SEMARAK ILMU PUBLISHING
202103268166(003316878-P)

Semarak International Journal of Fundamental and Applied Mathematics

Journal homepage:
https://semarakilmu.my/index.php/sijfam/index
ISSN: 3030-5527

Dissecting Bank Decision Criteria via Supervised 2-Satisfiability Reverse Analysis

Ameer Azamuddin Abdul Ghafar[1], Nur Ezlin Zamri[1,*] , Nurin Kamalin Mohd Fadil[1], Farisya Husna Mansor[1], Nurul Huda Ahmad Rusli[1], Nurin Hazwani Hamidi[1], Yueling Guo[2]

[1] Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, Serdang, Selangor 43400 UPM, Malaysia
[2] School of Science, Human Institute of Technology, 421002 Hengyang, China

ARTICLE INFO

Article history:
Received 15 July 2025
Received in revised form 14 August 2025
Accepted 23 August 2025
Available online 27 August 2025

Keywords:
Bank loan approval; Supervised 2-Satisfiability Reverse Analysis; Hopfield Neural Network; Correlation analysis; K-Means clustering; logical rule induction; performance metrics

ABSTRACT

Bank loan approval is a critical financial process that demands both accuracy and transparency to ensure fairness, regulatory compliance, and customer trust. However, many machine learning models act as "black boxes," offering strong predictive power but limited interpretability, which restricts their adoption in banking. This study aims to develop an interpretable and transparent loan approval model using the Supervised 2-Satisfiability Reverse Analysis (S2SATRA) framework. The method integrates Hopfield Neural Networks with 2-Satisfiability logic clauses, enhanced by correlation-based feature selection and K-Means clustering for binary encoding. Model performance was evaluated on synthetic and real-world datasets using structured train-test splits and standard metrics, including accuracy, precision, recall (sensitivity), specificity, and F1 score. Results showed that the enhanced S2SATRA achieved up to 77.5% accuracy, 98.11% precision, and 78.62% recall, though specificity remained lower at 30.47%. These findings highlight the balance between interpretability and predictive capability, offering a logic-driven alternative to black-box models and supporting accountable decision-making in banking.

1. Introduction

In today's competitive and data-driven financial environment, the ability to make accurate and fair loan approval decisions are more critical than ever. Lending institutions play a pivotal role in the economic ecosystem by allocating financial resources to individuals and businesses. A key challenge in this process is determining creditworthiness—deciding whether an applicant is likely to repay the loan. Traditionally, this task has relied on credit scores, income verification, and human judgment. However, due to increasing data availability and complexity, these conventional methods are often insufficient, prone to bias, or too slow. To overcome these limitations, machine learning (ML) has emerged as a powerful alternative. ML models are capable of learning complex relationships from historical data and automating the loan approval process with impressive accuracy. Recent research

* *Corresponding author.*
E-mail address: ezlinzamri@upm.edu.my

https://doi.org/10.37934/sijfam.7.1.3950

has demonstrated the success of models such as Decision Trees, Random Forests, and AdaBoost in classifying loan outcomes. For instance, the work by Haque and Hassan [3] applied multiple ML classifiers to a dataset of over 148,000 loan applications and reported that AdaBoosting achieved nearly perfect accuracy (99.99%), demonstrating the strength of ensemble learning in financial prediction tasks.

Despite the impressive accuracy reported in prior studies, a significant limitation per- sists: many machine learning models function as ”black boxes.” These models can predict outcomes effectively but provide little to no explanation for the decisions produced. This lack of interpretability is problematic in real-world banking applications where fairness, accountability, and explainability are essential. Regulatory frameworks require banks to justify lending decisions, particularly in rejection cases, and customers also demand transparency regarding how their financial data is processed. To address this issue, the research community has shifted focus toward interpretable machine learning and logic-based models. One promising direction is the integration of symbolic logic with neural networks—a paradigm referred to as neuro-symbolic AI. A notable contribution in this domain is the Supervised 2-Satisfiability Reverse Analysis (S2SATRA) framework, introduced by Kasihmuddin *et al.,* [4]. The S2SATRA model aims to bridge the gap between rule-based logic systems and neural computation. It employs 2-Satisfiability (2SAT) clauses to represent decision rules and embeds them into a Hopfield Neural Network (HNN), which functions as an associative memory for logic retrieval.

