Machine Learning for Credit Approval: Enhancing Decision Accuracy and Explainability

Mohamed Azhar Ahamed
Wellington Institute of Technology
Wellington, New Zealand
zahvie@gmail.com

Abubakar Siddique
Auckland University of Technology
Auckland, New Zealand
abubakar.siddique@aut.ac.nz

Trung Nguyen
Auckland University of Technology
Auckland, New Zealand
trung.nguyen@aut.ac.nz

Muhammad Iqbal
Higher Colleges of Technology
Fujairah, United Arab Emirates
miqbal1@hct.ac.ae

Will N. Browne
Queensland University of Technology
Brisbane, Australia
will.browne@qut.edu.au

Abstract

Machine learning is widely used to improve predictive accuracy in complex domains like credit scoring, but many models (e.g., deep neural networks) remain opaque. This lack of interpretability is problematic in regulated domains (banking, finance) where transparency is required. Rule-based learning methods, such as Learning Classifier Systems (LCS), offer a trade-off between accuracy and explainability. We introduce a novel Ranked Attribute Selection with Midpoint Filtering (RASF) framework that extends LCS (EXTRACS) to enhance feature selection and rule validation for credit approval. RASF first ranks features by mutual information, then employs rank-guided randomized selection to diversify rule conditions, and finally filters new rules by midpoint-based Euclidean distance to the current instance. We evaluate RASF-enhanced LCS on public loan approval datasets, comparing against a baseline LCS and logistic regression. Results show that RASF improves predictive accuracy by about $3.6 - 4.64\%$ over the base LCS, while producing an inherently interpretable rule set. By bridging accuracy and transparency, RASF-LCS supports explainable AI in credit scoring.

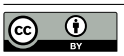
CCS Concepts

• **Computing methodologies** → **Rule learning**; *Supervised learning by classification.*

Keywords

Credit approval, learning classifier systems, interpretability, feature selection, mutual information, explainable AI

ACM Reference Format:
Mohamed Azhar Ahamed, Abubakar Siddique, Trung Nguyen, Muhammad Iqbal, and Will N. Browne. 2026. Machine Learning for Credit Approval: Enhancing Decision Accuracy and Explainability. In *Genetic and Evolutionary Computation Conference (GECCO Companion '26), July 13–17, 2026, San Jose, Costa Rica.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3795101.3814684



GECCO Companion '26, San Jose, Costa Rica

ACM ISBN 979-8-4007-2488-6/2026/07
https://doi.org/10.1145/3795101.3814684

1 Introduction

Credit scoring is a cornerstone of modern banking, enabling lenders to quantify borrower risk and make informed decisions about loan approvals, interest rates, and credit limits. Accurate creditworthiness assessments directly impact financial stability by reducing non-performing loans, which have risen globally due to macroeconomic pressures following the COVID-19 pandemic and subsequent inflationary cycles [8]. Beyond institutional risk management, credit scoring affects millions of individuals: overly conservative models deny credit to qualified applicants, while overly permissive models expose lenders to defaults and borrowers to unsustainable debt. Optimising credit assessments can therefore reduce default losses while expanding credit access for underserved populations.

Traditional credit scoring relies on static Fair Isaac Corporation (FICO)-style scores, manual reviews of financial statements, and structured interviews—processes that are time-consuming, opaque, and prone to human bias. These methods typically employ simple linear models (e.g., logistic regression, discriminant analysis) that, while transparent, struggle to capture non-linear interactions among borrower attributes such as income volatility, employment history, and debt-to-income dynamics [1]. The result is a trade-off: conventional models are interpretable but may yield suboptimal accuracy, leading to either unwarranted rejections of creditworthy applicants or approvals of high-risk borrowers.

The advent of machine learning (ML) promises to alleviate these shortcomings. ML models can ingest large volumes of heterogeneous applicant data, identify latent patterns, and update credit scores dynamically as new information becomes available. Surveys confirm that ML techniques—including neural networks, gradient boosting, and ensemble methods—are rapidly gaining traction in business and finance, with documented improvements in predictive accuracy over traditional statistical models [3, 8, 9]. For instance, deep learning architectures have achieved accuracy gains of $5 - 10\%$ on benchmark credit datasets compared to logistic regression baselines [11].

However, complex ML models often function as "black boxes" that hinder trust, auditability, and regulatory compliance [17]. In regulated domains such as banking, lenders are legally obligated to provide clear reasons for adverse credit decisions. Under U.S. law, the Equal Credit Opportunity Act (ECOA) obligates lenders to specify the main factors that led to a denial of credit [6], whereas

in the European Union, data protection rules such as the General Data Protection Regulation provide individuals with a legal entitlement to obtain an explanation when automated processing produces decisions with significant impacts on them [5, 10]. Failure to comply can result in regulatory penalties, litigation, and reputational damage [6, 10]. Furthermore, opaque models risk encoding discriminatory patterns present in historical data, potentially violating fair lending laws and perpetuating societal inequities [12]. Stakeholders—including regulators, consumer advocates, and borrowers themselves—increasingly demand that AI systems be auditable for fairness and free from algorithmic discrimination [14, 15].

Explainable AI (XAI) methods attempt to mitigate opacity by providing post-hoc explanations for black-box predictions. Techniques such as LIME (Local Interpretable Model-agnostic Explanations) [16] and SHAP (SHapley Additive exPlanations) [13] generate feature attributions that approximate a model's local behaviour [17]. While useful, these add-on explanations do not make the underlying model itself transparent; they are approximations that may be unfaithful to the true decision boundary, especially in high-dimensional or non-linear settings. Rudin [17] argues forcefully that for high-stakes decisions, inherently interpretable models should be preferred over explained black boxes. This perspective has motivated renewed interest in rule-based learners and decision trees that produce transparent decision logic by design [5].

