation of random k satisfiability proposed by [13,41] into the supervised learning-based reverse analysis method. By implementing the flexible logical rules, the generalization of the dataset will improve dramatically.

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

Funding: This research was supported by Ministry of Higher Education Malaysia for Transdisciplinary Research Grant Scheme (TRGS) with Project Code: TRGS/1/2020/USM/02/3/2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express special dedication to all of the researchers from AI Research Development Group (AIRDG) for the continuous support.

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

References

1. Malik, A.; Walker, C.; O'Sullivan, M.; Sinnen, O. Satisfiability modulo theory (smt) formulation for optimal scheduling of task graphs with communication delay. *Comput. Oper. Res.* **2018**, *89*, 113–126. [CrossRef]
2. Shukla, A.; Bhattacharyya, A.; Kuppusamy, L.; Srivas, M.; Thattai, M. Discovering vesicle traffic network constraints by model checking. *PLoS ONE* **2017**, *12*, e0180692. [CrossRef] [PubMed]
3. de Azevedo, G.H.I.; Pessoa, A.A.; Subramanian, A. A satisfiability and workload-based exact method for the resource constrained project scheduling problem with generalized precedence constraints. *Eur. J. Oper. Res.* **2021**, *289*, 809–824. [CrossRef]
4. Mansor, M.A.; Kasihmuddin, M.S.M.; Sathasivam, S. VLSI circuit configuration using satisfiability logic in Hopfield network. *Int. J. Intell. Syst. Appl.* **2016**, *8*, 22–29. [CrossRef]
5. Mansor, M.A.; Kasihmuddin, M.S.M.; Sathasivam, S. Enhanced Hopfield network for pattern satisfiability optimization. *Int. J. Intell. Syst. Appl.* **2016**, *8*, 27–33. [CrossRef]
6. Abdullah, W.A.T.W. Logic programming on a neural network. *Int. J. Intell. Syst.* **1992**, *7*, 513–519. [CrossRef]
7. Sathasivam, S. Upgrading logic programming in Hopfield network. *Sains Malays.* **2010**, *39*, 115–118.
8. Hamadneh, N.; Sathasivam, S.; Tilahun, S.L.; Choon, O.H. Learning logic programming in radial basis function network via genetic algorithm. *J. Appl. Sci.* **2012**, *12*, 840–847. [CrossRef]
9. Mansor, M.; Mohd Jamaludin, S.Z.; Mohd Kasihmuddin, M.S.; Alzaeemi, S.A.; Md Basir, M.F.; Sathasivam, S. Systematic boolean satisfiability programming in radial basis function neural network. *Processes* **2020**, *8*, 214. [CrossRef]
10. Kasihmuddin, M.S.M.; Mansor, M.A.; Sathasivam, S. Hybrid genetic algorithm in the Hopfield network for logic satisfiability problem. *Pertanika J. Sci. Technol.* **2017**, *25*, 139–151.
11. Bin Mohd Kasihmuddin, M.S.; Bin Mansor, M.A.; Sathasivam, S. Genetic algorithm for restricted maximum k-satisfiability in the Hopfield network. *Int. J. Interact. Multimed. Artif. Intell.* **2016**, *4*, 52–60.
12. Kasihmuddin, M.S.M.; Mansor, M.; Sathasivam, S. Robust Artificial Bee Colony in the Hopfield network for 2-satisfiability problem. *Pertanika J. Sci. Technol.* **2017**, *25*, 453–468.
13. Karim, S.A.; Zamri, N.E.; Alway, A.; Kasihmuddin, M.S.M.; Ismail, A.I.M.; Mansor, M.A.; Hassan, N.F.A. Random satisfiability: A higher-order logical approach in discrete Hopfield neural network. *IEEE Access* **2021**, *9*, 50831–50845. [CrossRef]
14. Sathasivam, S.; Abdullah, W.A.T.W. Logic mining in neural network: Reverse analysis method. *Computing* **2011**, *91*, 119–133.
15. Mansor, M.A.; Sathasivam, S.; Kasihmuddin, M.S.M. Artificial immune system algorithm with neural network approach for social media performance metrics. In Proceedings of the 25th National Symposium on Mathematical Sciences (SKSM25): Mathematical Sciences as the Core of Intellectual Excellence, Pahang, Malaysia, 27–29 August 2017.