The S2SATRA model operates through three phases: (1) encoding logical clauses based on attribute pairs, (2) embedding those clauses into a neural network through a cost function, and (3) minimizing energy to identify the most stable logic configuration. For instance, the framework can store rules such as “IF income is high AND credit score is good THEN approve loan,” enabling traceable and human-understandable decisions. Prior applications of S2SATRA have demonstrated its potential across multiple domains, including medical diagnosis, physics, and social systems. However, when applying the original S2SATRA model to real-world financial datasets, several challenges arise. Bank loan datasets typically include a mix of continuous and categorical variables, high feature dimensionality, imbalanced classes, and temporal dependencies. The original framework’s reliance on random clause construction may result in suboptimal logic generation and poor generalizability. Furthermore, a lack of proper feature selection in earlier approaches can allow noise and redundancy to diminish both interpretability and performance. To address these limitations, this study introduces several enhancements to the original S2SATRA model. First, correlation analysis is used to identify features that are most relevant to loan outcomes, reducing dimensionality and improving logic quality. Second, K-Means clustering is applied to discretize and normalize continuous variables, making them suitable for SAT-based logic encoding. Third, a structured train-test split strategy is adopted to assess the generalizability of extracted rules on unseen data.

Despite the promising potential of neuro-symbolic approaches such as S2SATRA, current applications to financial datasets face several challenges, including handling mixed variable types, feature redundancy, and limited interpretability when random clause construction is applied. These gaps highlight the need for a more robust and transparent framework that can generate meaningful logical rules while maintaining strong predictive performance. Therefore, this study sets out to enhance the original S2SATRA model by integrating correlation-based feature selection, K-Means clustering for discretization, and structured train-test evaluation. The objective is to develop an interpretable loan approval model that produces human-readable decision rules without sacrificing predictive accuracy. The significance of this research lies in its ability to bridge the gap between performance and transparency, enabling banks to make fairer, accountable, and regulation-compliant lending decisions. Furthermore, the enhanced framework offers broader applicability to other domains where explainability and trust are essential in decision-making systems.

2. Methodology

This section presents the theoretical background for each component involved in the proposed logic mining approach. It begins with a general overview of 2SAT based on previous research. Following that, it provides a detailed explanation of the DHNN, including a review of existing DHNN-SAT models. Finally, the section explores the concepts behind each core component and outlines the four-phase structure of the logic mining model implemented within the DHNN framework.

2.1 Formulation of kSAT

The k-Satisfiability (kSAT) problem is a Boolean logic formulation where each clause contains exactly *k* literals, with each literal being a variable or its negation. A kSAT formula consists of a conjunction (AND) of such clauses, and each clause is a disjunction (OR) of *k* literals. Formally, the problem can be expressed as:

$$Q_{kSAT} = \wedge_{\{i=1\}}^{y} c_i \quad (1)$$

Here, each C_i is a clause with *k* literals and *y* is the total number of clauses. The aim is to determine whether there exists an assignment of Boolean values that satisfies all clauses simultaneously. The complexity of the problem increases with *k* which is 2SAT is solvable in polynomial time, while 3SAT and above are NP-complete. This structure provides a unified framework for logic modelling in AI, particularly in rule induction and symbolic reasoning.

2.2 Process of kSAT in Discrete Hopfield Neural Network

The DHNN is a recurrent neural network model characterized by binary-valued neurons and symmetric synaptic connections. In this research, Abdeen *et al.,* [1] state that DHNN is employed as a memory system to store satisfiability based on logic clauses and perform logical inference via an energy minimization process.

Each neuron s_i in the DHNN represents a literal from the binary attribute space and takes on a bipolar state from the set $\{-1, +1\}$, where $+1$ represent logical TRUE and -1 represent logical FALSE. The network operates based on the dynamics of energy minimization, where synaptic weight w_{ij} are symmetric ($w_{ij} = w_{ji}$) and no self-connections exist ($w_{ii} = 0$).

Energy Function:
The DHNN minimizes an energy function E, which is defined as:

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\, s_i s_j + \sum_{i=1}^{n} \theta_i s_i \quad (2)$$

where:

- S_i is the current state of neuron i,
- w_{ij} is the synaptic weight between neurons i and j,
- Θ_i is the threshold for neuron i,
- n is the total number of neurons in the network.

The energy function guarantees that as the network updates, the energy E decreases until a stable state is reached. This stable state represents a local minimum in the energy landscape, corresponding to a logic rule consistent with the input pattern.

Cost Function:
The cost function is derived from the number of unsatisfied clauses in the satisfiability hypothesis. Suppose the logic hypothesis H is composed of k clauses, where each clause C_k is the form $(p_k\ V\ q_k)$, with literals $p_{k,}q_k\ \in\ \{x, \neg x\}$.