Learning Classifier Systems (LCS) offer a compelling framework for interpretable credit scoring. An LCS is an evolutionary algorithm that learns a population of IF–THEN rules (classifiers) through genetic operators and reinforcement-style feedback [19, 22, 24]. Each rule explicitly encodes conditions on input features and a predicted class, making the model inherently human-readable. Unlike decision trees, which partition the input space hierarchically and can become unwieldy when deep, LCS rules operate as an unordered set that collectively covers the input space, allowing overlapping conditions and flexible generalisation. Recent advances have extended LCS to supervised learning tasks, with systems such as ExSTraCS [24] and XCS [4, 18, 20, 21] demonstrating competitive accuracy on biomedical and benchmark datasets [25].

Despite these strengths, standard LCS implementations exhibit limitations that hinder their adoption in credit scoring. First, conventional covering mechanisms treat all features equally, generating rules that may include irrelevant or noisy attributes. In credit risk, where domain knowledge indicates that certain features (e.g., income, prior defaults, debt-to-income ratio) are critical, this agnostic approach wastes computational resources and can produce overly general rules with poor discriminative power. Second, newly generated rules are not validated for contextual relevance: a rule may technically match an instance yet specify conditions far from the instance's actual values, reducing precision. Third, although feature selection techniques such as mutual information (MI) are well-established in ML [7], their integration into the LCS rule-generation pipeline has received limited attention.

We address these limitations by introducing a Ranked Attribute Selection with Midpoint Filtering (RASF) framework that improves the LCS covering mechanism for credit approval. RASF has three components: (i) Mutual Information–based feature ranking, which computes MI scores between each feature and the target to produce a single, training-time ranking that guides rule generation; (ii) rank-guided randomized attribute selection, which prioritizes highly informative features while maintaining controlled exploration of lower-ranked ones to preserve diversity; and (iii) midpoint-based distance filtering, which evaluates the Euclidean distance between a rule's condition midpoints and the current instance, discarding rules that are too distant to ensure locally relevant and well-centered coverage.

The contributions of this paper are as follows:

- We introduce RASF, a novel enhancement to LCS that integrates MI-based feature ranking, rank-guided attribute selection, and midpoint-based rule filtering into the covering mechanism.
- We demonstrate that RASF-LCS achieves competitive accuracy on credit approval tasks while maintaining the inherent interpretability of rule-based models.
- We provide an interpretability analysis showing that the evolved rules align with domain knowledge and can be readily understood by non-technical stakeholders.

The original RASF-LCS framework has been accepted as a poster at GECCO [2]. In this work, we provide a comprehensive exposition of the method, encompassing its theoretical motivation, algorithmic design, and empirical evaluation, to rigorously demonstrate its effectiveness and practical relevance for interpretable credit scoring.

The remainder of this paper is organised as follows. Section 2 reviews related work on credit scoring, interpretable ML, and Learning Classifier Systems. Section 3 describes the RASF methodology in detail, including pseudocode for each component. Section 4 presents the experimental setup, datasets, and baseline algorithms. Section 5 reports predictive performance and interpretability analyses. Section 6 discusses implications, limitations, and comparisons with related approaches. Finally, Section 7 concludes with a summary of contributions and directions for future work.

2 Related Work

This section presents related work across four areas: traditional credit scoring methods, machine learning approaches to credit risk, explainable AI and interpretable models, and Learning Classifier Systems. We conclude by positioning our contribution relative to the existing literature.

2.1 Traditional Credit Scoring

Credit scoring has a long history rooted in statistical methods. Logistic regression remains the industry standard due to its simplicity, interpretability, and well-understood statistical properties [1]. A logistic model estimates the probability of default as a function of applicant attributes (e.g., income, employment status, existing debt), with coefficients that indicate each feature's marginal effect on risk. Discriminant analysis, another classical approach, separates approved and rejected applicants by finding a linear boundary in feature space. These methods are transparent—a credit officer can inspect the model coefficients—and satisfy regulatory requirements for explainability.

However, traditional models have known limitations. Linear assumptions may fail to capture non-linear relationships, such as

the interaction between income and debt load, or threshold effects where risk increases sharply beyond certain values [9]. Moreover, manual feature engineering is required to incorporate domain knowledge (e.g., creating debt-to-income ratios), and model updates are infrequent, leaving scores static even as borrower circumstances change. These shortcomings motivate the exploration of more flexible machine learning techniques.

2.2 Machine Learning in Credit Scoring

The past two decades have seen extensive application of machine learning (ML) to credit risk prediction. Neural networks were among the earliest ML methods applied, with Angelini et al. [3] demonstrating improved accuracy over logistic regression on Italian bank data. Subsequent work has explored a wide range of architectures, including multilayer perceptrons, radial basis function networks, and deep learning models [8].

Ensemble methods have proven particularly effective. Random forests aggregate predictions from multiple decision trees, reducing variance and improving generalisation. Gradient boosting machines (e.g., XGBoost, LightGBM) sequentially fit weak learners to residual errors, achieving state-of-the-art performance on tabular credit data [9]. Dastile et al. [8] conducted a systematic review of 187 studies and found that ensemble methods and neural networks consistently outperformed traditional statistical models, with accuracy improvements ranging from 2% to 15% depending on the dataset.

Despite these gains, complex ML models introduce significant challenges. Deep networks and large ensembles contain thousands or millions of parameters, making it difficult to understand why a particular applicant was approved or denied. This opacity is problematic in regulated industries where decisions must be justified [17]. Gao et al. [11] surveyed ML applications in business and finance, noting that interpretability remains a barrier to adoption in credit scoring, insurance underwriting, and fraud detection. The tension between accuracy and transparency has become a central theme in the field.