16. Mansor, M.A.; Sathasivam, S.; Kasihmuddin, M.S.M. 3-satisfiability logic programming approach for cardiovascular diseases diagnosis. In Proceedings of the 25th National Symposium on Mathematical Sciences (SKSM25): Mathematical Sciences as the Core of Intellectual Excellence, Pahang, Malaysia, 27–29 August 2017.
17. Kasihmuddin, M.S.M.; Mansor, M.A.; Sathasivam, S. Satisfiability based reverse analysis method in diabetes detection. In Proceedings of the 25th National Symposium on Mathematical Sciences (SKSM25): Mathematical Sciences as the Core of Intellectual Excellence, Pahang, Malaysia, 27–29 August 2017.
18. Kasihmuddin, M.S.M.; Mansor, M.A.; Sathasivam, S. Students' performance via satisfiability reverse analysis method with Hopfield Neural Network. In Proceedings of the International Conference on Mathematical Sciences and Technology 2018 (MATHTECH2018): Innovative Technologies for Mathematics & Mathematics for Technological Innovation, Penang, Malaysia, 10–12 December 2018.
19. Kho, L.C.; Kasihmuddin, M.S.M.; Mansor, M.; Sathasivam, S. Logic mining in football. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *17*, 1074–1083.
20. Kho, L.C.; Kasihmuddin, M.S.M.; Mansor, M.; Sathasivam, S. Logic mining in league of legends. *Pertanika J. Sci. Technol.* **2020**, *28*, 211–225.
21. Alway, A.; Zamri, N.E.; Mohd Kasihmuddin, M.S.; Mansor, A.; Sathasivam, S. Palm oil trend analysis via logic mining with discrete Hopfield neural network. *Pertanika J. Sci. Technol.* **2020**, *28*, 967–981.
22. Mohd Jamaludin, S.Z.; Mohd Kasihmuddin, M.S.; Md Ismail, A.I.; Mansor, M.; Md Basir, M.F. Energy based logic mining analysis with Hopfield neural network for recruitment evaluation. *Entropy* **2021**, *23*, 40.
23. Peng, Y.L.; Lee, W.P. Data selection to avoid overfitting for foreign exchange intraday trading with machine learning. *Appl. Soft Comput.* **2021**, *108*, 107461.
24. Bottmer, L.; Croux, C.; Wilms, I. Sparse regression for large data sets with outliers. *Eur. J. Oper. Res.* **2022**, *297*, 782–794.
25. Tripepi, G.; Jager, K.J.; Dekker, F.W.; Zoccali, C. Linear and logistic regression analysis. *Kidney Int.* **2008**, *73*, 806–810. [CrossRef] [PubMed]
26. Yan, X.; Zhao, J. Application of Neural Network in National Economic Forecast. In Proceedings of the 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, 27–29 June 2018.
27. Sun, S.; Lu, H.; Tsui, K.L.; Wang, S. Nonlinear vector auto-regression neural network for forecasting air passenger flow. *J. Air Transp. Manag.* **2019**, *78*, 54–62. [CrossRef]
28. Khairi, M.T.M.; Ibrahim, S.; Yunus, M.A.M.; Faramarzi, M.; Yusuf, Z. Artificial neural network approach for predicting the water turbidity level using optical tomography. *Arab. J. Sci. Eng.* **2016**, *41*, 3369–3379. [CrossRef]
29. Vallejos, J.A.; McKinnon, S.D. Logistic regression and neural network classification of seismic records. *Int. J. Rock Mech. Min. Sci.* **2013**, *62*, 86–95. [CrossRef]
30. Kasihmuddin, M.S.M.; Mansor, M.A.; Basir, M.F.M.; Jamaludin, S.Z.M.; Sathasivam, S. The Effect of logical Permutation in 2 Satisfiability Reverse Analysis Method. In Proceedings of the 27th National Symposium on Mathematical Sciences (SKSM27), Bangi, Malaysia, 26–27 November 2019.
31. Zamri, N.E.; Mansor, M.; Mohd Kasihmuddin, M.S.; Alway, A.; Mohd Jamaludin, S.Z.; Alzaeemi, S.A. Amazon employees resources access data extraction via clonal selection algorithm and logic mining approach. *Entropy* **2020**, *22*, 596. [CrossRef]
32. Karp, R.M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*; Raymond, E.M., James, W.T., Jean, D.B., Eds.; Springer: Boston, MA, USA, 1972; pp. 85–103.
33. Hopfield, J.J.; Tank, D.W. "Neural" computation of decisions in optimization problems. *Biol. Cybern.* **1985**, *52*, 141–152. [CrossRef] [PubMed]
34. Sejnowski, T.J.; Tesauro, G. The Hebb rule for synaptic plasticity: Algorithms and implementations. In *Neural Models of Plasticity*; Academic Press: Cambridge, MA, USA, 1989; pp. 94–103.
35. Jha, K.; Saha, S. Incorporation of multimodal multiobjective optimization in designing a filter based feature selection technique. *Appl. Soft Comput.* **2021**, *98*, 106823. [CrossRef]
36. Singh, N.; Singh, P. A hybrid ensemble-filter wrapper feature selection approach for medical data classification. *Chemom. Intell. Lab. Syst.* **2021**, *217*, 104396. [CrossRef]
37. Zhu, Q. On the performance of Matthews correlation coefficient (MCC) for imbalanced dataset. *Pattern Recognit. Lett.* **2020**, *136*, 71–80. [CrossRef]
38. Mohd Kasihmuddin, M.S.; Mansor, M.; Md Basir, M.F.; Sathasivam, S. Discrete mutation Hopfield neural network in propositional satisfiability. *Mathematics* **2019**, *7*, 1133. [CrossRef]
39. Dietterich, T.G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.* **1998**, *10*, 1895–1923. [CrossRef] [PubMed]
40. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
41. Bazuhair, M.M.; Jamaludin, S.Z.M.; Zamri, N.E.; Kasihmuddin, M.S.M.; Mansor, M.; Alway, A.; Karim, S.A. Novel Hopfield neural network model with election algorithm for random 3 satisfiability. *Processes* **2021**, *9*, 1292. [CrossRef]
42. Bonet, M.L.; Buss, S.; Ignatiev, A.; Morgado, A.; Marques-Silva, J. Propositional proof systems based on maximum satisfiability. *Artif. Intell.* **2021**, *300*, 103552. [CrossRef]
43. Li, C.M.; Zhu, Z.; Manyà, F.; Simon, L. Optimizing with minimum satisfiability. *Artif. Intell.* **2012**, *190*, 32–44. [CrossRef]

44. Alzaeemi, S.A.S.; Sathasivam, S. Examining the forecasting movement of palm oil price using RBFNN-2SATRA Metaheuristic algorithms for logic mining. *IEEE Access* **2021**, *9*, 22542–22557. [CrossRef]
45. Sathasivam, S.; Mamat, M.; Kasihmuddin, M.S.M.; Mansor, M.A. Metaheuristics approach for maximum k satisfiability in restricted neural symbolic integration. *Pertanika J. Sci. Technol.* **2020**, *28*, 545–564.