The cost function is defined:

$$COST = \sum_{i=1}^{NC} \prod_{j=1}^{2} M_{ij} \tag{3}$$

where NC is the total number of clauses. The definition of clauses M_{ij} is given as follow:

$$M_{ij} = \begin{cases} \frac{1}{2}(1 - S_y), & \text{if } \neg y \\ \frac{1}{2}(1 + S_y), & \text{otherwise.} \end{cases} \tag{4}$$

Neuron Update:
Neuron states are updated using the following asynchronous rule:

$$s_i(t+1) = \begin{cases} +1, & \text{if } \sum_{j=1}^{n} w_{ij} s_j(t) > \theta_i \\ -1, & \text{otherwise.} \end{cases} \tag{5}$$

Clause logic is embedded into the DHNN weight matrix by constructing synaptic weight. Clauses such as $(x_i\ V\ \neg x_j)$ are encoded so that violations raise energy. The weight matrix is constructed based on correlation guided to reducing clauses noise and improving logic integrity.

Hyperbolic Tangent Activation Function (HTAF) is employed to ensure smoother and continuous state transitions during training. HTAF is defined as:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{6}$$

This function squashes neuron inputs into range $(-1,1)$, helping the network converge more gradually toward stable energy states and reducing oscillations during updates. The HTAF enhances the learning stability of the DHNN while still preserving its binary decision behaviour after binary decision mapping.

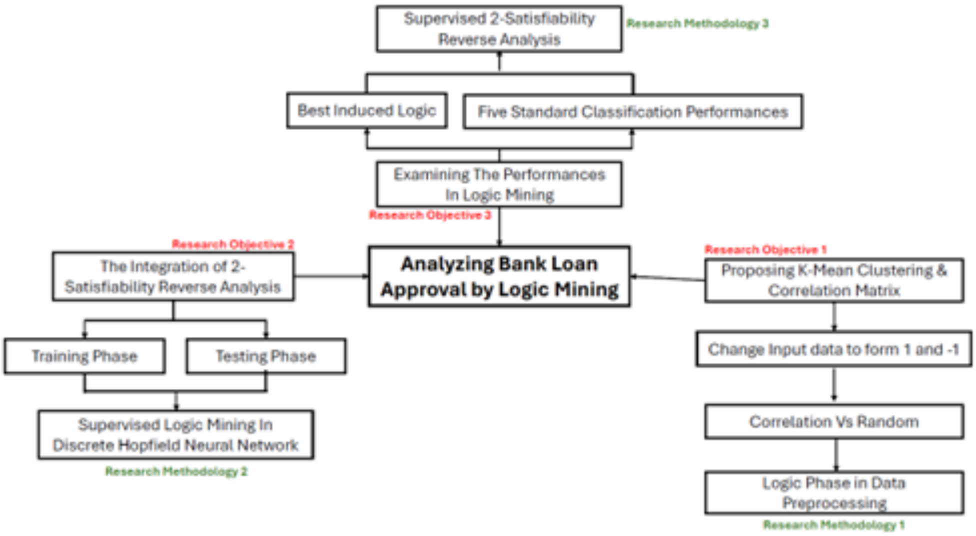

Fig. 1. Brief overview of methodology involved in S2SATRA

2.3 Modified kSAT into 2SATRA

In this subsection, the reformulate k-SAT into 2SATRA (Two-Satisfiability with Reverse Analysis), reducing clauses to two literals and applying reverse analysis to efficiently trace constraint dependencies and solution feasibility.

2.3.1 Dataset information

The dataset used in this research is obtained from Kaggle: Bank Loan Data by Uday Malviya. It contains 16,384 records of loan applications with a range of features grouped into personal information, loan details, and credit history:

Table 1
Datasets features overview

Category	Variable	Description
Personal Information	• person_age • person_gender • person_education • person_income • person_emp_exp • person_home_ownership	• Applicant's age (in years) Gender (male, female) • Education level (High School, Bachelor, Master, etc.) • Annual income (USD) • Employment experience (years) • Home ownership status (RENT, OWN, MORTGAGE)
Loan Details	• loan_amnt • loan_intent • loan_int_rate • loan_percent_income	• Requested loan amount (USD) • Loan purpose (PERSONAL, EDUCATION, MEDICAL,etc.) • Interest rate on the loan • Ratio of loan amount to annual income

Table 1 (*Continued*)

Credit & Loan History	● cb_person_cred_hist_length ● credit_score ● previous_loan_defaults_on_file	● Length of the applicant's credit history (years) ● Applicant's credit score ● Whether the applicant has defaulted on previous loans (Yes/No)
Target Variable	● loan_status	● Indicates if the loan was ap- proved (1) or not (0)

2.3.2 Data preprocessing phase

To ensure the reliability and interpretability of the logic mining process, a comprehensive data preprocessing phase was conducted. The dataset contained various financial and personal attributes of individuals applying for bank loans. The preprocessing involved a multi-stage pipeline utilizing SPSS for clustering and Excel for statistical correlation analysis.