2.3 Explainable AI and Interpretable Models

Explainable AI (XAI) aims to make model predictions understandable, distinguishing between *post-hoc explanations* and *inherently interpretable models* [17]. Post-hoc methods such as LIME and SHAP provide local or global feature attributions for black-box models and are widely used in credit scoring [5]. However, these explanations are approximations and may not faithfully reflect true model behaviour, particularly in complex or high-dimensional settings. For high-stakes applications, inherently interpretable models are often preferred [17].

Interpretable models, including decision trees, rule lists, and generalised additive models, produce transparent decision logic by design and can achieve competitive accuracy in financial applications [5]. Nevertheless, they involve trade-offs: deep trees reduce interpretability, while rule lists impose restrictive sequential structures. These limitations motivate alternative rule-based approaches such as Learning Classifier Systems.

2.4 Learning Classifier Systems

Learning Classifier Systems (LCS) are a family of evolutionary algorithms that learn a population of IF—THEN rules (classifiers) through genetic operators and reinforcement-style feedback [24]. Originating from Holland's work in the 1970s, LCS combines ideas from reinforcement learning, genetic algorithms, and cognitive science. The key distinguishing feature of LCS is that rules operate as an *unordered set*: all rules that match an input instance contribute to the prediction, typically through a fitness-weighted vote. This contrasts with decision trees (hierarchical partitioning) and rule lists (sequential evaluation).

Several LCS variants have been developed. XCS (eXtended Classifier System) uses accuracy-based fitness rather than prediction strength, promoting the evolution of maximally general and accurate rules [4]. UCS (sUpervised Classifier System) adapts XCS for supervised learning by removing the reinforcement learning components. ExSTraCS (Extended Supervised Tracking and Classifying System) further extends UCS with mechanisms for handling heterogeneous data, missing values, and expert knowledge integration [24]. ExSTraCS has demonstrated competitive performance on biomedical datasets, including genetic association studies and clinical prediction tasks [25].

A study on XCSF introduces a guided mutation operator that uses accuracy-weighted covariance of stored samples to steer rule evolution toward more relevant regions of the feature space. This approach improves learning efficiency and scalability in high dimensional problems [23]. This work highlights the value of incorporating data-driven guidance into LCS evolution, which aligns with the motivation behind RASF-LCS. However, while guided XCSF focuses on optimizing classifier shape for regression using mutation only, RASF-LCS instead emphasizes feature ranking and rule filtering to enhance interpretability and performance in credit scoring.

LCS offers several advantages for interpretable classification. Each rule is human-readable, specifying conditions on a subset of features and a predicted class. The rule population can be inspected to understand which features drive predictions and how conditions combine. Unlike decision trees, LCS rules can overlap, allowing multiple rules to jointly cover complex regions of the input space. Zhang and Urbanowicz [25] released a scikit-learn compatible implementation of ExSTraCS, facilitating integration with standard ML pipelines.

However, standard LCS mechanisms have limitations. The *covering* operator, which generates new rules when no existing rule matches an instance, typically selects attributes uniformly at random. This ignores feature importance: in domains like credit scoring, where certain attributes (e.g., income, default history) are known to be highly predictive, random selection wastes resources on irrelevant features and produces overly general rules. Additionally, newly generated rules are not validated for contextual relevance—a rule may technically match an instance but specify conditions far from the instance's actual values, reducing precision and increasing noise in the population.

2.5 Feature Selection and Mutual Information

Feature selection is a fundamental preprocessing step in ML that identifies the most relevant attributes for prediction. Methods fall

Table 1: Comparison of credit scoring approaches.

Approach	Accuracy	Interpretable	Feature-Guided
Logistic Regression	Medium	Yes	No
Decision Trees / Rule Lists	Medium	Yes	No
Neural Networks / Ensembles	High	No	No
Post-hoc XAI (LIME, SHAP)	High	Partial	No
Standard LCS	Medium	Yes	No
RASF-LCS (Ours)	**Medium–High**	**Yes**	**Yes**

into three categories: filter methods (ranking features by statistical criteria), wrapper methods (evaluating subsets using a learning algorithm), and embedded methods (performing selection during model training) [7].

Mutual information (MI) is a widely used filter criterion that measures the statistical dependence between a feature and the target class. Formally, the MI between feature X_j and class Y is defined as:

$$I(X_j; Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)},$$

where $p(x,y)$ is the joint probability and $p(x)$, $p(y)$ are marginals. MI captures non-linear dependencies and is zero if and only if the feature and class are independent. In credit scoring, MI has been used to identify predictive features such as payment history, credit utilisation, and account age [8].

Despite its utility, MI-based feature ranking has not been systematically integrated into LCS rule generation. Existing LCS implementations either treat all features equally or rely on expert-provided importance scores [24]. There is an opportunity to automatically derive feature importance from data and use it to guide the covering process, biasing rule generation toward the most informative attributes.

2.6 Positioning of This Work

Table 1 summarises how our work relates to prior approaches. Traditional statistical models and inherently interpretable ML methods (decision trees, rule lists) offer transparency but may sacrifice accuracy. Complex ML models achieve high accuracy but lack interpretability. Post-hoc XAI methods provide explanations for black boxes but do not make the model itself transparent. Standard LCS produces interpretable rules but does not leverage feature importance during rule generation.

Our RASF-LCS framework addresses the gap by integrating MI-based feature ranking directly into the LCS covering mechanism. By biasing attribute selection toward high-MI features and filtering rules based on midpoint distance, RASF-LCS produces more relevant, accurate, and compact rule populations. This positions RASF-LCS as an inherently interpretable model that leverages data-driven feature importance—combining the transparency of rule-based learning with the feature-selection benefits typically associated with more complex pipelines. To our knowledge, this is the first work to integrate MI-guided attribute selection and distance-based rule filtering into an LCS for credit approval.

