Applied Soft Computing Journal 186 (2026) 114214

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Stacking-based heterogeneous genetic programming for interpretable credit risk evaluation

Zixue Zhao [a], Qiao Lin [b], Yiran Li [c], Yue Li [d], Tianxiang Cui [b,*]

[a] School of Statistics and Mathematics, Yunnan University of Finance and Economics, 237 LongQuan Rd, Kunming, 650221, Yunnan, China
[b] School of Computer Science, University of Nottingham Ningbo China, 199 Taikang E Rd, Ningbo, 315104, China
[c] Network Security Department, Shanxi Police College, Xibeifanglu 799, Taiyuan, 030401, China
[d] School of Advanced Technology, Xi'an Jiaotong-Liverpool University, 111 Ren'ai Rd, Suzhou, 215123, China

HIGHLIGHTS

- An end-to-end pipeline for generating a heterogeneous ensemble model is proposed.
- Genetic programming is used as the meta-classifier in a stacking credit risk model.
- NGBoost and TabNet are used as base-classifiers.
- Our SH-GPC model outperforms other meta-classifiers and provides better interpretability.
- Visualization methods are used to enhance the trustworthiness of the stacking model.

ARTICLE INFO

Keywords:
Credit risk prediction
Heterogeneous stacking
Genetic programming
Shapley additive explanations
Natural gradient boosting

ABSTRACT

The development of advanced ensemble models to handle complex and large-scale datasets has become a central focus in credit risk prediction. Although ensemble methods offer strong predictive performance, stacking lacks a standardized construction pipeline and its complex structure often reduces transparency and robustness. To address these challenges, this study proposes a stacking based heterogeneous genetic programming classifier (SH-GPC), an end to end pipeline in which genetic programming serves as the meta-classifier to enhance interpretability and generalization. Through experiments on a large-scale credit risk dataset comprising millions of records, SH-GPC is shown to significantly outperform conventional homogeneous ensemble methods and several emerging models, including XGBoost, LightGBM, NGBoost, and TabNet, in terms of AUC. Compared to stacking frameworks with logistic regression or XGBoost as the meta-classifier, SH-GPC achieves better predictive accuracy while relying on a smaller number of base classifiers, thereby improving simplicity and interpretability. Transparency is further enhanced by representing the GP meta-classifier as evolved symbolic expressions and syntax trees. Additionally, the incorporation of the Shapley additive explanations (SHAP) technique enables visualization and attribution of each base classifier's contribution, offering insights into the model's internal decision logic. This study demonstrates the applicability of evolutionary algorithms in ensemble learning and introduces a new framework for credit risk modeling that achieves a balance between accuracy, stability, and interpretability.

1. Introduction

The credit risk prediction model is a type of predictive model that uses historical data to forecast the probability of default (PD) of a credit customer in the future. This prediction helps determine whether to approve a loan or calculate the expected loss. Currently, research focuses more on categorized credit risk models to differentiate between good and bad credit groups, known as a binary classification task [27,60]. Traditional credit risk prediction classifiers handle linear relationships between features, such as the Z-score using linear discriminant analysis (LDA) and logistic regression (LR). Some advanced

* Corresponding author.
Email addresses: zz1937@ynufe.edu.cn (Z. Zhao), qiao.lin@nottingham.edu.cn (Q. Lin), 250048@sxpc.edu.cn (Y. Li), Yue.Li@xjtlu.edu.cn (Y. Li), tianxiang.cui@nottingham.edu.cn (T. Cui).

https://doi.org/10.1016/j.asoc.2025.114214
Received 15 December 2024; Received in revised form 4 September 2025; Accepted 9 November 2025
Available online 12 November 2025

classifiers manage more complex nonlinear relationships, such as support vector machine classifier (SVC) and decision tree (DT). However, the era of big data has recently introduced more challenges to credit risk prediction for banks and P2P companies due to the complexity of the global economic situation and continuous advancements in data mining technology. New credit risks manifest in the following ways: first, there has been a dramatic increase in the amount of customer data and feature dimensions [67]; second, since the COVID-19 pandemic, online lending has dominated credit institutions, intensifying risks and requiring higher accuracy in predicting bad customers and model credibility [44]. The predictive performance of single algorithms has declined when faced with complex feature relationships, making it difficult to meet model assumptions. Consequently, integrated models have become increasingly popular for credit risk prediction. For example, a less powerful DT, known as a weak or base-classifier, can be combined with several other trees to form a forest, called an ensemble classifier. This ensemble classifier performs better than a single classifier [37]. If the base-classifiers of this integrated approach are all algorithms of the same nature, they form homogeneous ensemble models, such as random forest (RF) [10], eXtreme gradient boosting (XGB) [13], light gradient boosting machine (LGBM) [35], and CatBoost (CAT) [51]. When heterogeneous base-classifiers are combined, they form heterogeneous ensemble models, such as a stacking based classifier. Various studies have shown that homogeneous ensemble models significantly improve credit risk prediction [1,3]. On the other hand, heterogeneous ensemble classifiers show varied performance due to the different choices of base-classifiers. However, most researchers believe that heterogeneous models perform better due to the greater diversity in base-classifiers [39,67]. Models using the stacking strategy are considered more stable than those using the voting strategy [39] and are better suited to addressing class imbalance in datasets [37]. However, heterogeneous ensemble classifiers also present certain disadvantages. They increase the risk of overfitting and bias [42], reduce model transparency, and potentially turn into a black-box model that can conceal numerous errors [54]. In domains involving high-stakes decision-making, such as risk management and financial forecasting, understanding the logic behind model predictions is essential to meet regulatory and stakeholder requirements [30]. The interpretability of credit risk models should be standardized by LR, ensuring they are at least as transparent [7].

Creating a robust heterogeneous ensemble classifier to meet the stringent requirements of financial institutions for credit risk prediction models necessitates careful consideration of integrating multiple classifiers. Each individual base-classifier should be leveraged for its strengths while mitigating its weaknesses. Simultaneously, attention must be given to the critical trade-off between accuracy and interpretability in machine learning models [72]. This work intends to tackle the following research questions: How to choose base-classifiers for a heterogeneous model? How to select a meta-classifier in the stacking model? How to improve interpretability, and balance the trade-off between accuracy and interpretability?

To address the challenges, an end-to-end pipeline named Stacking-based Heterogeneous Genetic Programming Classifier (SH-GPC) is introduced. It follows a two-level stacking structure comprising heterogeneous base classifiers in the first level and employs genetic programming (GP) as the meta-classifier in the second level. By following the SH-GPC framework, practitioners can directly construct an interpretable and effective predictive model tailored specifically for credit risk prediction on large datasets. Compared to traditional stacking methods, SH-GPC offers several notable contributions:

- Although GP appeared in early credit scoring, it has rarely been used as a meta-classifier in stacking, and this study investigates its potential in that role. GP provides flexible nonlinear composition of base-learners and supports explicit control of model form. This study defines clear dimensions of expression complexity and then establishes principled rules and a streamlined procedure to select GP expressions, prioritizing accuracy and parsimony through a lexicographic ordering of complexity measures. Experiments show consistent gains, with AUC improvements of up to 3 % over baseline meta-classifiers.
- Natural gradient boosting (NGB) and tabular neural network (TabNet) are introduced to the pool of base-classifiers for the first time and yield strong gains. NGB captures predictive uncertainty and stabilizes performance, while TabNet exploits the structure of tabular data. Empirical results show that combining NGB with tree ensembles such as XGB improves overall accuracy rather than causing redundancy, confirming their complementary value. The pipeline supports an extensible pool of base-classifiers in other scenarios.
- Previous ensemble studies have mainly focused on the diversity principle. This work instead introduces an explicit EDP selection process that screens base-classifiers by effectiveness (E), diversity (D), and pruning (P). Diversity is not treated as a static rule but is dynamically quantified with adaptive thresholds, which prevents redundancy while retaining complementary models. Combined with the EDP process, a GP meta-classifier achieves higher performance with compact and interpretable expressions.
- Interpretability is integrated into the automated stacking pipeline. Shapley additive explanations (SHAP), combined with the symbolic GP expression, provide a fully traceable decision path from Level–1 base outputs to the Level–2 meta decision and the final score. The system reports global contributions of each base-classifier and case level rationales for every customer, which mitigates the complexity of stacking and enables clear end to end explanations to stakeholders.

The remainder of the paper is organized as follows. Section 2 reviews the various types of individual and ensemble models used for credit risk prediction and explores the combination of integrated approaches and strategies. Section 3 introduces our proposed SH-GPC framework. The experimental results are reported in Section 4. Section 5 concludes the paper.

2. Literature review

Generally, credit risk modeling algorithms can be categorized into three main types: traditional single classifiers, intelligent single classifiers and ensemble classifiers [67]. This section will expand on this classification, offering a brief introduction to each type, and highlighting their strengths and weaknesses in credit risk prediction. The section will also introduce the underlying logic for constructing the SH-GPC model, focusing on the selection and utilization of base and meta-classifiers.

2.1. Single classifiers

Traditional single classifiers, including linear models like probit model (Probit), LDA, and LR and simple non-parametric methods such as naïve Bayes (NB) and k-nearest neighbors (KNN)), rely on fundamental statistical assumptions, offering strong interpretability, computational efficiency, and stability on moderate-sized datasets. However, strict assumptions limit their performance on complex, nonlinear, or high-dimensional data. Probit and LR remain widely used benchmarks in credit risk due to their simplicity and interpretability, but often require enhancements like hybrid forms or tree-based integrations to address nonlinearity [25,42,59]. LDA typically assumes normally distributed data, restricting its applicability and making it more common as a feature-extraction method rather than a primary classifier [18]. Despite strong independence assumptions, NB remains competitive on highly imbalanced datasets, occasionally outperforming advanced methods such as RF and SVM [4,69].

Intelligence single classifiers, including SVC, DT, multilayer perceptrons (MLP), differ from traditional methods by using data-driven approaches rather than strong statistical assumptions. These methods

excel in credit risk scenarios involving high dimensionality, nonlinearity, and imbalance. SVC handles nonlinear classification effectively through kernel functions, yet parameter tuning complexity and high computational cost limit its broader use [33,69]. DT (especially CART) offers intuitive structure and efficiency but tends toward overfit, typically requiring integration with regularization techniques or hybrid methods to improve generalization [18,25]. MLP, as general-purpose neural networks, usually outperforms traditional classifiers in predictive accuracy [32]. Addressing structured tabular data specifically, TabNet, proposed by [6], incorporates a sparse attention mechanism and sequential decision structure. It improves both predictive performance and interpretability, outperforming traditional models like AdaBoost in fraud detection [70] and significantly enhancing stacking frameworks' performance when combined with XGB as a meta learner [61].

2.2. Ensemble classifiers

Ensemble learning is a method that combines multiple learners to improve overall performance by reducing the errors of individual models which was first confirmed to be effective in 1990. Building an ensemble classification model typically involves specifying three key elements: the base classifiers to be integrated, the diversity among them, and the method of combining them [39]. Ensemble learning can be categorized into two main types based on the base classifiers used: homogeneous ensemble learning and heterogeneous ensemble learning.

2.2.1. Homogeneous ensemble classifiers

The bagging algorithm, which relies on the independence of base classifiers, is one of the simplest homogeneous ensemble models, with RF being a prime example [1]. In contrast, the boosting algorithm emphasizes the correlation between base classifiers, with new classifiers updating their weights based on previous training [72]. Prominent examples of boosting algorithms include XGB, LGBM, and CAT. XGB has become one of the most powerful models for credit risk prediction, demonstrating excellent performance across various datasets and often serving as a base classifier in ensemble models due to its high robustness [39]. Building on this, LGBM is faster and can handle larger datasets thanks to its leaf-wise growth strategy, as opposed to XGB's level-wise approach [17]. CAT introduces a greedy method to solve the dimensionality explosion problem caused by feature interaction, making it more suitable for datasets rich in categorical variables [51]. However, some studies suggest that these three boosting algorithms do not show significant performance differences in typical P2P datasets [71].

Another promising approach is NGB [24], which extends GBDT by incorporating natural gradient optimization and predictive distribution modeling. Unlike conventional models that output point estimates, NGB produces full probability distributions, offering a distinct advantage in uncertainty modeling tasks such as risk assessment. This property makes it particularly suitable for financial risk scenarios. In recent years, it has been gradually applied to corporate financial risk prediction [73]. Although NGB performs comparably to XGB and LGBM in terms of accuracy, it has not yet emerged as a comprehensive replacement for traditional tree-based models in practical applications [46].