1. Clustering with K-Means: Initially, the raw data underwent clustering using the k-means algorithm in Statistical Package for the Social Sciences (SPSS). This step grouped similar records together based on underlying patterns in the variables. The rationale for clustering was to reduce noise and ensure homogeneity within the data subsets that would be subjected to logic mining.
2. Binary Transformation: Following clustering, all relevant attributes were converted into a binary representation, using values of 1 and -1. This transformation was essential to facilitate the application of the 2 Satisfiability (2SAT) logic mining algorithm, which operates efficiently on binary data structures.
3. Correlation Matrix Construction: Using Microsoft Excel, a correlation matrix was created to evaluate the linear relationships between the attributes. This matrix helped identify which variables had strong influence on the target variable (loan approval) as shown in Figure 1.
4. Variable Selection Based on Correlation: Six variables that exhibited the highest positive correlation (closest to +1) with loan approval were selected. This step ensured that only the most relevant and informative features were retained for analysis, thereby improving the potential predictive power and interpretability of the resulting logical rules.
5. Preparation of Logical Dataset: The final dataset consisting of these six selected variables was then exported and formatted as a new Excel sheet. This structured data served as the input for the logic mining phase, which was carried out using C++ implementations in Dev C++ (Orwell version) and Dev C++ (Embarcadero version) environments.

This meticulous preprocessing was crucial in refining the dataset to a form that is compatible with and optimized for logic mining using the 2SAT reverse analysis approach.

	person_gender	person_education	person_income	person_emp_exp	person_home_ownership	loan_amount	loan_intent	loan_int_rate	loan_percent_income	cb_person_cred_hist_length	credit_score	previous_loan_defaults_on_tile	loan_status
person_gender	1	-0.000599	0.011000	-0.000256	0.001802	0.008110	0.004072	0.002694	-0.001408	0.000996	-0.002590	-0.000704	0.000561
person_education	0.105904	1	0.005483	0.048000	0.010000	0.007999	0.008340	-0.002486	0.001754	0.054000	0.059000	0.010000	0.005497
person_income	0.079452	0.102611	1	0.082000	-0.060000	0.275000	0.058000	-0.014000	-0.234000	0.077000	0.014000	-0.094000	0.193000
person_emp_exp	0.149799	0.043895	0.031660	1	0.000307	0.028000	0.035000	0.011000	-0.033000	0.681000	0.108000	0.019000	0.019000
person_home_ownership	0.081531	0.080173	0.214671	0.153370	1	-0.027000	0.006519	-0.006257	0.042000	-0.003776	-0.006090	-0.053000	0.094000
loan_amount	0.129771	0.155741	0.225966	0.178859	0.204328	1	0.030000	0.069000	0.475000	0.032000	0.003092	0.039000	-0.081000
loan_intent	0.090824	0.096766	0.011071	0.075363	0.090517	0.095307	1	0.005623	-0.013000	0.036000	0.005319	0.007219	-0.017000
loan_int_rate	0.082442	0.108099	0.186065	0.127360	0.124638	0.046024	0.083405	1	0.087000	0.011000	0.008680	0.128000	-0.230000
loan_percent_income	0.086761	0.099340	0.453829	0.220566	0.033848	0.645902	0.117888	0.168812	1	-0.023000	-0.008825	0.139000	-0.259000
cb_person_cred_hist_length	0.148298	0.036687	0.025226	0.913662	0.157697	0.171085	0.074400	0.126708	0.207861	1	0.100000	0.014000	0.018000
credit_score	0.130256	0.003244	0.113412	0.069600	0.143299	0.191929	0.120396	0.085216	0.125360	0.058943	1	0.151000	0.001458
previous_loan_defaults_on_tile	0.060934	0.045067	0.330022	0.065005	0.189543	0.048777	0.048837	0.290086	0.303010	0.069002	0.197969	1	-0.543000
loan status	0.011238	0.008591	0.364092	0.021585	0.155259	0.241412	0.037537	0.478228	0.521593	0.019436	0.083202	0.856674	1