3 Methods

Our approach extends the *ExSTraCS* framework [25] with a Ranked Attribute Selection with Midpoint Filtering (RASF) process during the covering (rule creation) phase. The algorithm's overall framework is shown in Figure 1, highlighting the area where our contributions are integrated. The main area for improvement is the covering process, where new rules need to be generated in *ExSTraCS*. The RASF methodology has three main components: feature importance scoring, rank-guided randomized attribute selection, and midpoint-based distance filtering. We will elaborate further the three main components in the following subsections.

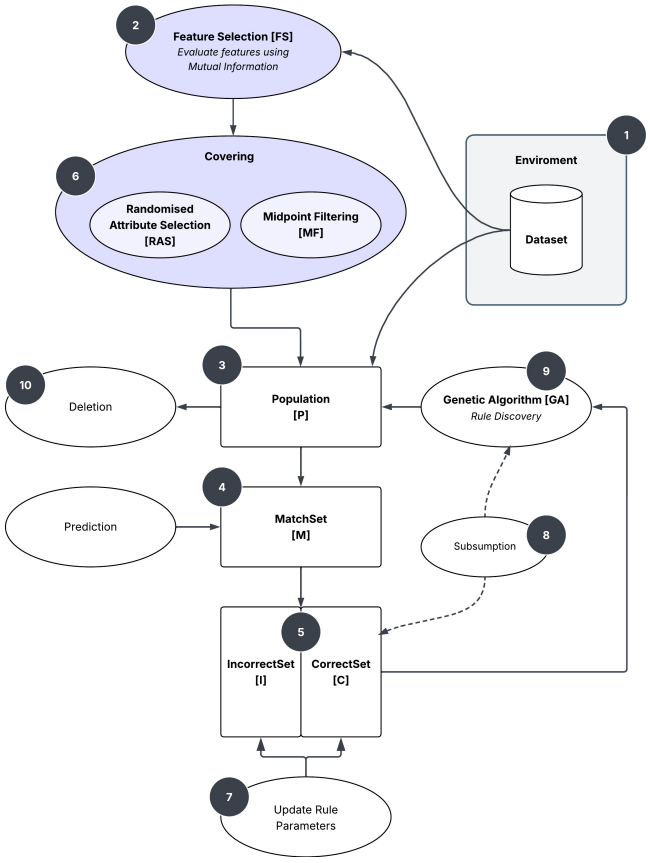


Figure 1: Overview of RASF-Enhanced LCS.

3.1 Feature Importance via Mutual Information

We first rank input features by their mutual information (MI) with the target class. Given training data (X, y), where each instance has attributes X_j and class label y, the mutual information $I(X_j; y)$ measures the reduction in uncertainty about y provided by X_j. It is defined as:

$$I(X_j; Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)},$$

where $p(x,y)$ is the joint probability of feature value x and class y, and $p(x)$, $p(y)$ are marginal probabilities. Features with higher

MI scores are more predictive of the class. We compute $I(X_j; Y)$ for each feature implemented with scikit-learn and sort features in descending order of I. A top fraction (e.g., p%) of features is then selected as the focus for rule generation. This ranked list guides the LCS to consider the most relevant attributes first.

3.2 Rank-Guided Randomized Attribute Selection

When covering a new instance, the LCS must choose a subset of attributes (of size up to a predefined rule length R) to form the rule condition. We integrate the selected feature list with computed feature importance in a rank-guided randomized attribute selection algorithm as shown in Algorithm 1. We combine deterministic ranking with randomness to balance exploitation and exploration. First, we copy and shuffle the list of top-ranked features. Then we take the first R features from this shuffled list as an *initial candidate set*. Let $m = \lceil p\,R \rceil$ where p is a user-defined fraction; we randomly select m features from this candidate set. We then randomly select the remaining $R - m$ features from lower-ranked (unselected) attributes. The union of these selections (total size R) becomes the set of features used in the new rule. This procedure biases toward high-ranking features while still occasionally including other features, maintaining diversity.

Algorithm 1 Rank-guided randomized attribute selection.

Input: Ranked feature list F, rule size R, fraction p
Output: Selected attribute subset S for rule generation
Copy and shuffle top of F into list F'
Take first R features from F' as candidate set C
Compute $m = \lceil p\,R \rceil$, $n = R - m$
Randomly select m features from C as S_1
Randomly select n features from remaining lower-ranked features as S_2
$S \leftarrow S_1 \cup S_2$
return S

3.3 Midpoint-Based Rule Filtering

Once a candidate rule is formed (with ranges on its selected attributes), we enforce a contextual relevance check before accepting the rule into the LCS population (Algorithm 2). For each attribute in the rule condition, we compute its *midpoint* with respect to the current instance $\mathbf{x}$. Specifically, if the rule for attribute i has continuous bounds $[l_i, u_i]$ around the instance value, we set

$$m_i = \frac{l_i + u_i}{2}.$$

If the attribute is discrete, we use $m_i = x_i/2$. We then calculate the Euclidean distance between the midpoint vector $\mathbf{m}$ and the instance $\mathbf{x}$:

$$d = \sqrt{\sum_{i \in S} (m_i - x_i)^2}.$$

If d is below a threshold θ, we accept the new rule (adding it to the population and match set); otherwise, we discard it. If rejected, we raise the threshold to $\theta \leftarrow d + 0.1d$ (10% above the current distance) to remain flexible. This filtering ensures that only rules whose condition midpoints are sufficiently close to the instance are used, improving relevance. The covering process will repeat until enough rules are generated with an adapted threshold θ. We illustrate how three components of our methodology are connected in Figure 2

Algorithm 2 Midpoint-based distance filtering

Input: Current instance $\mathbf{x}$, new rule with condition bounds, initial threshold θ
Output: Accept or reject the rule
for each attribute i in the rule **do**
  if attribute i is continuous **then**
    $m_i \leftarrow (l_i + u_i)/2$ {midpoint of bounds}
  else
    $m_i \leftarrow x_i/2$ {for discrete attribute}
  end if
end for
Compute $d = \sqrt{\sum_i (m_i - x_i)^2}$
if $d < \theta$ **then**
  Accept rule (add to population and match set)
else
  $\theta \leftarrow d + 0.1\,d$ {raise threshold}
  Reject rule
end if