2.2.2. Heterogeneous ensemble classifiers

Heterogeneous ensemble learning integrates diverse base classifiers to enhance predictive robustness. Central to this approach is the combination rule, with popular methods including majority vote, weighted average, reliability-based, confidence-weighted [31,66], and stacking [36]. Compared with single or homogeneous methods, heterogeneous ensembles effectively handle data imbalance and reduce overfitting which are common in credit risk datasets [37]. Stacking is a powerful but underutilized integration strategy in credit risk prediction due to its structural complexity and interpretability challenges [42,67]. Despite these limitations, stacking often outperforms bagging and boosting ensembles, prompting ongoing research focused on balancing high performance with interpretability and practicality [39,65].

A survey of heterogeneous ensemble models used in credit risk prediction since 2009 is conducted. As shown in the Table 1, recent stacking models in the credit risk domain either combine diverse algorithms as base-classifiers [1,33,60,65] or employ similar boosting-based algorithms [49,67]Padih, 2021; Wang and Zhan, 2024). Regarding meta-classifier selection, linear models continue to dominate the choices [15,19]. Therefore, recent advances in heterogeneous ensemble learning are discussed further from these two perspectives.

(1) Effectively selecting base classifiers is crucial to the performance of stacking ensembles. Historically, diversity has been considered a fundamental principle in ensemble learning, suggesting that enhancing the differences among classifiers can significantly improve overall predictive accuracy. [1] describes the relationship between the diversity of base classifiers and the ensemble effect based on the voting method. Suppose θ is the difference between T base classifiers and $\theta \in [-0, 1]$, $error(H)$ in Eq. (1) is aggregate error of ensemble learning. $\overline{err}(h)$ is average error of all individual learners. Then

$$error(H) = 1/\mathrm{T}(1 + \theta(T-1))\overline{err}(h) \tag{1}$$

Obviously when $\theta = 0$, which means base classifiers are independent of each other, the error of ensemble learner will be reduced by T times. However, if classifiers with little difference in predictive power ($\theta = 1$) are combined, the final classifier will not improve much either. Therefore, the greater the diversity between base classifiers, the better the combination.

However, recent studies have critically reassessed this principle. [62] argue that excessive diversity is not universally beneficial, as it involves a clear trade-off with model bias and variance; prioritizing diversity excessively can reduce individual classifiers' accuracy, thereby harming the ensemble's overall performance. [9] further validate this viewpoint by highlighting a boundary effect of diversity, proposing a pruning strategy to balance diversity and accuracy by removing redundant classifiers. Additionally, [8] confirm that excessive diversity may lead to overfitting in stacking models. In recent years, several novel strategies have emerged to address redundancy and inefficiencies in classifier selection. [34] introduced a genetic algorithm-based method to dynamically optimize base-classifier combinations, significantly improving predictive performance. [26] proposed a dual-criterion strategy based on feature generation, emphasizing that classifier selection should simultaneously consider individual accuracy and predictive diversity among classifiers. These studies indicate a clear trend toward more refined and targeted strategies for selecting base-classifiers in stacking ensembles.

(2) In terms of meta-classifiers, LR remains popular as a meta-learner due to its simplicity, interpretability, and stability [36,42]. More expressive models such as XGB, LGBM and neural networks have gained increasing popularity [61]. A particularly promising development is the adoption of GP as a meta-learner, given its powerful symbolic regression capabilities. [8] demonstrated that GP could automatically generate effective nonlinear ensemble rules with built-in feature selection, significantly improving predictive accuracy and interpretability. [55] further confirmed that GP-generated combination rules outperformed traditional linear meta-learners in terms of complexity and efficiency. However, the application of GP as a meta-learner remains relatively limited, with most studies employing GP solely for feature selection or optimization of base classifiers, and lacking comparative analyses against traditional meta-learners. Additionally, GP-based stacking can be computationally intensive and prone to overfitting, necessitating careful parameter tuning and regularization strategies (e.g., semantic constraints and pruning methods) [9].

Table 1
A review of various heterogeneous ensemble models.

Year	Authors	Combination rules	meta-model	Classifiers applied										
				SVC	RF	DT	NB	LR	Boosting			kNN	MLP	Others
									XGB	LGBM	CAT			
2009	Tsai&Chen [60]	stacking	Cluster			✓	✓	✓					✓	
2009	Hung&Chen [33]	stacking	selective method	✓		✓							✓	
2010	Hsei & Hung [31]	confidence-weighted average		✓			✓						✓	
2016	Alaraj [2]	consensus approach		✓	✓	✓	✓						✓	
2017	Abellan [1]	stacking	multi-rules	✓		✓		✓						
2018	Xia et al. [63]	stacking and bagging	XGB	✓	✓									gaussian process classifer
2020	Li et al. [38]	linear-weighted						✓	✓				✓	
2021	Padih et al. [49]	stacking	boosting methods					✓	✓	✓	✓			AdaBoost
2021	Zhang et.al [66]	confidence-weight voting		✓			✓						✓	
2021	Cui et al. [15]	stacking	linear regression		✓	✓		✓		✓				
2021	Hugo et al. [19]	stacking	linear model		✓			✓					✓	LSTM,CNN
2023	Fan&Liu [39]	weighted voting			✓				✓	✓				
2023	Wei [65]	stacking	DT		✓	✓	✓	✓					✓	random committee
2024	Yang&Xiao [64]	stacking	constrained optimization		✓	✓			✓	✓				GBDT, extra tree
2024	Wang&Zhang [61]	stacking	XGB	✓					✓	✓	✓	✓		TabNet
2024	Proposed method (This paper)	stacking	LR,XGB,GP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	NGB,TabNet

2.3. Interpretability

The importance of explainable artificial intelligence (XAI) has gained increasing attention among researchers [48]. "Explaining a prediction or a model" involves providing qualitative explanations, whether in text or visual form, to help human users comprehend the relationship between the characteristics of each example and the final prediction [52]. Thus, the interpretability of models is increasingly regarded with the same importance as accuracy [52]. To address the interpretability issue, several model-agnostic methods have been proposed. These methods are independent of specific models and are also known as ex-post methods [28]. Specifically, they include: (1) feature importance ranking, such as information-gain-based XGBoost algorithms, to identify key reasons behind customer defaults in credit risk management; (2) visualization of the classifier's training process, which is effective only for low-dimensional data due to dimensional constraints; (3) intrinsic explainable models, such as typicality selection, which simulate complex ensemble behaviors to enhance model generalization and interpretability; and (4) analysis of the internal structure and weights of models to further understand their decision-making mechanisms [43].

On the other hand, interpretability can be categorized into global and local interpretability. Local interpretability pertains to understanding the behavior of a specific instance, while global interpretability refers to the model's performance across an entire dataset. Methods focused on local interpretability, such as Local Interpretable Model-Agnostic Explanations (LIME) proposed by [52], aim to provide insights into individual predictions. Other notable approaches include SHAP [41], and Anchors [53]. LIME emphasizes improving local interpretability by quantifying the impact of each feature on individual predictions, thereby enhancing trust in the model among decision-makers [52]. SHAP directly computes the marginal contribution of each feature to the model's output. At the local level, SHAP decomposes each prediction to show the cumulative impact of each feature. Globally, SHAP provides insights into global feature importance [41]. While tree models can also indicate feature importance, they do not explicitly reveal the direction (positive or negative) of each feature's influence on predictions or interactions between features. SHAP values address these gaps [40]. By conducting a comparative analysis of various XAI tools, including LIME, Anchors, and SHAP, across different classifiers without specific metrics, [44] and [68] argue that SHAP stands out as one of the most powerful methods for explaining black-box models.

However, there are arguments against the notion that model accuracy and interpretability are mutually exclusive. Some researchers contend that modern machine learning algorithms can achieve comparable accuracy to simpler models in certain applications [5], such as linear models or DT combined with LR. The opacity of black-box models poses challenges for human stakeholders because trusting these models means relying on both the model itself and the entire dataset, where issues like missing or biased data can significantly impact outcomes [54]. Research suggests that integrating explanation-friendly mechanisms into complex models does not necessarily compromise prediction accuracy [12].

Current research on stacking-based credit risk prediction has several limitations. The base classifiers used are still mostly traditional models such as LR, DT, SVC, GBDT, and XGB. Few newer algorithms have been explored as base classifiers. Interpretability is also limited in stacking models, restricting their real-world use. In addition, there is a lack of clear guidelines on how to select base classifiers effectively, especially regarding diversity and accuracy. Therefore, it is necessary to explore more diverse and advanced models, such as Genetic Programming (GP). An in-depth study of GP as a meta-classifier is particularly important due to its strong interpretability and flexibility. Future research should also focus on improving interpretability, clearly defining classifier-selection principles, and making stacking models more cost-effective for practical use with large datasets.

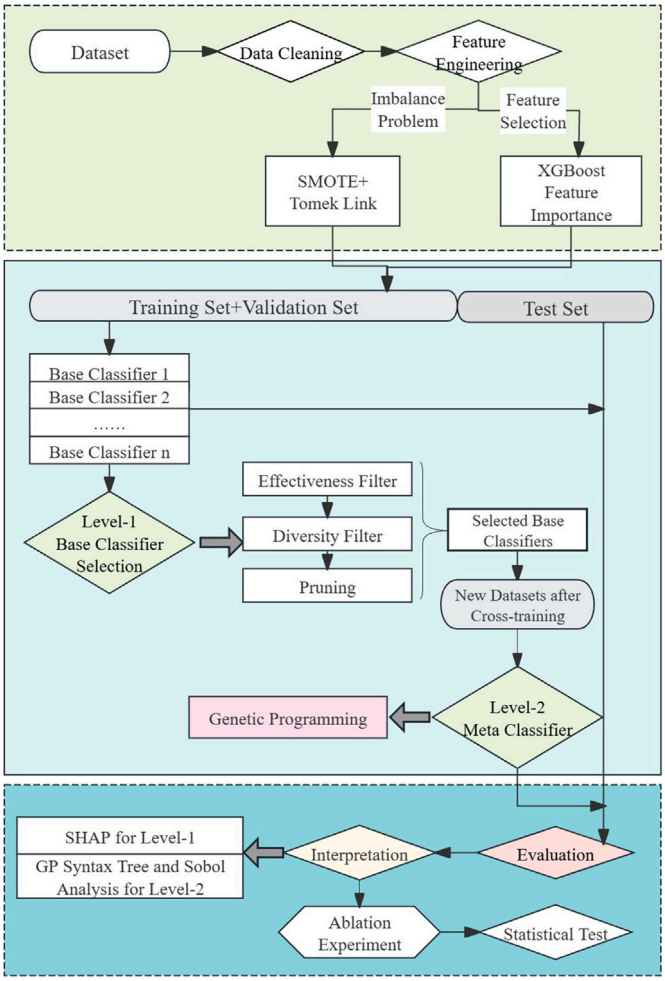

Fig. 1. Experimental framework and roadmap. The green section indicates data preprocessing, the light blue section represents the model development process, and the dark blue section denotes model evaluation, explanation and supplementary tests.

3. Methodology

This section focuses on the SH-GPC model building process and the methods used for its evaluation and interpretation. The process consists of three parts: data preprocessing, model building, and model explanation (see Fig. 1).

3.1. Data preprocessing

The dataset used for modeling in this study is based on credit approval records from 2007 to 2022, published by Lending Club (LC),[1] Inc. It features a large volume of data, a wide range of features, and a long time span, making it well-suited for training complex models. This dataset has also been widely adopted in previous studies on credit risk prediction [42]. It includes both loan application and repayment records, with 151 original features describing borrower profiles and credit behavior. These variables cover loan terms, employment status, income levels, credit utilization, and repayment history. The target variable, *loan_status*, indicates whether a loan was fully repaid (labeled as 0) or was charged off or delinquent (labeled as 1), and is used for binary classification in credit risk modeling tasks. The dataset contains a total of 2,925,486 samples, of which 394,629 are labeled as positive cases (i.e., bad customers) and 2,530,857 as negative cases (i.e., good customers), resulting in a default rate of 13.489 %. The imbalance ratio (the proportion of majority class to minority class) is approximately 6.4, indicating a serious class imbalance problem. Accordingly, the following data preprocessing procedures were applied.