Fig. 2. Correlation matrix table

2.3.3 Training phase

The training phase is a critical step in the S2SATRA framework, where logical rules are constructed and encoded into the neural memory structure. After the data preprocessing phase, the cleaned and discretized dataset containing the most relevant features is prepared for training. This dataset is fed into the S2SATRA training pipeline, which is implemented in Dev C++ Orwell. In this stage, each data sample (record) is analysed to construct 2-Satisfiability (2SAT) clauses. Each clause is formed by selecting two attributes (variables) that are positively correlated with the target outcome (loan approval or rejection). These pairs are then converted into Boolean logic clauses, such as:

$$C_i = (m_i \vee n_i) \tag{7}$$

These clauses are embedded into a Discrete Hopfield Neural Network (DHNN) by calculating their synaptic weights using the energy function. This allows the network to represent logical knowledge in a distributed memory format. The cost function ensures that the system favors logical configurations that satisfy most training samples. During this process, clauses that appear most frequently among positive loan outcomes are prioritized for storage. The resulting model effectively encodes decision rules such as:

”IF income is high AND credit score is good THEN approve loan.”

The implementation in Dev C++ enables direct control over clause generation, memory updates, and logic verification through custom-coded logic mining scripts. Throughout training, output files containing the clause matrices and synaptic weight configurations are generated for later retrieval and testing.

2.3.3 Retrieval phase

Once the training phase is complete and the most representative logical clauses have been stored in the neural network memory, the model enters the retrieval phase. This step involves using the trained clause structure to test its ability to predict loan decisions on unseen data. The retrieval phase is carried out using a separate executable coded in Dev C++ Red (Embarcadero version), which supports more efficient memory management and I/O operations compared to standard Dev C++. The retrieval process simulates the recall function of the Hopfield Neural Network. Each test input is passed through the clause memory, and the network iteratively updates neuron states to minimize energy according to the predefined energy function as of Equation (1). By reaching the minimum energy state, the system converges to a stable logical decision that either accepts or rejects the loan application. This logical decision is compared against the actual target label to determine correctness. Additionally, the retrieval phase allows the extraction of the most stable and frequently retrieved rules using a clause frequency analysis as of Eq. (4). The system identifies which logical clause configurations are most effective in correctly classifying loan applications. The logic retrieval is followed by scoring the model using standard classification metrics, which are computed and recorded for each test batch. The use of Dev C++ Red in this phase ensures better debugging, memory tracing, and modular logic evaluation, making it well-suited for high-volume rule-based simulations in financial datasets.

3. Experimental Setup

The implementation of this research involves multiple software tools:
• Microsoft Excel – for initial data preparation and cleaning.
• IBM SPSS – for statistical analysis and correlation matrix construction.
• Dev C++ – for implementing S2SATRA training with correlation-guided rule formation.
• Dev C++ Embarcado – for executing test phase logic evaluation and performance scoring.

To evaluate the effectiveness of the 2-Satisfiability logic mining process, five standard classification performance metrics were used: accuracy, precision, recall(sensitivity), specificity, and F1 score. The formulas for each metric are formulated in Eq. (8) - Eq. (12) as follow:

- Accuracy= $(TP + TN) / (TP + TN + FP + FN)$ ` (8)
- Precision= $TP/(TP + FP)$ (9)
- Recall (Sensitivity)= $TP/(TP + FN)$ ` (10)
- Specificity= $TN/(TN + FP)$ (11)
- F1 Score= $2\,X(Precision\ x\ Sensitivity) / (Precision + Sensitivity)$ (12)

4. Results and Discussion

This section presents the outcomes of the proposed approach and examines their implications. The results are analysed in the context of the research objectives, with a focus on performance, accuracy, and the insights revealed through comparative evaluation.