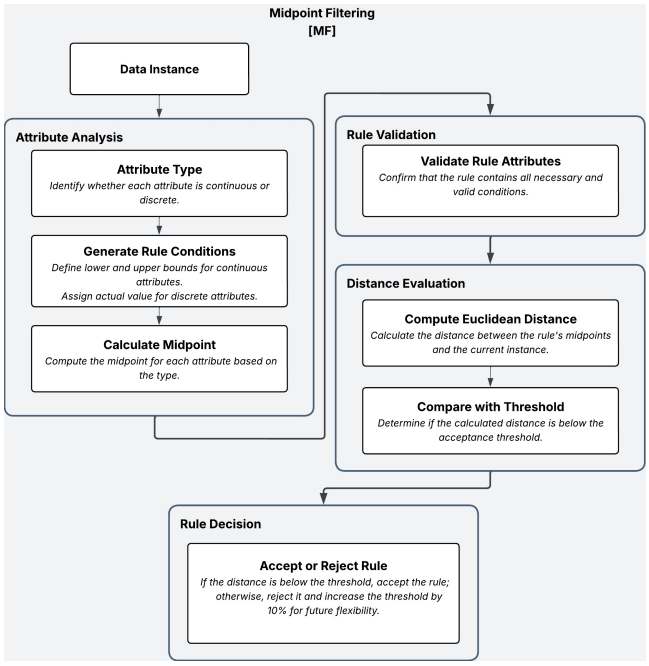


Figure 2: Midpoint filtering of newly generated rules.

3.4 Integration into LCS

The above components are integrated into the standard LCS framework as shown in Figure 1. When no existing rule covers a new

instance, the *covering* operator creates new rules to match the instance. In our RASF-LCS, covering is processed as follows: (1) Compute feature importance and feature rankings based on the computed importance. (2) Invoke Algorithm 1 to select attributes for the rule. (3) Form the rule conditions around the instance's values. (4) Apply Algorithm 2 to decide whether or not to keep the rule: if the rule passes the midpoint-distance filter, add it to the population and match set; otherwise, discard it. Subsequent LCS steps (parameter updates, subsumption, and the GA) proceed normally. By focusing rule creation on high-MI features and enforcing contextual filters, RASF-LCS produces more precise rules that accelerate learning.

4 Experimental Setup and Data

4.1 Data

The dataset utilized for this study is primarily composed of two key datasets: the Loan Approval Classification dataset and the Loan Approval Prediction dataset.

The Loan Approval Classification dataset consists of 45,000 records and 13 attributes that concern credit approval. It has both numerical and categorical attributes that include age, income, work experience, education, home ownership, loan purpose, interest rate, credit score, and loan default history. The target class is binary with a distribution of 10,000 approvals and 35,000 rejections, which represents the class imbalance that is typical of real life financial data. To be clear, attributes were assigned IDs (A01–A14) and are mentioned throughout the analysis. These feature codes, their types, and short descriptions are presented in Table 2.

Table 2: Description of Loan Approval Classification Dataset

ID	Feature	Type	Description
A01	Age	Continuous	Applicant's age
A02	Income	Continuous	Applicant's monthly or annual income
A03	Work Experience	Continuous	Years of employment
A04	Loan Amount	Continuous	Requested loan value
A05	Loan Purpose	Categorical	Type of loan (e.g., personal, education)
A06	Home Ownership	Binary	1 = Own home, 0 = Rent/Other
A07	Loan-to-Income Ratio	Continuous	Ratio of loan amount to income
A08	Interest Rate	Continuous	Applied loan interest rate
A09	Credit Score	Continuous	Normalised credit-worthiness indicator
A10	Loan Grade	Categorical	Assigned grade from credit bureau
A11	Education	Categorical	Highest qualification level
A12	Employment Type	Categorical	Full-time, self-employed, etc.
A13	Previous Default	Binary	1 = Default history, 0 = No default
A14	Approval Status	Binary	Target class: 1 = Approved, 0 = Rejected

The Loan Approval Prediction dataset comprises 58, 645 records with additional features, such as loan grade and alternative names, while maintaining a binary classification target variable indicative of the approval decision. Both datasets have been systematically assigned attribute IDs (A01-A12) for consistent reference.

In the initial phase of the study, three supplementary datasets were explored:

- Default Credit Card Clients dataset (UCI ML Repository, 30,000 records, 25 attributes),
- Application Data dataset (Kaggle, 307,000 records, 122 attributes), and
- Clean Data dataset (Kaggle, 690 records, 16 attributes).

Although these datasets were not utilized in the final experiments due to dimensionality issues, missing values, or inadequate size, they proved valuable during the early testing stages for optimizing preprocessing and model-tuning methodologies.

A systematic preprocessing pipeline was implemented on the primary datasets to enhance the quality of input features prior to model training. Key preprocessing steps included:

(1) Handling Missing Values: Incomplete records were addressed via mean substitution for numerical attributes and mode substitution for categorical attributes, preserving dataset size and minimizing bias.
(2) Categorical Encoding: Categorical variables (e.g., education, home ownership, loan purpose) were converted to numerical representations using one-hot encoding and label transformation methods to avoid introducing ordinal bias.
(3) Feature Scaling and Normalization: Numerical attributes with varying scales, such as age, income, and loan amount, were normalized to a [0,1] range, thereby improving stability during the learning process, particularly for the RASF-enhanced Learning Classifier System (LCS) sensitive to scale variations.
(4) Target Variable Adjustment: The target class distribution was inherently imbalanced, exemplified by 10,000 approved versus 35,000 rejected instances in the Loan Approval Classification dataset. No resampling was performed to artificially balance the classes, as the objective was to maintain realistic representation reflective of practical credit approval scenarios.