Data cleaning. This study follows the generic template for financial machine learning data processing and systematically cleanses the raw data. First, very low information variables are eliminated: all features with a proportion of missing values or a proportion of unique values exceeding 90 % are removed to eliminate redundant and invalid information. At the same time, based on data compliance considerations, features that may involve legal or ethical risks, including sensitive attributes such as gender and race, are removed. Second, missing values in the remaining variables are filled in with the mean or median, depending on their data type, to mitigate the interference of missingness in the modeling process. Additionally, label encoding is applied to the categorical variables to transform them into a numerical format that can be handled by the model. Temporal variables are interval binned based on their distributional characteristics to reduce noise fluctuations and extract long-term trend information. For the skewed features in numerical variables, Z-score normalization and logarithmic transformation are applied to alleviate the scale inconsistency and long-tailed distribution problems, and to enhance the stability of subsequent modeling.

Feature selection. The Pearson correlation coefficients between all numerical features were calculated, and if the absolute value of the correlation coefficient between a certain pair of features was higher than

[1] https://www.lendingclub.com

0.9, the one with stronger business interpretability or more significant impact on the model performance was retained, and the other one was deleted to reduce redundancy. Next, the built-in feature importance assessment mechanism provided by XGB was used to further filter the variables after the initial cleaning. Based on the gain-based importance ranking generated in the training phase, the features with very low contribution were eliminated, and only the subset with higher cumulative contribution was retained for the subsequent modeling process.

Balancing technique. Given the large sample size and high imbalance ratio, we refer to the research of [69] on the application of balancing techniques for large datasets, opting for the SMOTETomek comprehensive resampling method. This involves initially performing a 1:1 oversampling using the SMOTE [47], followed by the under sampling method of Tomek Links [58], thereby enhancing the clarity of the decision boundary. Specifically,Each each new sample is generated by the Eq. (2):

$$x_{new} = x + ran(0,1) \times |x - x_k| \tag{2}$$

where x is a random instance of minority class,x_k is one of the neighbors, selected from other minorities by a certain ratio, and $ran(0,1)$ is a random factor to avoid duplication. Following the generation of new samples, instances from the majority class x_m and minority class x_n located at all boundaries will be paired based on the nearest neighbor principle. A pair (x_m, x_n) is identified as a Tomek link if there is no third sample z such that its distance from x_m satisfies $d(x_m, z) < d(x_m, x_n)$ or its distance from x_n satisfies $d(z, x_n) < d(x_m, x_n)$. All Tomek links will subsequently be eliminated.

3.2. End to end SH-GPC framework

A two-level ensemble architecture is employed. Level–1 fits a pool of heterogeneous base classifiers on the training data. Level–2 evolves a symbolic GP meta-classifier, taking as its terminal set the retained Level–1 outputs. Model selection is driven by a lexicographic objective:

$$\max_f \ \mathrm{Perf}_{\mathrm{val}}(f) \quad \text{s.t.} \quad \text{if } \mathrm{Perf}_{\mathrm{val}}(f_a) \approx \mathrm{Perf}_{\mathrm{val}}(f_b),\ \min C(f), \tag{3}$$

where $\mathrm{Perf}_{\mathrm{val}}(\cdot)$ denotes the chosen validation metric (e.g., AUC) and serves as the primary criterion for predictive quality. The notation "≈" indicates equality within a tolerance ε, and $C(\cdot)$ is the explicit structural complexity measure defined in §3.2.3. This tie-breaking ensures that, among models with statistically indistinguishable validation performance, preference is given to those with lower complexity, thereby improving interpretability and deployability. Fig. 1 provides an overview of the pipeline (Level–1 → Level–2 → final selection).

3.2.1. Level–1: EDP selection protocol for base-classifier screening

EDP integrates the *effectiveness* (E) and *diversity* (D) principles with a final *pruning* (P) step. Diversity arises both inherently (heterogeneous algorithms) and ex post via perturbations. In this study, base predictions are represented as binary column vectors; pairwise similarity is quantified by correlation ρ_{pq} (Eq. (4)).

$$\rho = \frac{ad - bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}} \tag{4}$$

The values of a, b, c, and d refer to the Table 2. A larger ρ indicates lower diversity. Additional variability is induced by (i) data perturbation (cross-validation subspaces) and (ii) output perturbation (stacking-produced probabilities), which expose nuanced differences to the meta-classifier. Low diversity alone does not trigger immediate exclusion; the effectiveness principle is assessed jointly. Single-model accuracy and stability remain necessary but not sufficient: a strongest-alone classifier (e.g., SVC) may underperform in certain ensembles, whereas unstable learners (e.g., DT) can enhance complementarity and improve the stack [1]; conversely, inflating weak learners or removing

Table 2
Confusion matrix for measuring pairwise diversity.

		h_i	
		Positive (+1)	Negative (−1)
Truth	Positive (+1)	a(TP)	c(FN)
	Negative (−1)	b(FP)	d(TN)

all strong-but-similar ones solely for "diversity" can be harmful [39]. Accordingly, EDP first filters out underperforming or unstable classifiers (effectiveness), then applies a diversity matrix to avoid highly redundant pairs, while retaining structurally similar models when they exhibit clear complementarity (feature extraction, parameter sensitivity, or local decision behavior). Because effectiveness plus a single diversity score may still preserve low-value models [16], a pruning stage is added: global SHAP importance is computed from a logistic-regression combiner over base predictions, yielding a fairer, more interpretable contribution ranking than tree-based importance (tree-SHAP can inherit model selection bias) [57]; local SHAP values ϕ (Eq. (5)) are aggregated to global scores, and bases with negligible unique contribution are removed before the GP meta-classifier [41].

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S\cup\{i\}}\left(x_{S\cup\{i\}}\right) - f_S\left(x_S\right) \right] \tag{5}$$

where ! mark represents factorial, F denotes the set of all features, and S is a subset of F excluding the i^{th} feature. The term $|F|$ represents the total number of input features, while $|S|$ denotes the number of features in S. Additionally, x_S denotes the values of the input features within the set S. A model $f_{S\cup i}$ is trained on the feature values of S and i and another model f_S is trained on the values of the withheld feature i. Then the global SHAP value [68] I is calculated as Eq. (6).

$$I_j = \frac{1}{n} \sum_{k=1}^{n} \left| \phi_i^{(k)} \right| \tag{6}$$

where j denotes each feature, and n is the number of total samples.

To sum up, the selection procedure proceeds as Algorithm 1. First, an effectiveness filter is applied: every candidate classifier whose individual performance score $s(h)$ (e.g., AUC or F1-score) falls below the threshold τ_{eff} (Eq. (8)) is removed, leaving only models with acceptable standalone performance. Let the performance scores of all candidate classifiers be $S = \{s_1, s_2, \ldots, s_T\}, s_t = s(h_t)$, where $s(\cdot)$ denotes any scalar performance metric. Compute the sample mean and standard deviation

$$\mu = \frac{1}{T} \sum_{t=1}^{T} s_t, \sigma = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (s_t - \mu)^2}, \tag{7}$$

and define the effectiveness threshold as

$$\tau_{\mathrm{eff}} = \mu - \sigma. \tag{8}$$

On the reduced set, a diversity filter based on pairwise similarity ρ (Eq. (4)) is applied. Let $\bar{s}$ denote the mean validation score. The similarity cutoff $\delta(\bar{s})$ and the tiny gain tolerance $\epsilon(\bar{s})$ are set adaptively as monotonic functions of $\bar{s}$; the concrete values are specified in Algorithm 1. For any pair (h_i, h_j), if the two classifiers are deemed too similar and the averaged predictions bring only a negligible improvement over the better individual, the lower-scoring member will be discarded; otherwise both are retained. This removes near duplicate models while preserving genuinely complementary ones.

Next, the surviving classifiers are used to train a logistic-regression combiner on the validation data, from which global SHAP importance $I(h)$ is obtained using Eq. (6). Any classifier whose SHAP value satisfies $I(h) < \tau_{\mathrm{shap}}$ (Eq. (10)) is pruned, ensuring that each retained model

Algorithm 1 EDP screening based on AUC, diversity (ρ), and global SHAP(I_j).

Require: classifier set $\mathcal{H}_0 = \{h_1, \dots, h_T\}$, validation data D, thresholds $\tau_{\text{eff}}, \delta, \tau_{\text{shap}}$
Ensure: pruned set $\mathcal{H}_\star$
$\mathcal{H} \leftarrow \{h \in \mathcal{H}_0 : s(h) \geq \tau_{\text{eff}}\}$ ▷ effectiveness filter
$\bar{s} \leftarrow \text{mean}_{h\in\mathcal{H}}\, s(h)$
set δ by a three-tier rule based on $\bar{s}$:
  $\delta = 0.97$ if $\bar{s} \geq 0.95$; $\delta = 0.95$ if $0.85 \leq \bar{s} < 0.95$; $\delta = 0.92$ otherwise
set ϵ analogously: 0.002, 0.005, 0.010 for the same three tiers
for all (h_p, h_q) with $p < q$ in $\mathcal{H}$ **do**
  compute ρ_{pq} via Eq. (4)
  compute tiny-gain Δs_{pq} from averaging validation outputs of h_p, h_q
  if $\rho_{pq} \geq \delta$ **and** $\Delta s_{pq} \leq \epsilon$ **then**
    remove $\arg\min_{h\in\{h_p,h_q\}} s(h)$ from $\mathcal{H}$ ▷ diversity filter
  else
    keep both (preserve complementary pairs)
  end if
end for
train logistic regression on predictions of $\mathcal{H}$ over D
for all $h \in \mathcal{H}$ **do**
  compute global SHAP $I(h)$ via Eq. (6)
  if $I(h) < \tau_{\text{shap}}$ **then**
    remove h from $\mathcal{H}$
  end if
end for
$\mathcal{H}_\star \leftarrow \mathcal{H}$ ▷ Pruning to final set fed to GP meta-classifier **return** $\mathcal{H}_\star$

provides a unique and meaningful contribution to the ensemble. To determine the threshold we adopt an 80 % cumulative contribution rule as Eq. (9). First, sort the global SHAP importance in non-increasing order, $I_{(1)} \geq I_{(2)} \geq \cdots \geq I_{(t)} \cdots \geq I_{(T)}$, and let S be their total. The cut-off index k is defined by

$$k = \min\Big\{ \ell \;\Big|\; \sum_{t=1}^{\ell} I_{(t)} \geq 0.8\, S \Big\}, S = \sum_{t=1}^{T} I_{(t)}. \tag{9}$$

Where ℓ denotes the candidate cut-off index. The corresponding threshold is then

$$\tau_{\text{shap}} = I_{(k)}. \tag{10}$$

Every classifier satisfying $I(h) \geq \tau_{\text{shap}}$ is kept, and those with $I(h) < \tau_{\text{shap}}$ are pruned.

The resulting subset $\mathcal{H}_\star$ simultaneously satisfies individual effectiveness, sufficient diversity, and non-trivial marginal contribution, and it is supplied to the next level.

3.2.2. Cross-training bridge from level–1 to level–2

After EDP screening, a cross-training cascade generates Level–1 outputs for Level–2 training. The dataset is split once into training and test, and test labels are never used during training (see Fig. 2). Each retained base-classifier from $\mathcal{H}_\star$ is cross-trained with five-fold validation: in each round it is trained on four folds and predicts probability of default (PD) on the held-out fold of the training set, and it also produces PD predictions on the fixed test set. For the training set, the fold-wise predictions are concatenated to form a full-length column for that classifier. For the test set, the five predictions are averaged to obtain the corresponding test column. Stacking these columns across all base-classifiers yields a new training matrix for the meta-classifier and a matched test matrix of the same size as the original splits. Columns are denoted $X_1, X_2, \dots$ and are fed to meta-classifier.

3.2.3. Level–2: GP meta-classifier and expression selection

In stacking, the meta-classifier determines how Level–1 predictions are fused. Linear meta-learners such as LR provide transparency and robustness but often cap accuracy [37]. High-capacity meta-learners such as XGB can raise accuracy, yet interpretability drops. To balance these goals, this work uses GP as the meta-classifier. GP evolves symbolic expressions as trees of terminals and primitive operators through selection, crossover, and mutation, optimizing a validation objective [22,50]. The result is an explicit equation with growth and stopping controls that limit depth and size and support auditability [21,23]. GP typically yields several high-scoring candidates; a Metric then Parsimony rule selects compact formulas without sacrificing the primary metric. LR- and XGB-based stacks are built as baselines. The next subsection defines the GP search space and selection rule, followed by performance and interpretability analyses.