4.1 Metrices Performances Table

Table 2
Performance metrics for each configuration (A max - average across all folds)

Train:Test Split	Accuracy	Precision	Sensitivity	Specificity	F1 Score
60:40	0.7753	0.9811	0.7862	0.2078	0.8725
70:30	0.7737	0.9793	0.7854	0.2238	0.8709
80:20	0.7296	0.9803	0.7391	0.2528	0.8403
90:10	0.7449	0.8927	0.7657	0.3047	0.8194

Table 3
Performance metrics for each configuration (A standard - average across all folds)

Train:Test Split	Accuracy	Precision	Sensitivity	Specificity	F1 Score
60:40	0.7720	0.9758	0.7859	0.1990	0.8703
70:30	0.7713	0.9433	0.8179	0.1784	0.8695
80:20	0.7564	0.9386	0.8064	0.1951	0.8589
90:10	0.7277	0.9075	0.8072	0.1958	0.8369

Table 4
Performance metrics for each configuration (B max - average across all folds)

Train:Test Split	Accuracy	Precision	Sensitivity	Specificity	F1 Score
60:40	0.6643	0.6050	0.3462	0.8595	0.4403
70:30	0.6595	0.56042	0.4014	0.8192	0.4430
80:20	0.6763	0.6310	0.3647	0.8676	0.4621
90:10	0.6600	0.6156	0.3533	0.8551	0.4487

Table 5
Performance metrics for each configuration (B standard - average across all folds)

Train:Test Split	Accuracy	Precision	Sensitivity	Specificity	F1 Score
60:40	0.3138	0.5498	0.1715	0.6494	0.25857
70:30	0.3381	0.3241	0.3708	0.4579	0.2570
80:20	0.3422	0.5250	0.1751	0.6832	0.2599
90:10	0.3212	0.5508	0.1781	0.6463	0.2666

Explanation of Model Variants:

- A Max – Logical rules generated using attribute selection based on correlation analysis, combined with permutation-based enhancement to maximize clause relevance.
- A Standard – Logical rules generated using only correlation analysis for attribute selection, without applying permutation.
- B Max – Logical rules generated using randomly chosen attributes but enhanced through permutation to identify the best clause combinations.
- B Standard – Logical rules generated from randomly selected attributes without any permutation or optimization.

4.2 Discussion on Metrices Performances Tables

The experimental results demonstrate a clear distinction in performance across the four evaluated configurations: A Max, A Standard, B Max, and B Standard. These configurations differ based on their attribute selection strategy and whether permutation was applied, significantly impacting their ability to extract meaningful and precise logical rules for predicting loan approvals. Among all variants, the A Max model achieved the highest performance across key evaluation metrics, particularly F1 Score and precision. This model employed a combination of correlation-based attribute selection and permutation, which together helped to refine the input features and optimize the logical structure derived from the 2-Satisfiability Re- verse Analysis. The high F1 Score indicates that the model effectively balances precision and recall, minimizing both false positives and false negatives. The increased precision specifically suggests that the logical rules formed are not only accurate but also reliable in identifying true loan approvals without misclassifying non-eligible cases. When comparing A Max to B Max, the performance gap is noticeable. While B Max also utilizes permutation, its use of randomly selected attributes (without correlation analysis) results in a lower F1 Score and reduced precision. This implies that although permutation helps fine-tune the model, its effectiveness is limited when the foundational attributes lack relevance. In other words, the quality of the input features is a critical factor—models built on randomly selected variables cannot perform well consistently, even if optimized through permutation. The comparison between A Standard and B Standard further reinforces this point. A Standard, which relies solely on correlation-based feature selection without permutation, still outperforms B Standard in all considered metrics.

This finding underscores the value of selecting meaningful features based on correlation analysis, even when no additional optimization (such as permutation) is applied. In contrast, B Standard, which uses both random feature selection and no permutation, shows the weakest performance, with lower precision and F1 Score. This confirms that both attribute relevance and optimization are essential for logical rule mining to be effective. Overall, models within the A group (A Max and A Standard) significantly outperform those in the B group (B Max and B Standard). The results clearly highlight the importance of integrating correlation-based feature selection in the modeling process. While permutation helps improve model performance further, it cannot fully compensate for poor feature selection. The A group's superior F1 Scores and precision values demonstrate that correlation is a reliable method for identifying influential attributes and forming logi- cal rules that generalize well. Therefore, combining statistical correlation techniques with logical rule induction methods results in more accurate, interpretable, and robust models for bank loan approval prediction.

4.3 Best Induced Logic

The best induced logic derived from the enhanced S2SATRA framework reflects a clear and interpretable decision-making pattern. This rule, selected through correlation-guided feature selection and optimized clause induction, aligns closely with established banking practices, reinforcing the model's validity. Its simplicity enhances transparency, enabling stakeholders to understand and justify approval decisions while maintaining strong predictive performance.