4.2 RASF-enhanced ExSTraCS

The hyperparameters for the ExSTraCS [25] algorithm were configured to establish the Learning Classifier System (LCS) with a maximum population of 7, 000 classifiers and 100, 000 learning iterations. Standard evolutionary configurations were applied, featuring a crossover probability of 0.8 and a mutation probability of 0.04, which is consistent with conventional LCS practices. The proposed RASF-enhanced LCS was evaluated based on a combination of predictive performance, interpretability measures, computational efficiency, and statistical testing. More LCS configurations are shown in Table 3.

In the proposed framework, midpoint-based distance filtering begins with an initial threshold of zero, which adaptively increases by 10% until a new rule is considered sufficiently close to the current instance. The attribute selection process is hybrid, with approximately half of the attributes chosen based on the highest mutual-information-ranked features, while the remaining attributes are selected from those ranked lower. This approach ensures that both relevance and diversity are preserved in the rule construction process.

Three perspectives were used to measure interpretability:

- Number of rules generated in the final classifier population. The fewer and shorter the rules, the greater is the interpretability.
- The frequency of attribute use, i.e. the number of times specific features (e.g., previous loan default history, monthly income of the loan applicant, loan amount as a percentage

Table 3: Default LCS Parameter Settings

Parameter	Default Value	Description
learning_iterations	100000	Number of training cycles to run (non-negative integer)
N	7000	Maximum micro-classifier population size (sum of classifier numerosities)
nu	1	Power parameter for accuracy importance in fitness calculation
chi	0.8	Probability of applying crossover in GA (0–1)
mu	0.04	Probability of mutating an allele within an offspring (0–1)
theta_GA	25	GA threshold: GA applied when average time since last GA exceeds this value
theta_del	20	Deletion experience threshold used in deletion vote calculation
theta_sub	20	Experience threshold required for subsumption
acc_sub	0.99	Required classifier accuracy for subsumption (0–1)
beta	0.2	Learning parameter for calculating average correct set size
delta	0.1	Deletion parameter for computing deletion votes
init_fitness	0.01	Initial fitness value assigned to new classifiers
fitness_reduction	0.1	Fitness reduction factor applied to GA offspring
theta_sel	0.5	Fraction of correct set included in tournament selection (0–1)
rule_specificity_limit	None	Overrides automatically computed rule specificity limit (RSL) when not None
do_correct_set_subsumption	FALSE	Enables subsumption within the correct set
do_GA_subsumption	TRUE	Enables subsumption between offspring and parents after GA
selection_method	'tournament'	Selection method for GA ('tournament' or 'roulette')
do_attribute_tracking	FALSE	Enables attribute tracking (AT)
do_attribute_feedback	FALSE	Enables attribute feedback (AF)
attribute_tracking_method	'add'	Method for AT updates ('add' or 'wh' for Widrow–Hoff)
attribute_tracking_beta	0.1	Learning rate for attribute tracking
expert_knowledge	None	Array of attribute filter scores if using expert knowledge
rule_compaction	'QRF'	Rule compaction method (None, 'QRF', 'PDRC', 'QRC', 'CRA2', 'Fu2', 'Fu1')
reboot_filename	None	Filename of stored model to reboot
discrete_attribute_limit	10	Threshold for treating attributes as continuous/discrete (integer or 'c'/'d')
specified_attributes	np.array([])	List of attribute indices for forced continuous/discrete handling
track_accuracy_while_fit	TRUE	Tracks accuracy during training
random_state	None	Random seed for reproducibility
use_feature_ranked_RSL	TRUE	Use feature-ranked rule specificity limit during attribute selection
feature_selection_percentage	0.75	Percentage for feature selection calculation
use_midpoint_distance_filter	TRUE	Activates Euclidean distance-based filtering for newly generated covering classifiers

of the income) were used in accepted rules. This highlights the most influential attributes in decision-making.

- Rules are readable, since every classifier is formulated in human readable forms of if-then, and one can easily trace the decisions. This is a critical point especially in credit approval applications where regulators demand model transparency.

4.3 Baseline Algorithms

To evaluate the effectiveness of the RASF-enhanced LCS, several baseline models were employed for comparative analysis, including:

Table 4: Test Aaccuracy: Baseline LCS vs. RASF-LCS.

Dataset	Base LCS	RASF-LCS	Improvement
Loan Approval	81.87%	86.51%	+4.64%
Loan Approval Prediction	79.80%	83.40%	+3.60%

Table 5: Test accuracy: RASF-LCS vs. Logistic Regression (LR).

Dataset	RASF-LCS	Logistic Regression
Loan Approval	86.51%	88.94%
Loan Approval Prediction	83.40%	90.02%

- Logistic Regression: Serving as a benchmark for traditional credit scoring. Despite its high predictive accuracy, logistic regression offers limited interpretability compared to the RASF-enhanced LCS with its attribute-level rule structure.

- Standard LCS: The original LCS (ExSTraCS) was used as a control to ascertain the impact of the proposed RASF enhancements, which included feature ranking, rank-guided attribute selection, and midpoint-based filtering. Statistical significance tests confirmed the performance improvements of the RASF-enhanced LCS were not due to random variation.

Together, these baseline models provided a comprehensive framework for assessing the performance of the RASF-enhanced LCS, demonstrating its competitive accuracy and human-readable ruleset.

5 Experimental Results

We evaluated the RASF-enhanced LCS on two publicly available credit datasets ("Loan Approval" and "Loan Approval Prediction"). Both contain applicant financial attributes and a binary outcome (approve/deny). Data preprocessing (imputation, encoding, scaling) was applied uniformly. Each LCS model (baseline and RASF) was run for 100,000 iterations over 10 random train/test splits. We compare test accuracies averaged over runs. A standard logistic regression (LR) model is used as a benchmark.