Primitive set and terminals. The primitive set is strictly binary arithmetic $\mathcal{O} = \{+, -, \times\}$. Terminals comprise the Level–1 outputs $\mathcal{H}_\star = \{X_1, \dots, X_j, \dots\}$ and optional ephemeral constants; the count and value range of the latter are read from the problem-file header parameters. This configuration keeps the search space simple and expressions auditable.

Evolution setup. Individuals are full binary expression trees initialized by a ramped half-and-half scheme. Initialization and hard limits: maximum tree size $T \leq 10{,}000$, maximum depth $D \leq 5$, population size 100,000. Fitness $F(g)$ is a loss on a fixed validation split,

$$F(g) = \sum_{i=1}^{n} \ell\big(y_i,\ \hat{y}_i(g)\big), \tag{11}$$

minimized during evolution, where i indexes the validation samples, y_i is the observed label, and $\hat{y}_i(g)$ is the prediction of expression g for the i-th sample. Per generation: crossover probability 0.90, mutation probability 0.05, with reproduction filling the remainder; the run is capped at 100 generations. Stopping occurs when $F(g) = 0$, or when the best $F(g)$ shows no improvement for 5 consecutive generations; otherwise the algorithm terminates at generation 100.

Candidates and notation. For the m-th configuration, let $\mathcal{G}_m$ denote the set of candidate symbolic expressions produced by GP. For any $g \in \mathcal{G}_m$, let $s(g)$ denote the value of the chosen validation metric; let $U(g) = \{X_j : X_j \text{appears in } g\}$ and $|U(g)|$ its size. From the syntax tree, three structural measures are computed: $T(g)$ (total node count), $D(g)$ (maximum depth), and $O(g)$ (operator count; here equal to the number of binary operations). Given complexity thresholds (τ_T, τ_D), the feasible set is

$$\mathcal{F}_m = \{\, g \in \mathcal{G}_m :\ T(g) \leq \tau_T,\ D(g) \leq \tau_D \,\}. \tag{12}$$

Selection rule (Metric then Parsimony). For each m, define the feasible set $\mathcal{F}_m = \{\, g :\ T(g) \leq \tau_T,\ D(g) \leq \tau_D \,\}$, and retain only candidates that pass basic validity checks on their outputs. Let $s(g)$ denote the validation score and let $\hat{g}_{\text{best}} \in \arg\max_{g\in\mathcal{F}_m} s(g)$. Using the Hanley and McNeil formula [29], which provides a large-sample approximation of the standard error of an AUC based on the sample sizes of positive and negative cases, construct the one standard error cutoff threshold

$$c = s(\hat{g}_{\text{best}}) - z\,\text{SE}\big(\hat{g}_{\text{best}}\big), \qquad z = 1. \tag{13}$$

Here $s(\hat{g}_{\text{best}})$ is the validation score of the best candidate expression, $\text{SE}(\hat{g}_{\text{best}})$ is its estimated standard error under the Hanley and McNeil formula, z is the standard normal quantile. The final selection is restricted to $\{\, g \in \mathcal{F}_m :\ s(g) \geq c \,\}$. Ties, defined as $|s(g_a) - s(g_b)| \leq \epsilon$ with $\epsilon = 0$, are broken by smaller $T(g) \to D(g) \to O(g) \to |U(g)|$; if ties persist, the candidate generated earlier in the search is preferred, yielding $g^\star$. Analysis of per variable contributions is deferred to the ablation study; see §3.5. All tunable parameters of the above pipeline have been summarized in Table 3.

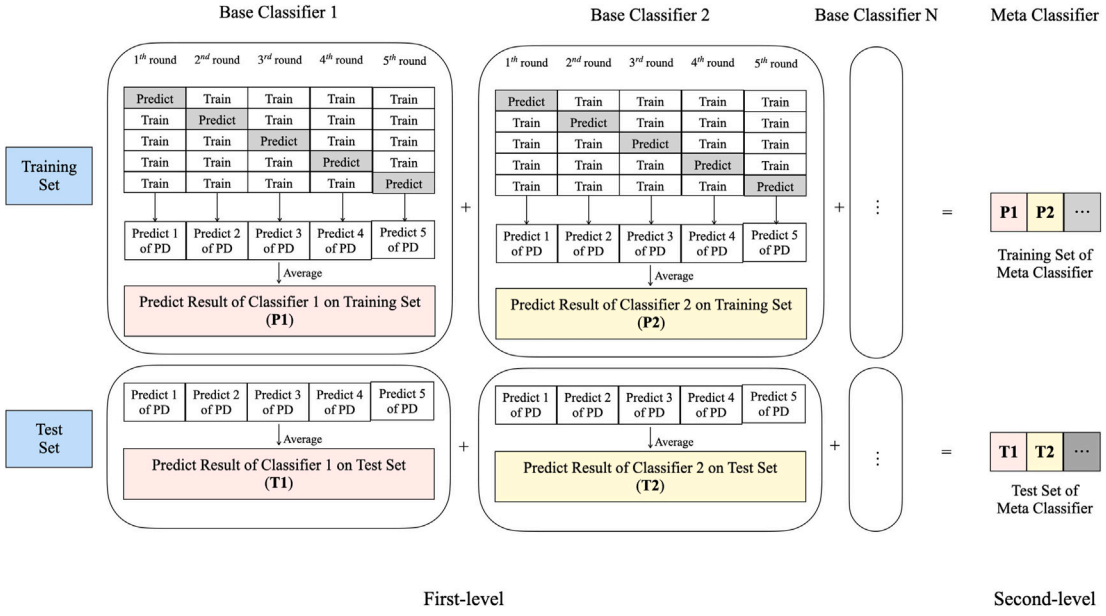

Fig. 2. The cross-training process from Level–1 to Level–2.

Algorithm 2 Metric then Parsimony selection in SH-GPC.

Input: candidate table with `model`, `expression` g, validation score $s(g)$
for each expression g **do**
  extract $U(g)$; compute $T(g)$, $D(g)$, $O(g)$ from the syntax tree
end for
Validity checks: discard candidates failing basic sanity checks on outputs
Feasible set: $\mathcal{F} \leftarrow \{\, g : T(g) \le \tau_T,\ D(g) \le \tau_D \,\}$
for each model/config m **do**
  $\mathcal{F}_m \leftarrow \{\, g \in \mathcal{F} : \texttt{model}(g) = m \,\}$
  $\hat{g}_m^{\text{best}} \in \arg\max_{g \in \mathcal{F}_m} s(g)$
  estimate $\mathrm{SE}_{\mathrm{HM}}\big(\hat{g}_m^{\text{best}}\big)$ via the Hanley–McNeil formula (based on n_+, n_-)
  set $c_m \leftarrow s(\hat{g}_m^{\text{best}}) - 1 \times \mathrm{SE}_{\mathrm{HM}}\big(\hat{g}_m^{\text{best}}\big)$
  One-SE band: $\mathcal{S}_m \leftarrow \{\, g \in \mathcal{F}_m : s(g) \ge c_m \,\}$
  Tie-break in $\mathcal{S}_m$:
  ties defined as $|s(g_a) - s(g_b)| \le \varepsilon$ with $\varepsilon = 0$
  select by smaller $T(g) \rightarrow D(g) \rightarrow O(g) \rightarrow |U(g)|$
  denote the per-model selection by $\hat{g}_m$
end for
Global selection over $\{\hat{g}_m\}$:
  $g^\star \in \arg\max_{g \in \{\hat{g}_m\}} s(g)$
  break ties by smaller $T(g) \rightarrow D(g) \rightarrow O(g) \rightarrow |U(g)|$ $\triangleright\ \varepsilon = 0$
Output: per-model selections $\{\hat{g}_m\}$ and the final expression $g^\star$

3.3. Interpretability

In the Level–1, both global and local interpretability are emphasized. SHAP values (see Eq. (5) and Eq. (6)) are used to explain how the most influential base classifiers contribute to predictions, and summary plots and decision plots are generated accordingly.

In the Level–2, a three-step analysis is applied to the selected symbolic model $g^\star$: (i) Syntax & Symbolization: Extract the human-readable formula from the GP tree (Fig. 3); (ii) Analytical Sensitivity: Compute partial derivatives. $\frac{\partial \hat{g}}{\partial X_{(i)}}$ to quantify local influence of each terminal; (iii) Global Sensitivity (Sobol analysis): Treat $g^\star$ as $g^\star(X)$ and estimate first-order Sobol indices on the validation distribution of X to measure variance contributions [56]. The first-order index (Eq. (14)) measures the fraction of output variance explained by the main effect of X_i.

$$S_i = \mathrm{Var}_{X_{-i}}(\mathbb{E}[f|X_i])/\mathrm{Var}(f) \tag{14}$$

The total-effect index (Eq. (15)) measures the fraction explained by the main effect plus all interactions.

$$S_{T_i} = 1 - \mathrm{Var}_{X_{-i}}(\mathbb{E}[f|X_{-i}])/\mathrm{Var}(f) = \mathbb{E}_{X_{-i}}[\mathrm{Var}(f|X_{-i})]/\mathrm{Var}(f) \tag{15}$$

3.4. Evaluation metrics and statistical tests

Given the high likelihood of class imbalance persisting in the test set, AUC was selected as the primary evaluation metric due to its robustness [27,64]. Furthermore, accuracy, recall, precision, and F1-score [14] were considered secondary indicators to assess classifier performance (see Tables 2 and Eq. (16)). All the metrics were comprehensively tested and ranked using the Friedman rank test. Additionally, the DeLong's test was applied specifically to the comparisons involving the AUC metric.

$$AUC = \frac{1}{2}\left(1 + \frac{TP}{TP+FN} - \frac{FP}{FP+TN}\right), Accuracy = \frac{TP+TN}{TP+FP+TN+FN},$$
$$Recall = \frac{TP}{TP+FN}, Precision = \frac{TP}{TP+FP}, F1 = \frac{2TP}{2TP+FP+FN} \tag{16}$$

The Friedman rank test is a non-parametric statistical procedure widely employed for evaluating the performance differences among multiple classifiers across various datasets. Given n classifiers (including individual and ensemble classifiers) evaluated on N different datasets, the Friedman test first ranks the results of each classifier on every dataset from best to worst, assigning ordinal ranks $1, 2, 3, \ldots, n$. In cases where classifier performances are statistically indistinguishable, average ranks are assigned accordingly. Let the average rank of the i^{th} classifier be denoted as $\bar{k}_i$. The Friedman test statistic (τ_{χ^2}) is calculated as Eq. (17) [11].

$$\tau_{\chi^2} = \frac{n-1}{n} \cdot \frac{12N}{n^2-1} \sum_{i=1}^{n} \left(\bar{k}_i - \frac{n+1}{2}\right)^2 \tag{17}$$

When the number of datasets (N) is sufficiently large, the statistic τ_{χ^2} approximately follows a chi-squared distribution (χ^2) with $n-1$ degrees

Table 3
Tunable parameters by module and their typical settings.

Module	Symbol	Typical setting	Brief description
Effectiveness	$s(\cdot)$	AUC / F1-score	Scalar score to rate each base-classifier in Level–1.
	λ in $\tau_{\text{eff}} = \mu - \lambda\sigma$	[0, 2])	Strictness of the effectiveness cutoff.
Diversity	$\delta(\bar{s})$	0.97/0.95/0.92	Similarity cutoff on ρ chosen by the mean score $\bar{s}$.
	$\epsilon(\bar{s})$	0.002/0.005/0.010	Minimal gain required when averaging a similar pair.
Pruning	p_{SHAP}	0.80/0.85/0.90	Cumulative share of global SHAP to retain the top k.
GP selection	τ_T	10,000	Maximum number of nodes in the GP expression tree.
	τ_D	[1, 10]	Maximum tree depth.
	$s(g)$	AUC / F1-score	Validation metric used in Level–2.
	z in $c = s(\hat{g}_{\text{best}}) - z\,\text{SE}$	1 (e.g., 0.5–1.64)	Width of the one-standard-error band.
	ϵ	0 (e.g., 10^{-3})	Tie tolerance when comparing scores.