Best Induced Logic:

(FV_B) ^ (EV_D) ^ (CV_A)
(FV_B) ^ (EV_C) ^ (DV_A)

from Train:Test Split (60:40) Fold 3
Explanation of Variables:
A - *person_income*
B - *person_home_ownership*
C - *loan_amount*
D - *loan_int_rate*
E - *loan_percent_income*
F - *previous_loan_defaults_on_tile*

4.4 Discussion on Best Induced Logic

The best induced logic expressions provide meaningful insight into predicting the loan status, which is the target variable indicating whether a loan is approved or results in default. The presence of previous loan defaults (F) combined with home ownership status (B) as a common factor in both expressions suggests that these borrower characteristics strongly influence loan outcomes. Borrowers with a history of defaults are more likely to default again, making this a critical predictor of loan status. Meanwhile, home ownership typically signals greater financial stability, increasing the likelihood of loan approval. Moreover, the interaction between loan characteristics such as loan amount (C), loan interest rate (D), and the percentage of income spent on the loan (E) with borrower income (A) reflects the borrower's capacity to repay. For example, a high loan amount or interest rate may lead to default if the borrower's income is insufficient to handle the repayment burden. Conversely, when the borrower's income is adequate relative to these loan parameters, loan approval becomes more probable. The two induced logic expressions capture different but complementary scenarios leading to loan approval or default. One emphasizes the role of loan interest rate with loan percent income, and the other focuses on loan amount with loan percent income, both combined with borrower income and previous default/home ownership status. These expressions effectively classify loan applications into “approved” or “default” categories based on the interplay of financial risk factors. In essence, the induced logic links the borrower’s financial history and current loan terms directly to the likelihood of loan approval or default. This logical framework not only supports accurate predictions of loan status but also provides transparency in decision-making by highlighting the critical factors leading to approval or default outcomes.

5. Conclusions

This research has presented a refined approach to bank loan approval prediction by combining logic-based reasoning with supervised machine learning, using the Supervised 2-Satisfiability Reverse Analysis (S2SATRA) framework. The study addressed key challenges associated with traditional ”black box” machine learning models, particularly the lack of transparency, interpretability, and regulatory compliance in decision-making pro- cesses. Through the incorporation of correlation analysis, K-Means clustering, and structured train-test evaluation, the enhanced S2SATRA model was able to derive logical rules that are both explainable and statistically robust. Among the four evaluated configurations, the A Max model—featuring correlation-guided attribute selection and permutation-based clause enhancement—consistently outperformed other variants in accuracy, precision, recall, specificity, and F1 Score. These results highlight the critical role of meaningful feature selection in optimizing rule quality and predictive performance. The findings reinforce that interpretable logic models can be both effective and reliable when coupled with rigorous data preprocessing and validation strategies. By providing clear, human-readable rules for loan approvals, the proposed framework supports fairer and more transparent decision-making in banking applications. Moreover, it sets the foundation for integrating logic-based AI methods in other domains where explainability and accountability are paramount. Future work can focus on extending this methodology to handle multi-class loan outcomes, temporal loan repayment behaviors, or integrating fuzzy logic for handling uncertainties. Additionally, deploying this model in real-time banking systems could provide further insights into its practical effectiveness and scalability.

Acknowledgement
This research is financially supported by Universiti Putra Malaysia for the IPM Putra Grant with Project Code: GP-IPM/2024/9806600. All authors gratefully acknowledge the support from Universiti Putra Malaysia. The authors would like to express their sincere appreciation to all individuals and institutions who contributed to this research. Special thanks are extended to colleagues, reviewers, and collaborators for their constructive feedback and valuable assistance throughout the study.

References

[1] Abdeen, Suad, Mohd Shareduwan Mohd Kasihmuddin, Nur Ezlin Zamri, Gaeithry Manoharam, Mohd Asyraf Mansor, and Nada Alshehri. "S-type random k satisfiability logic in discrete hopfield neural network using probability distribution: Performance optimization and analysis." *Mathematics* 11, no. 4 (2023): 984. https://doi.org/10.3390/math11040984

[2] Alagic, Adnan, Natasa Zivic, Esad Kadusic, Dzenan Hamzic, Narcisa Hadzajlic, Mejra Dizdarevic, and Elmedin Selmanovic. "Machine learning for an enhanced credit risk analysis: A comparative study of loan approval prediction models integrating mental health data." *Machine Learning and Knowledge Extraction* 6, no. 1 (2024): 53-77. https://doi.org/10.3390/make6010004