Table 4 summarizes the accuracy of the baseline LCS and RASF-LCS. In both datasets, RASF-LCS achieves higher accuracy. For example, on the Loan Approval data, test accuracy improved from 81.9% (base LCS) to 86.5% (RASF-LCS). Table 5 compares RASF-LCS to logistic regression: the LR model obtains slightly higher accuracy (around 89%), but without producing explicit rules.

In addition to evaluating predictive performance, we analysed the interpretability of the evolved rule population. The RASF–LCS framework produced a final population of 6,484 IF–THEN rules on the Loan Approval data. We inspected this rule set to understand which attributes influenced decisions. Features such as the loan–to–income ratio (`loan_percent_income`), the number of previous loan defaults (`previous_loan_defaults_on_file`), the applicant's interest rate (`loan_int_rate`) and home ownership (`person_home_ownership`) occurred most frequently in rule conditions. These attributes correspond to known risk factors in credit underwriting and indicate that the evolutionary search reinforces domain knowledge. Conversely, attributes like education

level (`person_education`), gender (`person_gender`) and credit history length (`cb_person_cred_hist_length`) appeared in few rules, suggesting limited predictive relevance under the current setup.

The rules were generally short, containing one to three conditions, and therefore easy to interpret. For example, one high–support rule states:

'*If* `loan_percent_income` < 0.35 *and* `previous_loan_defaults_on_file` $= 0$ *then* approve loan"

Such rules encode transparent decision logic that can be traced back to individual applicants. The RASF attribute filtering encourages specificity by removing irrelevant features during rule generation; the resulting rules focus on a few salient predictors and thus align closely with underwriting guidelines.

To quantify interpretability, we considered the size of the rule population and the distribution of attribute usage. The RASF–LCS model's 6.5k rule population is orders of magnitude smaller than the weight vectors used in dense models, and only a subset of attributes dominated the rules. Consequently, explaining an individual decision reduces to examining the handful of rules that match that applicant. This transparency contrasts with logistic regression (LR): although LR is mathematically simpler, its coefficients combine all features linearly, making it difficult to explain a particular approval or rejection without post–hoc analysis. In regulated financial settings, the ability to provide human–readable rationales for decisions is crucial; the rule–based structure of RASF–LCS satisfies this requirement and thus offers an interpretable alternative to black–box models.

The RASF–LCS population consists of compact, human–readable rules. Besides the earlier example ("If `loan_percent_income` $<$ 0.35 and `previous_loan_defaults_on_file` $= 0$ then approve loan"), typical rules learned from the Loan Approval data included:

- **Reject:** If `previous_loan_defaults_on_file` = "Yes" and `loan_percent_income` ≥ 0.5 then reject the application.
- **Approve:** If `person_home_ownership` $\in$ {OWN, MORTGAGE}, `loan_percent_income` < 0.4 and `loan_int_rate` < 0.12 then approve.
- **Reject:** If `loan_int_rate` ≥ 0.18 or `loan_intent` $\in$ {VENTURE, PERSONAL} then reject.

These rules reflect simple conditional logic: high debt-to-income ratios, prior defaults or very high interest rates lead to rejection, while stable home ownership combined with manageable loan payments favour approval. Such examples illustrate how the rule-based model encodes domain knowledge in a format that credit officers can readily audit.

6 Discussion

The experiments demonstrate that the RASF enhancements improve the LCS performance. Although the absolute accuracy gains (3–5%) appear modest, in credit risk even small improvements can be valuable: each percentage point of accuracy can translate to substantially fewer misclassified loans in large portfolios. The gains suggest that directing the LCS to informative features and filtering noisy rules leads to a more effective model.

Compared to logistic regression, RASF-LCS remains slightly less accurate, but gains in interpretability. Logistic regression achieved about 89% accuracy, whereas RASF-LCS achieved 86–87%. However, RASF-LCS produces an explicit rule set that can be audited and understood by humans. This interpretability is crucial in high-stakes settings [5, 17]. Regulators and domain experts can inspect the rules to verify that decisions do not rely on impermissible attributes or biases. In contrast, a LR coefficient indicates average effect but does not provide per-decision rationale.

The rule-based nature of RASF-LCS also facilitates insights. For example, we observe that default history and income ratios are prevalent in rules, matching economic intuition. Future work could enhance this by applying rule compaction methods to further simplify the rule set. Additionally, we note that the RASF-LCS required more iterations to converge than LR, reflecting LCS complexity. This could be mitigated by optimizing parameters or parallelizing training.

Limitations include potential sensitivity to the choice of threshold update factor and the percentage of top features p. We used 10% threshold buffer and $p = 50\%$ in our experiments, but these hyperparameters may require tuning for different datasets. The RASF method was tested on balanced, structured data; credit datasets often have class imbalance and categorical heterogeneity. Future research should assess RASF-LCS on larger, real-world credit datasets, and consider fairness constraints (e.g., equal opportunity) in rule selection.

7 Conclusion

This work presented RASF, a novel strategy to enhance Learning Classifier Systems for credit approval tasks. By incorporating mutual information feature ranking, rank-guided randomized attribute selection, and midpoint-distance filtering, the RASF-LCS focuses rule learning on the most relevant attributes and enforces contextual relevance of rules. Empirical evaluation on credit datasets showed that RASF-LCS improves accuracy over a baseline LCS by up to about 5% while maintaining the interpretable, rule-based representation of decisions. These results contribute to explainable AI in finance by demonstrating that a rule-based model can achieve competitive performance while providing transparency required by regulators. In future work, we plan to refine the rule set for compactness and to extend the framework to multi-class credit scenarios and fairness-aware learning.

8 Acknowledgments

We wish to acknowledge the use of New Zealand eScience Infrastructure (NeSI) high performance computing facilities to run our experimental work. The authors used ChatGPT to assist in improving the writing quality of a few selected sections of the manuscript. The authors remain fully responsible for the originality, accuracy, and overall integrity of the work.