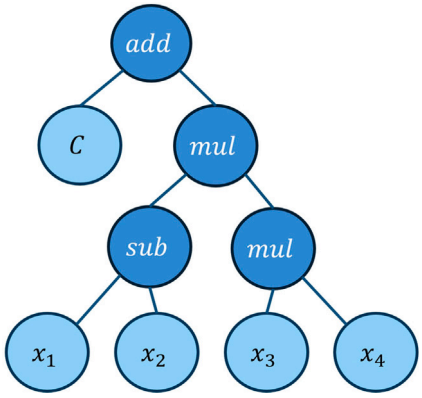

Fig. 3. An example of syntax tree. It represents a function expression $y = (x_1 - x_2)x_3x_4 + C$. The light blue nodes represent the terminal set, while the dark blue nodes represent the function set. "add" signifies addition of the two branches below, "mul" indicates multiplication, and "sub" denotes subtraction.

of freedom. In practical scenarios, it is common to convert this statistic into an F-distributed statistic (τ_F) as Eq. (18) for improved hypothesis testing accuracy.

$$\tau_F = \frac{(N-1)\tau_{\chi^2}}{N(n-1) - \tau_{\chi^2}} \tag{18}$$

Under the null hypothesis (that all classifiers have identical performance), the adjusted statistic τ_F approximately follows an F-distribution with degrees of freedom $(n-1)$ and $(n-1)(N-1)$. By comparing the calculated τ_F to the critical value from the F-distribution, classifiers' differences are statistically assessed. When the Friedman test indicates significant differences among classifiers, the Nemenyi post-hoc test is applied to identify which pairs of classifiers differ significantly. The critical difference (CD) used by the Nemenyi test [45] is computed as Eq. (19).

$$CD = q_\alpha \sqrt{\frac{n(n+1)}{6N}} \tag{19}$$

where q_α is the critical value from the Studentized range distribution at significance level α, n is the number of classifiers, and N is the number of datasets. Classifiers whose average ranks differ by more than the CD are considered significantly different. The results of the Nemenyi post-hoc test are commonly visualized using a Critical Difference (CD) diagram.

The DeLong's test is used to statistically compare differences between areas under ROC curves for two correlated classifiers [20]. Let AUC_1 and AUC_2 denote the AUC values of two classifiers. The variance-covariance matrix is computed based on the empirical placement values. Finally, the DeLong's test statistic (Z) is given by Eq. (20).

$$Z = \frac{AUC_1 - AUC_2}{\sqrt{\text{Var}(AUC_1 - AUC_2)}} \tag{20}$$

3.5. Ablation studies

Global SHAP values from a logistic regression combiner (Eq. (6)) are used to rank the retained base-classifiers after EDP. Let $m = |\mathcal{H}_\star|$ and relabel the Level–1 outputs as $X_1, \ldots, X_m$ from highest to lowest contribution. Starting from the full set, nested ablations are formed by sequentially removing the lowest ranked classifier, yielding stacks built on $\{X_1, \ldots, Xm\}$ for $n = m, \ldots, 2$. For each n, a stacking model with GP as the meta-classifier is trained on the corresponding Level–1 outputs; this configuration is denoted *meta-GP-n*. For comparative validation, the same ablation grid is constructed with LR and XGB as meta-classifiers under identical inputs and default hyperparameters, denoted *meta-LR-n* and *meta-XGB-n*. This design isolates the effect of the base-classifier pool size and enables a matched comparison across meta-classifiers.

4. Experiment

4.1. Experimental setup

Dataset. After cleaning and feature selection, 48 features remain (11 categorical, including 3 ordinal; 37 numerical). The data are split 8:2 into training and test sets (2,340,388 / 585,098 samples). The training set preserves the original imbalance ratio (IR = 6.42); it is then resampled to 4,038,358 entries with IR = 1. The test set is untouched (no balancing).

Level–1 pool and software. Fourteen base-classifiers spanning major families: linear (LR), distance (KNN), kernel (SVM), Bayesian (NB; with isotonic calibration), tree/boosting (DT, RF, XGB, LGBM, CAT, NGB), and neural (MLP, TabNet) compose the Level–1 pool for EDP, covering the mainstream models used in credit risk since 2016 [42]. Implementations use *scikit-learn* v1.6.1 defaults (Table 4) to ensure fair, transparent, and reproducible comparisons. Generic training controls only: TabNet uses a reduced mini-batch and early stopping; NGB uses

Table 4
Default hyperparameters of the 14 classifiers (scikit-learn v1.6.1 and related libraries).

Model	Key Parameters
LR	penalty = l2, solver = lbfgs, C = 1.0, max_iter = 100
LDA	solver = svd, tol = 1e-4
Probit	method = newton, no regularization
KNN	n_neighbors = 5
NB	var_smoothing = 1e-9
DT	criterion = gini, splitter = best, max_depth = None
RF	n_estimators = 100, criterion = gini, max_features = sqrt
SVC	kernel = "linear", max_iter = 10000, C = 1.0
XGB	n_estimators = 100, max_depth = 3, learning_rate = 0.1
LGBM	n_estimators = 100, max_depth = −1, learning_rate = 0.1
CAT	iterations = 1000, learning_rate = 0.03, depth = 6
MLP	hidden_layer_sizes = 100, activation = relu, solver = adam, max_iter = 200
NGB	n_estimators = 400, learning_rate = 0.03, natural_gradient = True
TabNet	n_d = 8, n_a = 8, n_steps = 3, gamma = 1.3, batch_size = 65536, virtual_batch_size = 2048, max_epochs = 100

per-iteration subsampling and early stopping. All hyperparameters are fixed *ex ante*; the test set is held out and used solely for final evaluation. GP runs under JDK 17; all other modeling uses Python 3.11.

Hardware. 12th Gen Intel Core i7-12700 (2.10 GHz), 32 GB RAM, Windows 10 Pro 64-bit.

4.2. Experimental results

4.2.1. Results of SH-GPC construction.

Following Algorithm 1, the AUC values of all candidate classifiers were evaluated on the validation set, and ROC curves were plotted (see Fig. 4). With the current scores, $\mu \approx 0.927$ and $\sigma \approx 0.079$ were obtained as in Eq. (7), yielding $\tau_{\text{eff}} \approx 0.849$ as in Eq. (8). Classifiers below τ_{eff} were excluded: KNN (0.760) and SVC (0.752).

Table 5
SHAP-based importance ranking of base-classifiers.

Input label	Base-classifier	Mean absolute SHAP value
X_1	RF	3.9341
X_2	XGB	2.1788
X_3	NGB	1.7787
X_4	CAT	1.6641
X_5	DT	0.4597
X_6	LR	0.4262
X_7	TabNet	0.2054
X_8	NB	0.1735

Higher pairwise values indicate lower diversity. After the effectiveness filter ($\bar{s} \approx 0.9615$), we applied an adaptive diversity filter: the similarity cutoff was set to $\delta \approx 0.97$ (base 0.97 softened by the empirical q_{90} of ρ), with a tiny-gain tolerance $\epsilon = 0.002$. Models are scanned in descending standalone score; a candidate is removed only if it is too similar to any already selected model ($\rho \geq \delta$) and the averaged predictions of the pair yield negligible improvement ($\Delta s \leq \epsilon$) over the better member. Under this rule, LGBM ($\rho = 0.98$ with CAT) and MLP ($\rho \approx 0.97$ with TabNet) were pruned because their pairwise averages improved by at most 0.002 and they had the lower individual scores. Pairs such as RF–XGB or TabNet–CAT showed high similarity but passed the tiny-gain check, so they were retained. This procedure removes near-duplicates while preserving genuinely complementary models.

The remaining eight base-classifiers were incorporated into a cross-trained cascade to produce new training and test datasets. SHAP values were computed using LR as in Eq. (6), yielding the importance ranking. Based on Eq. (8), $S = 11.215$ was obtained, giving $k = 3$ and $\tau_{\text{shap}} = I(3) = 1.7787$. This indicates that the top three base-classifiers already account for 80 % of the decision contribution. The final three base-classifiers were therefore X_1 (RF), X_2 (XGB), and X_3 (NGB), balancing interpretability and model performance (Table 5).

Next, X_1, X_2, and X_3 were fed into the pre-configured GP in Section 3. Fifteen expressions were generated, and the one with the highest validation AUC and the simplest structure was selected. The final

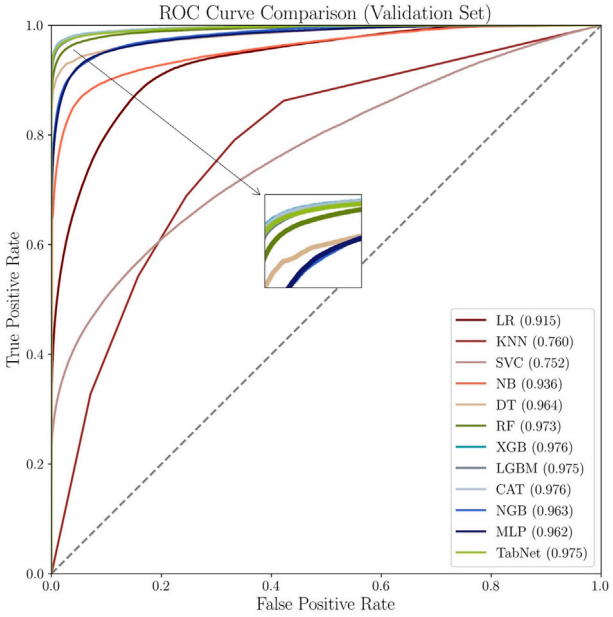

Fig. 4. The ROC curves for all classifiers on validation set.

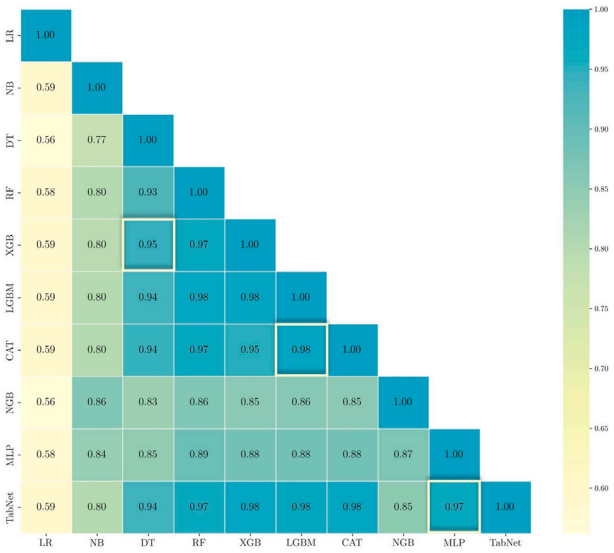

Fig. 5. The heatmap of pairwise diversity. The deeper the color, the lower the diversity between the two classifiers.

expression was

$$g^{\star} = (X_2(X_2 - X_3^2(X_2^2 - 1))).$$

Although three variables were provided to GP, the evolutionary process retained only X_2 and X_3 in the final expression.

The SH-GPC($X_2(X_2 - X_3^2(X_2^2 - 1))$) ensemble yielded excellent results on the independent test set. It achieved an AUC of 0.9963, with Accuracy = 0.9905, Recall = 0.9415, Precision = 0.9968, and F1-score = 0.9684. These metrics confirm the effectiveness of the end-to-end ensemble approach in capturing nearly all positive cases while maintaining very high precision.

4.2.2. Ablation results

The first objective is to examine the effectiveness of EDP protocol. Table 6 presents results for different combinations of base-classifiers, denoted as meta-GP-*n*. For each combination, the mean AUC across all evolved expressions and the complexity of the best expression are recorded. Meta-GP-2 shows slightly higher complexity (tree size 17, maximum depth 6) but achieves almost the same performance (mean AUC 0.9962) as meta-GP-3, which corresponds to the previously selected SH-GPC expression. The results also indicate that including more base classifiers tends to lower the average AUC and increase model complexity. Across all expressions, those involving X_1 and X_3 consistently deliver higher AUC, as shown in Fig. 6. X_2 and X_4 appear most frequently in GP expressions (76 and 66 occurrences), indicating strong substitutability and general utility during construction. X_6 and X_7 are rarely activated, and expressions including X_7 correspond to the lowest AUC, suggesting minimal contribution. X_8 is unused across all iterations, indicating a negligible marginal effect. These findings align with the earlier importance rankings and validate the proposed selection approach. Although X_1 through X_4 share tree-based structures, their strong individual predictive strength ensures significant contribution within stacking when paired with GP. The expression combining X_2 (XGB) and X_3 (NGB) attains the highest AUC and the simplest structure. Using fewer but more informative base classifiers enhances predictive performance and interpretability.