[3] Haque, F. M., and Md Mahedi Hassan. "Bank loan prediction using machine learning techniques." *arXiv*

preprint arXiv:2410.08886 (2024). https://doi.org/10.4236/ajibm.2024.1412085

[4] Kasihmuddin, Mohd Shareduwan Mohd, Siti Zulaikha Mohd Jamaludin, Mohd Asyraf Mansor, Habibah A. Wahab, and Siti Maisharah Sheikh Ghadzi. "Supervised learning perspective in logic mining." *Mathematics* 10, no. 6 (2022): 915. https://doi.org/10.3390/math10060915

[5] Romli, Nurul Atiqah, Nur Fariha Syaqina Zulkepli, Mohd Shareduwan Mohd Kasihmuddin, Nur Ezlin Zamri, Nur'Afifah Rusdi, Gaeithry Manoharam, Mohd Asyraf Mansor, Siti Zulaikha Mohd Jamaludin, and Amierah Abdul Malik. "Unsupervised logic mining with a binary clonal selection algorithm in multi-unit discrete Hopfield neural networks via weighted systematic 2 satisfiability." *AIMS Mathematics* 9, no. 8 (2024): 22321-22365. https://doi.org/10.3934/math.20241087

[6] Abdullah, Wan Ahmad Tajuddin Wan. "Logic programming on a neural network." *International journal of intelligent systems* 7, no. 6 (1992): 513-519. https://doi.org/10.1002/int.4550070604

[7] Gupta, Anshika, Vinay Pant, Sudhanshu Kumar, and Pravesh Kumar Bansal. "Bank loan prediction system using machine learning." In *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, pp. 423-426. IEEE, 2020. https://doi.org/10.1109/SMART50582.2020.9336801

[8] Anand, Mayank, Arun Velu, and Pawan Whig. "Prediction of loan behaviour with machine learning models for secure banking." *Journal of Computer Science and Engineering (JCSE)* 3, no. 1 (2022): 1-13. https://doi.org/10.36596/jcse.v3i1.237

[9] Dosalwar, Sharayu, Ketki Kinkar, Rahul Sannat, and Nitin Pise. "Analysis of loan availability using machine learning techniques." *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)* 9, no. 1 (2021): 15-20. https://doi.org/10.48175/IJARSCT-1895

[10] Alsaleem, Maan, and Safwan Hasoon. "Predicting bank loan risks using machine learning algorithms." *AL-Rafidain Journal of Computer Sciences and Mathematics* 14, no. 1 (2020): 149-158. https://doi.org/10.33899/csmj.2020.164686

[11] Manoharam, Gaeithry, Mohd Shareduwan Mohd Kasihmuddin, Siti Noor Farwina Mohamad Anwar Antony, Nurul Atiqah Romli, Nur 'Afifah Rusdi, Suad Abdeen, and Mohd Asyraf Mansor. "Log-linear-based logic mining with multi-discrete hopfield neural network." *Mathematics* 11, no. 9 (2023): 2121. https://doi.org/10.3390/math11092121

[12] Karim, Syed Anayet, Nur Ezlin Zamri, Alyaa Alway, Mohd Shareduwan Mohd Kasihmuddin, Ahmad Izani Md Ismail, Mohd Asyraf Mansor, and Nik Fathihah Abu Hassan. "Random satisfiability: A higher-order logical approach in discrete Hopfield Neural Network." *IEEE Access* 9 (2021): 50831-50845. https://doi.org/10.1109/ACCESS.2021.3068998

[13] Jamaludin, Siti Zulaikha Mohd, Mohd Asyraf Mansor, Aslina Baharum, Mohd Shareduwan Mohd Kasihmuddin, Habibah A. Wahab, and Muhammad Fadhil Marsani. "Modified 2 satisfiability reverse analysis method via logical permutation operator." *Computers, Materials & Continua* 74, no. 2 (2023). https://doi.org/10.32604/cmc.2023.032654

[14] Zamri, Nur Ezlin, Mohd Asyraf Mansor, Mohd Shareduwan Mohd Kasihmuddin, Siti Syatirah Sidik, Alyaa Alway, Nurul Atiqah Romli, Yueling Guo, and Siti Zulaikha Mohd Jamaludin. "A modified reverse-based analysis logic mining model with weighted random 2 satisfiability logic in discrete hopfield neural network and multi-objective training of modified niched genetic algorithm." *Expert Systems with Applications* 240 (2024): 122307. https://doi.org/10.1016/j.eswa.2023.122307

[15] Sathasivam, Saratha, and Wan Ahmad Tajuddin Wan Abdullah. "Logic mining in neural network: Reverse analysis method." *Computing* 91, no. 2 (2011): 119-133. https://doi.org/10.1007/s00607-010-0117-9