References

[1] Hussain Abdou and Jonathan Pointon. 2011. Credit Scoring, Statistical Techniques and Evaluation Criteria: A Review of the Literature. *Intelligent Systems in Accounting, Finance and Management* 18, 2-3 (2011), 59–88. doi:10.1002/isaf.325

[2] Mohamed Azhar Ahamed, Abubakar Siddique, Trung Nguyen, Muhammad Iqbal, and Will Browne. 2026. RASF-LCS: Ranked Attribute Selection and Distance-Based Rule Filtering for Interpretable Credit Scoring. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion.* doi:10.1145/3795101.3805344

[3] E. Angelini, G. di Tollo, and A. Roli. 2008. A Neural Network Approach for Credit Risk Evaluation. *The Quarterly Review of Economics and Finance* 48, 4 (2008), 733–755. doi:10.1016/j.qref.2007.04.001

[4] Jason T. Bishop, Matthew Gallagher, and W. N. Browne. 2022. Pittsburgh Learning Classifier Systems for Explainable Reinforcement Learning: Comparing with XCS. *Proc. Genetic and Evolutionary Computation Conference* (2022), 323–331. doi:10.1145/3512290.3528767

[5] Michael Bücker, Gero Szepannek, Alicja Gosiewska, and Przemyslaw Biecek. 2022. Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society* 73, 1 (2022), 70–90. doi:10.1080/01605682.2021.1922098

[6] Consumer Financial Protection Bureau. 2022. Consumer Financial Protection Circular 2022-03: Adverse Action Notification Requirements in Connection with Credit Decisions Based on Complex Algorithms. https://files.consumerfinance.gov/f/documents/cfpb_2022-03_circular_2022-05.pdf. Issued under the Equal Credit Opportunity Act and Regulation B; guidance on adverse action notification requirements when using complex algorithms in credit decisions.

[7] Thomas M. Cover and Joy A. Thomas. 1991. *Elements of Information Theory*. Wiley, New York. First edition.

[8] Xolisa Dastile, Tansel Celik, and Murendeni Potsane. 2020. Statistical and Machine Learning Models in Credit Scoring: A Systematic Literature Survey. *Applied Soft Computing* 91 (2020), 106263. doi:10.1016/j.asoc.2020.106263

[9] Maria Dumitrescu, Salvador Hué, Christophe Hurlin, and Sedley Tokpavi. 2022. Machine Learning for Credit Scoring: Improving Logistic Regression with Non-Linear Decision-Tree Effects. *European Journal of Operational Research* 297, 3 (2022), 1178–1192. doi:10.1016/j.ejor.2021.06.053

[10] Regulation EU. 2016. General Data Protection Regulation (Regulation (EU) 2016/679). https://eur-lex.europa.eu/eli/reg/2016/679/oj. [Online; accessed April, 2026].

[11] Haifeng Gao, Gang Kou, Hu Liang, Hong Zhang, Xueli Chao, Chak-chun Li, and Yingfan Dong. 2024. Machine Learning in Business and Finance: A Literature Review and Research Opportunities. *Financial Innovation* 10, 1 (2024), 86. doi:10.1186/s40854-024-00629-z

[12] Massimo Guidolin and Manuela Pedio. 2021. Sharpening the Accuracy of Credit Scoring Models with Machine Learning Algorithms. In *Data Science for Economics and Finance*. Springer. doi:10.1007/978-3-030-57802-5_4

[13] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/1705.07874 Introduces SHAP (SHapley Additive exPlanations) for feature attributions.

[14] Cathy O'Neil, Holli Sargeant, and Jacob Appel. 2024. Explainable Fairness in Regulatory Algorithmic Auditing. *West Virginia Law Review* 127, 1 (2024), 79–133.

[15] Emmanouil Papagiannidis, Patrick Mikalef, and Kieran Conboy. 2025. Responsible artificial intelligence governance: A review and research framework. *The Journal of Strategic Information Systems* 34, 2 (2025), 101885.

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://arxiv.org/abs/1602.04938 Introduces LIME (Local Interpretable Model-agnostic Explanations).

[17] Cynthia Rudin. 2019. Stop Explaining Black Box Machine Learning Models for High-Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215.

[18] Abubakar Siddique. 2021. *Lateralized Learning to Solve Complex Problems*. Ph. D. Dissertation. Open Access Te Herenga Waka-Victoria University of Wellington.

[19] Abubakar Siddique and Will Browne. 2022. Learning classifier systems: cognitive inspired machine learning for eXplainable AI. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1081–1110.

[20] Abubakar Siddique, Will N Browne, and Gina M Grimshaw. 2020. Lateralized learning for robustness against adversarial attacks in a visual classification system. In *Proc. Gen. and Evol. Comput. Conf.* 395–403.

[21] Abubakar Siddique, Will N Browne, and Gina M Grimshaw. 2021. Frames-of-reference-based learning: Overcoming perceptual aliasing in multistep decision-making tasks. *IEEE Transactions on Evolutionary Computation* 26, 1 (2021), 174–187.

[22] Abubakar Siddique, Will N. Browne, and Gina M. Grimshaw. 2023. Lateralized Learning to Solve Complex Boolean Problems. *IEEE Transactions on Cybernetics* 53, 11 (2023), 6761–6775. doi:10.1109/TCYB.2022.3166119

[23] Patrick O Stalph and Martin V Butz. 2012. Guided evolution in XCSF. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*. 911–918.

[24] Ryan J. Urbanowicz and William N. Browne. 2017. *Introduction to Learning Classifier Systems*. Springer.

[25] R. F. Zhang and R. J. Urbanowicz. 2020. A Scikit-Learn Compatible Learning Classifier System. In *Proc. Genetic and Evolutionary Computation Conference Companion*. 1816–1823. doi:10.1145/3377929.3398097