The second objective is to verify the superiority of GP as the meta-classifier. The ablation procedure is repeated with XGB and LR as meta-classifiers. Test set results appear in Table 7 and Fig. 7. The best configuration, meta-GP-3, records the highest AUC, precision, and F1-score, indicating superior discrimination with a balanced error profile. Meta-XGB-7 and meta-XGB-8 achieve the top accuracy and recall, indicating a stronger tendency to capture positives at the expense of precision. Averaged over input sizes, the GP meta-classifier attains the highest mean AUC, exceeding LR by 2.39 % and XGB by 2.58 %. Mean precision improves by 0.98 % over LR and 0.34 % over XGB. Lower overall means for GP arise when many base-classifiers are included, which is also reflected in the larger standard deviations. Given the overall trends in Fig. 7, the focus is on AUC and F1-score, which are central in credit risk prediction. For meta-XGB and meta-LR, AUC and F1-score change little as the number of base-classifiers decreases, suggesting limited ability to identify or prune redundancy. In contrast, meta-GP improves in both metrics as redundant inputs are removed, indicating automatic selection of relevant classifiers and simplification of the ensemble. This explains the decline in recall and thus F1-score for GP in Table 7 when redundant classifiers are included. With suitable pruning at the outset, GP can outperform other meta-classifiers on all five metrics.

In summary, base-classifier ranking and ablation strategies have limited effect with LR and XGB as meta-classifiers. Even with pruning, performance remains similar or deteriorates. Appropriate selection and pruning align with GP's strengths, allowing it to leverage informative classifiers, improve generalization, and reduce model complexity.

4.2.3. Performance comparison and significance tests

This section presents the statistical comparison among the meta-classifiers. As shown in Table 7 in the ablation study, a Friedman test was conducted to statistically assess differences among the three meta-classifiers across the five evaluation metrics. The results indicate statistically significant differences at the 1 % level for all five metrics, as depicted in the radar plot in Fig. 8. The AUC metric exhibited notable differences among classifiers; therefore, pairwise comparisons using DeLong's test were performed. The results indicate that meta-GP significantly outperforms meta-XGB (p-value = 0.0373) and meta-LR (p-value = 0.0064).

A further analysis compares SH-GPC with all individual models to assess whether the proposed framework achieves statistically significant improvements. To ensure robustness and reliability, all 17 models underwent a bootstrap resampling procedure on the test set, repeated 20 times, yielding a total of 340 evaluations across five metrics per evaluation. Table 8 presents the average scores across the 20 bootstrap rounds. SH-GPC achieved the best performance among all 17 models, particularly in AUC, F1-score, and precision.

Table 6
Performance and complexity of simplified GP expressions.

Model ID	Mean AUC	Best Expression	#Variables	Tree Size	Max Depth
meta-GP-2	0.9962	$X_2^2\left(-X_1(X_1+2X_2)(X_2-1)+1\right)$	2	17	6
meta-GP-3	0.9963	$X_2(X_2-X_3^2(X_2^2-1))$	2	13	6
meta-GP-4	0.9864	$X_2^2X_4(-X_2^2X_4^2+2)$	2	19	7
meta-GP-5	0.9818	$X_2\left(-X_2^3X_4^2+X_2+X_4\right)$	3	13	5
meta-GP-6	0.9858	$X_2(-X_1X_2+X_1+X_5+(X_2-X_5)(X_4-X_5))$	4	19	6
meta-GP-7	0.9857	$X_4^2-X_5(X_4-X_5)(2X_1X_2^2+X_4X_7)$	5	23	6
meta-GP-8	0.9891	$X_3^2(-X_3^2X_6+X_3+X_6)+X_6^3(1-X_3^2)$	4	27	7

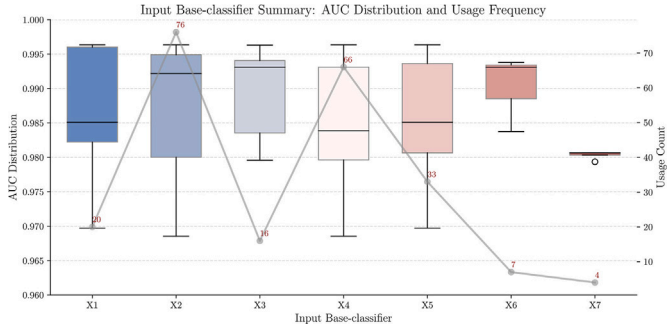

Fig. 6. AUC distributions and usage count with input base-classifiers.

Table 7
Performance comparison under different input sets and meta-classifiers. The values with underline are the best scores for each metric. The values in bold font are the best scores for each classifier.

Input classifiers	meta-classifier	AUC	Accuracy	Recall	Precision	F1 Score
X_1, X_2	LR	0.9707	0.9901	0.9440	0.9823	0.9627
	XGB	0.9690	0.9904	0.9397	0.9888	0.9637
	GP	0.9963	0.9902	0.9449	0.9921	0.9679
X_1, X_2, X_3	LR	0.9709	0.9903	0.9442	0.9833	0.9633
	XGB	0.9692	0.9905	0.9400	0.9890	0.9638
	GP	0.9963	0.9905	0.9415	0.9968	0.9684
X_1, X_2, X_3, X_4	LR	0.9710	0.9903	0.9444	0.9835	0.9636
	XGB	0.9701	0.9906	0.9469	0.9882	0.9645
	GP	0.9931	0.9878	0.9225	0.9966	0.9581
X_1, X_2, X_3, X_4, X_5	LR	0.9711	0.9904	0.9448	0.9833	0.9637
	XGB	0.9690	0.9906	0.9395	0.9902	0.9642
	GP	0.9949	0.9901	0.9317	0.9947	0.9622
$X_1, X_2, X_3, X_4, X_5, X_6$	LR	0.9711	0.9905	0.9445	0.9842	0.9639
	XGB	0.9692	0.9906	0.9397	0.9905	0.9645
	GP	0.9931	0.9879	0.9234	0.9873	0.9543
$X_1, X_2, X_3, X_4, X_5, X_6, X_7$	LR	0.9712	0.9905	0.9449	0.9839	0.9640
	XGB	0.9691	0.9907	0.9395	0.9910	0.9645
	GP	0.9928	0.9785	0.9340	0.9967	0.9643
$X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8$	LR	0.9712	0.9905	0.9449	0.9841	0.9641
	XGB	0.9691	0.9907	0.9495	0.9909	0.9645
	GP	0.9931	0.9891	0.9308	0.9879	0.9585
Mean	LR	0.9710	0.9904	**0.9445**	0.9835	0.9636
	XGB	0.9692	**0.9906**	0.9400	0.9898	**0.9642**
	GP	**0.9942**	0.9877	0.9327	**0.9932**	0.9626
SD	LR	0.0002	0.0001	0.0003	0.0006	0.0004
	XGB	0.0003	0.0001	0.0008	0.0010	0.0003
	GP	**0.0015**	**0.0039**	**0.0103**	**0.0038**	**0.0162**

Subsequently, a Friedman test was conducted based on ranks. It indicates significant performance differences among the models ($\tau_{\chi^2} = 315.061$, $p < 0.00001$). Thus, the Nemenyi post-hoc test was performed, and a Critical Difference (CD) diagram was generated (Fig. 9). Models connected by horizontal lines indicate no statistically significant difference at $\alpha = 0.05$ (note that, although significant differences between meta-GP and the other two meta-classifiers were confirmed by DeLong's test and the Friedman test, these differences may not appear significant here due to the larger number of models, which widens the critical difference). Notably, the proposed meta-GP model (SH-GPC) achieves the highest overall performance, sharing the best rank (2.9) with meta-XGB and significantly outperforming most individual classifiers. Conversely, traditional classifiers such as KNN, LR, SVC, and NB exhibit significantly inferior performance. This further validates the advantage of

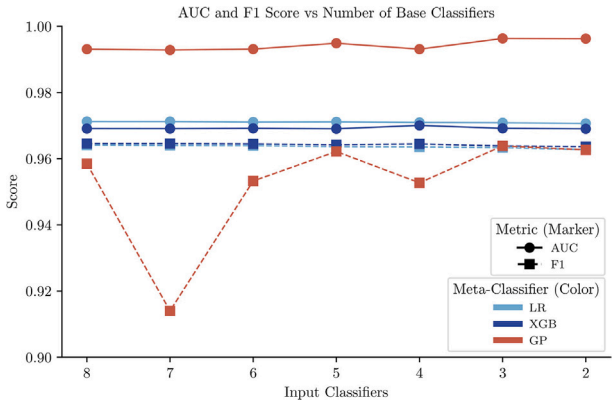

Fig. 7. AUC distributions and selection counts for three meta-classifiers across input base-classifiers.

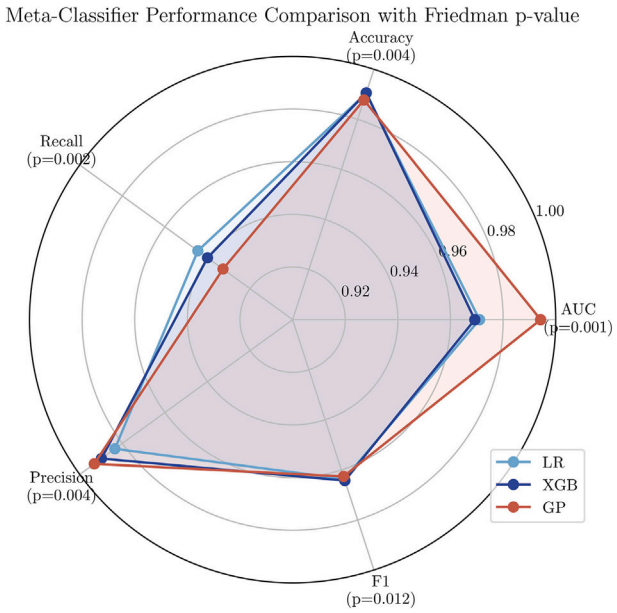

Fig. 8. Comparison among three meta-classifiers on five metrics.

stacking ensemble methods, especially SH-GPC. At comparable performance levels, GP also provides substantially higher interpretability than XGB.

4.3. Interpretation analysis

Model interpretation follows the building order. At Level–1, base classifiers directly interact with the original data; both global (feature ranking) and local (per sample) perspectives are used to explain how two base classifiers distinguish good from bad customers. At Level–2, a representative sample is used to illustrate in detail how GP aggregates the two base classifiers' opinions and produces the final prediction. The feature descriptions are provided in Table 9.

Level–1: global view (X_2 / XGB). From the perspective of X_2, an information-dense SHAP summary covering all important features is presented for global interpretation, displayed as a beeswarm plot (Fig. 10). It illustrates the impact of the top 20 features on predicting credit risk, where the positive class indicates a higher PD. Each dot represents an individual sample, and its position on the horizontal axis indicates the SHAP value, reflecting the influence on PD prediction. The color scale from blue to red corresponds to feature values from low to high. Positive SHAP values (right) indicate an increased likelihood of default; negative values (left) indicate a reduced likelihood. For instance, higher *recoveries* strongly correlate with greater PD, as they typically imply defaults have already occurred. Likewise, higher *logtotal_pymnt* (logarithm of total payments) and *out_prncp* (outstanding principal) correspond to increased PD, suggesting larger loans or higher unpaid balances raise default risk. High *loan_amnt* (loan amount) and *last_pymnt_amnt* (last payment amount) are also associated with higher PD, indicating greater borrowing or larger recent repayments may signal repayment stress. In addition, *last_fico_range_high* (recent high FICO), *grade* (credit grade), *issue_d* (loan issue date), and *total_rec_late_fee* (total late fees) show clear trends: lower grades and higher late fees increase PD, while higher recent FICO reduces PD. The beeswarm plot highlights the key drivers of PD and provides intuitive, business-aligned explanations, enhancing transparency and interpretability.

Level–1: local view (decision paths). Local interpretation is presented as decision plots. Twenty samples are selected, and the decision paths of X_2 for these samples are shown by Fig. 11. The plots depict how the most influential features affect each sample's PD. The horizontal axis represents model outputs in log-odds: positive values indicate higher PD (> 0.5), negative values indicate lower PD (< 0.5). The vertical axis lists features in descending order of mean absolute SHAP values, starting from a baseline that reflects the dataset's average

Table 8
Average Performance Metrics of Models. The values with underline are the best scores for each metric.

Model	AUC	Accuracy	Recall	Precision	F1
LDA	0.9324	0.9401	0.9218	0.7164	0.8062
Probit	0.9470	0.9730	0.9113	0.8917	0.9014
LR	0.8601	0.8536	0.8690	0.4772	0.6161
KNN	0.7220	0.7459	0.6892	0.3053	0.4231
NB	0.9038	0.9475	0.8438	0.7842	0.8129
SVC	0.5502	0.2517	0.9593	0.1486	0.2573
MLP	0.9150	0.9678	0.8426	0.9129	0.8763
DT	0.9470	0.9788	0.9035	0.9373	0.9201
RF	0.9490	0.9846	0.9004	0.9840	0.9403
LGBM	0.9625	0.9886	0.9267	0.9879	0.9563
XGB	0.9685	0.9903	0.9385	0.9893	0.9632
CAT	0.9677	0.9901	0.9370	0.9895	0.9625
TabNet	0.9659	0.9898	0.9330	0.9909	0.9611
NGB	0.9307	0.9606	0.8898	0.8306	0.8592
meta-LR	0.9709	0.9903	0.9444	0.9833	0.9634
meta-XGB	0.9692	0.9905	0.9359	0.9889	0.9618
meta-GP(SH-GPC)	0.9961	0.9902	0.9401	0.9931	0.9662

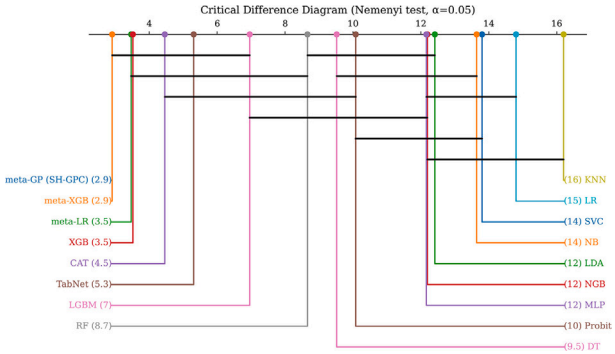

Fig. 9. Critical Difference Diagram showing average ranks of meta-classifiers and base-classifiers. Methods connected by black lines do not differ significantly in performance.

Table 9
Selected features used in the SHAP interpretation.

Feature	Type	Details
loan_amnt	numerical	listed amount of the loan applied
out_prncp	numerical	remaining outstanding principal for total amount funded
total_pymnt	numerical	payments received to date for total amount funded
recoveries	numerical	post chargeoff gross recovery
total_rec_int	numerical	interest received to date
total_rec_late_fee	numerical	late fees received to date
last_pymnt_amnt	numerical	last total payment amount received
last_fico_range_high	numerical	upper boundary of the borrower's last pulled FICO range
fico_range_low	numerical	lower boundary of the borrower's FICO at loan origination
grade	ordinal	LC-assigned loan grade
term	categorical	repayment term (36 or 60 months)
issue_d	categorical	month in which the loan was funded
verification_status	categorical	indicates if income was verified by LC
inq_last_6_mths	numerical	number of credit inquiries in the past six months
annual_inc	numerical	self-reported annual income
emp_length	ordinal	employment length in years
pub_rec	numerical	number of derogatory public records
hardship_flag	categorical	whether the borrower is on a hardship plan
debt_settlement_flag	categorical	whether a charged-off borrower is working with a settlement company
mo_sin_old_rev_tl_op	numerical	months since the oldest revolving account was opened

prediction. The color gradient from blue (low PD) to red (high PD) indicates increasing predicted PD.

In Fig. 11, a purple curve near the decision boundary is particularly notable: this sample shows a delicate balance between positive and negative influences. Despite favorable indicators such as higher cumulative repayment (*logtotal_pymnt*) and a high recent FICO score (*last_fico_range_high*), adverse signals including a larger *loan_amnt* and higher *out_prncp* offset these effects, placing the prediction near the threshold. Samples near the boundary are highly sensitive to small shifts in feature values and threshold adjustments, underscoring the importance of careful handling in practice. This sample is therefore examined further to see how X_2 and X_3 treat it.

For this sample, the final PDs from X_2 and X_3 are 0.446 and 0.804, respectively, indicating a large discrepancy (see Fig. 12). For

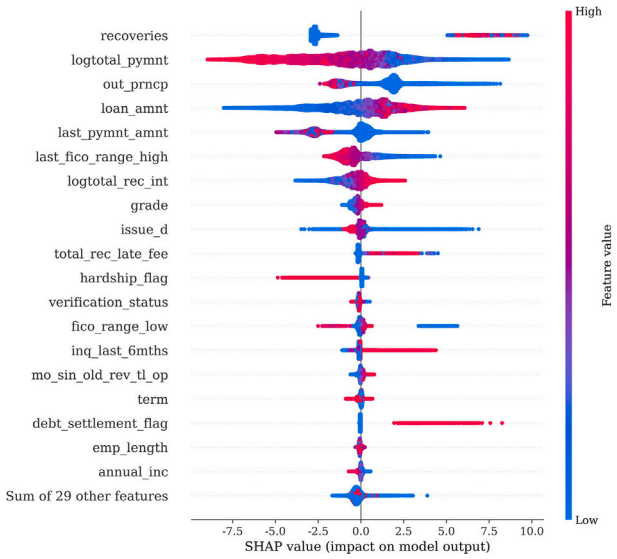

Fig. 10. The beeswarm plot produced by the X_2 (XGB) model. It presents a global ranking of SHAP values for all features by this classifier.

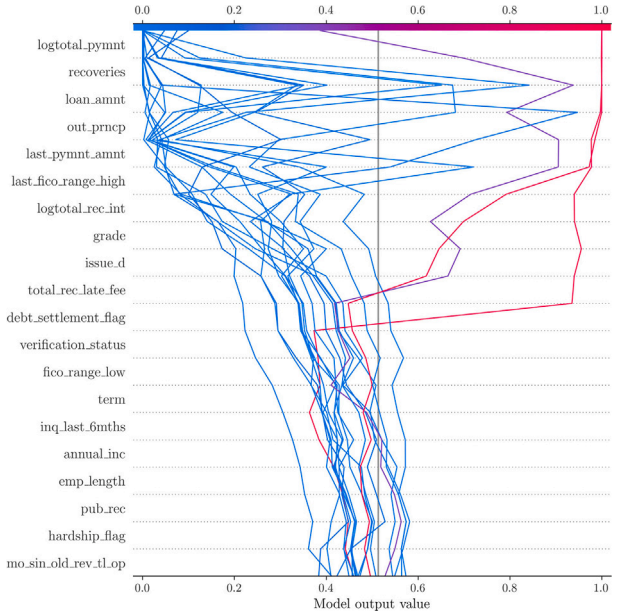

Fig. 11. The decision plot of the first 20 samples produced by the X_2.

X_2, effects are mixed: large exposures (*out_prncp*= 16,110.5, *loan_amnt*= 20,000) push PD up, while repayment signals (*logtotal_pymnt*= 9.209, *last_pymnt_amnt*= 556.37, *logtotal_rec_int*= 8.716) pull it down. Small offsets from *recoveries*= 0, *term*= 2, and *verification_status*= 2 lead to a final probability of 0.446. For X_3, upward forces dominate: a low upper FICO bound (*last_fico_range_high*= 554), no recoveries, and a large last payment (*last_pymnt_amnt*= 556.37) steadily raise the prediction, with little counteracting movement, yielding 0.804. The two decision paths show that identical feature values can produce different net effects across models, explaining the gap between the two probabilities. Determining the final decision thus requires GP at Level–2.

Level–2: expression analysis (GP). At the second level, the mathematical expression generated by GP is analyzed directly, using both syntax-tree and functional perspectives. The expression $X_2\left(X_2-X_3^2(X_2^2-1)\right)$ can be represented as a tree of depth six (Fig. 13), derived by backtracking from the deepest leaf to the root. The root is multiplication, so the output equals the product of the left branch X_2 and the right branch $g(X_2, X_3)$; the root thus acts as a global "gain", scaling the entire output with X_2. Leaves X_2, X_3, and the constant 1 correspond to the two base-classifier probabilities and a baseline. Internal nodes implement interaction (multiplication) and deviation from a reference (subtraction). From the right subtree $X_2 - X_3^2(X_2^2 - 1)$, multiplication at the root yields

$$g(X_2, X_3) = -X_2^3X_3^2 + X_2^2 + X_2X_3^2.$$

Partial derivatives are

$$\frac{\partial g}{\partial X_2} = 2X_2 + X_3^2 - 3X_2^2X_3^2, \qquad \frac{\partial g}{\partial X_3} = 2X_2X_3(1 - X_2^2).$$

When both inputs are high, the first derivative can turn negative, damping over consensus; the second tends to zero or negative as X_2 approaches 1, indicating that X_3's marginal effect weakens or reverses in that regime.

Fig. 12. The decision plot of a sample produced by the X_2 and X_3.

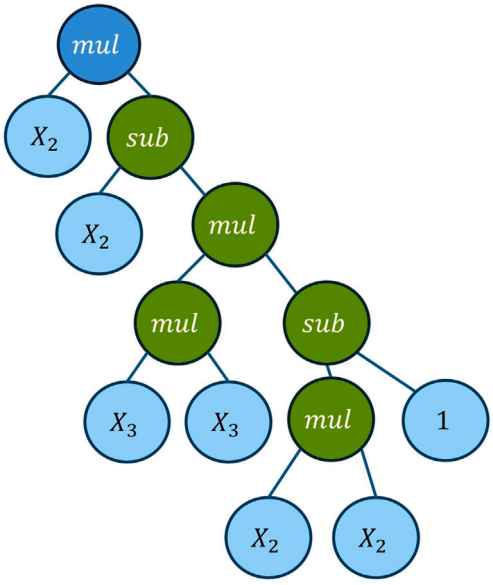

Fig. 13. The syntax tree of SH-GPC($X_2(X_2 - X_3^2(X_2^2 - 1))$).

Sobol sensitivity analysis was performed using Saltelli-style Monte Carlo sampling with 5×10^4 i.i.d. points over $[0, 1]^2$. Based on Eq. (14) and Eq. (15), the estimated indices are $S_{X_2} \approx 0.923$, $S_{X_3} \approx 0.058$, $S_{T,X_2} \approx 0.944$, and $S_{T,X_3} \approx 0.070$. The first-order indices indicate that XGB's main effect explains over 90 % of the variance, whereas NGB's main effect is small (see Fig. 14). The gaps $S_T - S$ reveal interaction strength: $S_{T,X_2} - S_{X_2} \approx 0.021$ and $S_{T,X_3} - S_{X_3} \approx 0.012$, confirming moderate interactions. NGB's influence arises mainly through interaction with XGB, consistent with the negative interaction term $-X_2^3 X_3^2$ and the positive interaction term $X_2 X_3^2$ in $g(X_2, X_3)$. These values align with the derivative-based reading: the root scales by X_2; when both scores are high, the negative interaction suppresses over-optimism; when X_2 is moderate and X_3 is high, the positive interaction provides compensating lift. Replacing the uniform input with the empirical joint distribution in practice would tailor Sobol indices to the data while likely preserving the main conclusion: XGB is the dominant factor, and NGB improves robustness via interaction.

Another lens is to draw function images and inspect the decision boundary (Fig. 15). For $X_2, X_3 \in [0, 1]$, the rule $g > 0.5$ yields a vertical decision boundary at $X_2 = \sqrt{0.5}$, independent of X_3: points to the right are always classified as default, and points to the left as non-default.

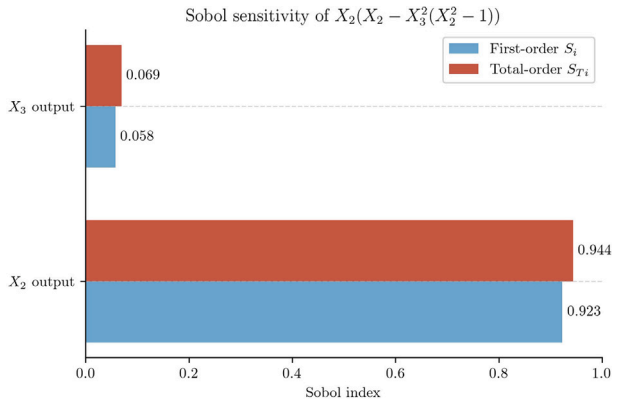

Fig. 14. Sobol sensitivity of SH-GPC($X_2(X_2 - X_3^2(X_2^2 - 1))$).

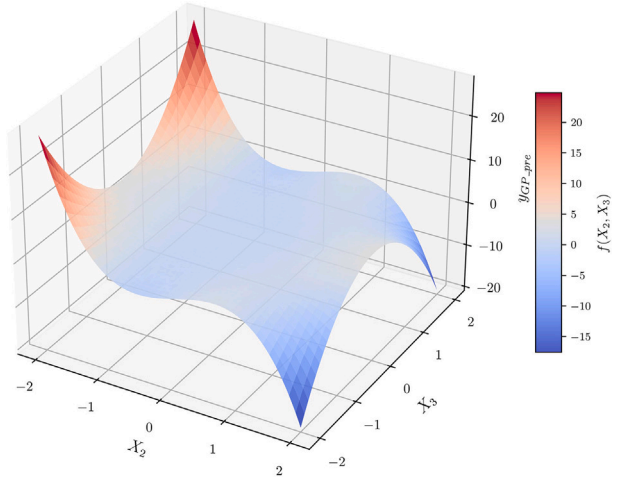

Fig. 15. The surface plot of SH-GPC($X_2(X_2 - X_3^2(X_2^2 - 1))$).

This boundary is simple, fixed, and entirely controlled by X_2, making the classification rule easy to interpret and visualize.

Finally, the explanations of both first and second levels can be presented to obtain a final prediction from the stacking ensemble. Using again the example of the previously analyzed sample represented by the purple curve, we have the outputs from the two base classifiers: $X_2 = 0.446$ (XGB) and $X_3 = 0.804$ (NGB). If relying solely on these base classifiers individually, the decision regarding default or non-default would remain ambiguous. However, by inputting these results into the Level–2 GP classifier, the final prediction can be computed as Eq. (21).

$$g(X_2, X_3) = X_2(X_2 - X_3^2(X_2^2 - 1)) = 0.446\,(0.446 - 0.804^2(0.446^2 - 1)) \approx 0.4299. \tag{21}$$

Given that this value is below the threshold of 0.5, the ensemble ultimately classifies this borrower as non-default, consistent with the actual observed outcome. This clearly demonstrates how the GP meta-model effectively corrects the incorrect prediction initially made by classifier X_3 by integrating information from classifier X_2, moving the final prediction closer to the correct decision, albeit still relatively near the decision boundary.

5. Conclusion and future work

In critical domains such as financial risk prediction, achieving 100 % accuracy is impractical due to the infinite variability of real-world scenarios. Even a 0.1 % deviation can result in substantial financial losses for a company. To improve the performance of a stacking model, this study introduces SH-GPC, an advanced framework based on heterogeneous base-classifiers. The key conclusions of SH-GPC include proposing new principles and strategies for selecting base-classifiers and demonstrating the effectiveness of GP as a meta-classifier in improving both predictive performance and interpretability. Experimental results indicate that SH-GPC significantly enhances predictive accuracy, especially in terms of AUC and precision, reflecting high discrimination capability, robustness, and a low false-positive rate.

Specifically, this paper addresses three key questions posed at the outset. The first concerns selecting appropriate base classifiers for a heterogeneous ensemble. The proposed EDP framework integrates three principles: effectiveness, diversity, and pruning. Effectiveness ensures that only models with adequate predictive strength are retained. Diversity avoids excessive redundancy while preserving complementary classifiers. Pruning removes models that contribute little unique information. This balanced process prevents the inclusion of weak yet diverse classifiers and avoids over-reliance on a single strong model. Experimental results show that ensembles built from robust and distinct classifiers such as RF, XGB, and NGB achieve superior predictive power compared to those combining many weak classifiers.

The second question is how to select meta-classifiers in stacking ensembles. Traditionally, stable classifiers like logistic regression or strong classifiers such as XGBoost are frequently chosen. However, these classifiers may not yield substantial performance gains and could increase model complexity. Thus, a meta-classifier balancing accuracy, stability,

and interpretability is more desirable. The experiments confirm that GP meets these criteria effectively.

The third question concerns interpretability and its balance with accuracy. In SH-GPC, genetic programming outputs explicit symbolic expressions, so the second layer is transparent. Model complexity is strictly controlled through hard budgets on tree size and depth, restricted operator sets, and mutation settings. These controls quantify parsimony and turn the accuracy versus interpretability trade off into a measurable curve. In parallel, SHAP provides global and local attributions for each base classifier, independent of learner type, which mitigates common trust concerns in stacking. Together, these mechanisms make the trade off explicit and allow practitioners to set a desired level of transparency.

Overall, the SH-GPC pipeline is not confined to credit risk and can be applied to a broad range of tabular prediction problems. Its pool of base learners is modular and extensible: practitioners may assemble task-appropriate sets, for example combining NGB and TabNet to capture predictive uncertainty and rich feature interactions, or substituting calibrated linear models, monotonic boosting, kernel methods, and domain-specific networks when the application requires. EDP screening and GP expression selection are governed by explicit formulae; all thresholds and selection rules are exposed as parameters in code, which allows straightforward retuning on new datasets without redesigning the procedure. As a result, SH-GPC offers a reusable and transparent construction of stacking that balances accuracy, stability, and interpretability and transfers to related problems with minimal changes. The full implementation is publicly available on GitHub[2]).

Based on the findings presented in this paper, several promising research directions can be identified. First, future studies should continue to explore the trade-off between interpretability and model performance by quantifying this relationship explicitly. For example, future work could investigate establishing precise quantitative metrics, such as determining exactly how much predictive performance (e.g., measured by AUC) improves with each incremental increase in model complexity. Second, there is potential for integrating emerging modeling techniques, particularly deep learning methods, into stacking frameworks to further enhance predictive accuracy and generalization capabilities. Investigating how advanced neural network structures (such as transformers or attention-based architectures) perform as base classifiers or meta-learners in stacking could provide valuable insights. Lastly, future research could explore incorporating time-series ensemble models specifically designed for credit risk prediction tasks. Integrating temporal dependencies through methods such as recurrent neural networks, temporal convolutional networks, or other advanced sequence modeling approaches could improve predictions by effectively capturing dynamic risk patterns over time.

CRediT authorship contribution statement

Zixue Zhao: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Qiao Lin:** Software, Methodology, Investigation. **Yiran Li:** Writing – original draft, Methodology, Investigation, Formal analysis. **Yue Li:** Supervision, Resources. **Tianxiang Cui:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

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

[2] https://github.com/Jasmine1937/sh_gpc

Acknowledgment

This work is supported in part by Ningbo Natural Science Foundation, China (Project ID 2023J194), in part by the Ningbo Government, China (Project ID 2021B-008-C), and in part by the Basic and Commonweal Programme of Zhejiang Natural Science Foundation (Project ID LY24F020006).

Data availability

Data will be made available on request.

References

[1] J. Abellán, J.G. Castellano, A comparative study on base classifiers in ensemble methods for credit scoring, Expert Syst. Appl. 73 (2017) 1–10.
[2] M. Ala'raj, M.F. Abbod, Classifiers consensus system approach for credit scoring, Knowl.-Based Syst. 104 (2016a) 89–105.
[3] M. Ala'raj, M.F. Abbod, A new hybrid ensemble credit scoring model based on classifiers consensus system approach, Expert Syst. Appl. 64 (2016b) 36–55.
[4] G. Amit, M. Mahesh, Financial fraud detection using naive Bayes algorithm in highly imbalance data set, J. Discrete Math. Sci. Cryptogr. 24 (2021) 1559–1572.
[5] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, C. Rudin, Learning certifiably optimal rule lists for categorical data, J. Mach. Learn. Res. 18 (2017).
[6] S.Ö. Arik, T. Pfister, Tabnet: attentive interpretable tabular learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI, 2021, pp. 6679–6687.
[7] B. Baesens, K. Smedts, Boosting credit risk models, Br. Account. Rev. (2023) 101241.
[8] I. Bakurov, M. Buzzelli, R. Schettini, M. Castelli, L. Vanneschi, Semantic segmentation network stacking with genetic programming, Genet. Program. Evolvable Mach. 24 (2023) 15.
[9] Y. Bian, H. Chen, When does diversity help generalization in classification ensembles? IEEE Trans. Cybern. 52 (2021) 9059–9075.
[10] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.
[11] L. Breiman, J.H. Friedman, O.R.C. Classification and Regression Trees, Routledge, 1984.
[12] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, J.K. Su, This looks like that: deep learning for interpretable image recognition, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2019, pp. 8928–8939.
[13] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, 2016, pp. 785–794.
[14] A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi, A. Imine, Credit card fraud detection in the ERA of disruptive technologies: a systematic review, J. King Saud Univ. - Comput. Inf. Sci. 35 (2023) 145–174.
[15] S. Cui, H. Qiu, S. Wang, Y. Wang, Two-stage stacking heterogeneous ensemble learning method for gasoline octane number loss prediction, Appl. Soft Comput. 113 (2021) 107989.
[16] Q. Dai, R. Ye, Z. Liu, Considering diversity and accuracy simultaneously for ensemble pruning, Appl. Soft Comput. 58 (2017).
[17] E.A. Daoud, Comparison between Xgboost, Lightgbm and Catboost using a home credit dataset, Int. J. Comput. Inf. Eng. 145 (2019) 6–10.
[18] X. Dastile, T. Celik, M. Potsane, Statistical and machine learning models in credit scoring: a systematic literature survey, Appl. Soft Comput. 91 (2020) 106263.
[19] H. Deleglise, R. Interdonato, A. Bégué, E. Maître d'Hôtel, M. Teisseire, M. Roche, Food security prediction from heterogeneous data combining machine and deep learning methods, Expert Syst. Appl. 190 (2021) 116–189.
[20] E.F. DeLong, Everything in moderation: archaea as 'non-extremophiles, Curr. Opin. Genet. Dev. 8 (1998) 649–654.
[21] S. Ding, T. Cui, A.G. Bellotti, M.Z. Abedin, B. Lucey, The role of feature importance in predicting corporate financial distress in pre and post covid periods: evidence from China, Int. Rev. Financ. Anal. 90 (2023) 102851.
[22] S. Ding, T. Cui, A.M. Du, J.W. Goodell, N. Du, Disentangling and hedging global warming risk: a machine learning approach, Environ. Impact Assess. Rev. 115 (2025a) 107987.
[23] S. Ding, X. Wu, T. Cui, J.W. Goodell, A.M. Du, Modeling climate policy uncertainty into cryptocurrency volatilities, Int. Rev. Financ. Anal. 102 (2025b) 104030.
[24] T. Duan, A. Avati, D.Y. Ding, P. Kohlmann, A. Ng, P. Schulam, Ngboost: natural gradient boosting for probabilistic prediction, in: Proceedings of the 37th International Conference on Machine Learning, PMLR, 2020, pp. 2690–2700.
[25] E. Dumitrescu, S. Hué, C. Hurlin, S. Tokpavi, Machine learning for credit scoring: improving logistic regression with non-linear decision-tree effects, Eur. J. Oper. Res. 297 (2022) 1178–1192.
[26] Y. Feng, F. Yan, A two-stage stacked-based heterogeneous ensemble learning for cancer survival prediction, Complex Intell. Syst. 8 (2022) 4619–4639.
[27] V. García, A.I. Marqués, J.S. Sánchez, An insight into the experimental design for credit risk and corporate bankruptcy prediction systems, J. Intell. Inf. Syst. 44 (2015) 159–189.
[28] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (2018) 93.
[29] J.A. Hanley, B.J. McNeil, The meaning and use of the area under a receiver operating characteristic curve, Radiology 143 (1982) 29–36.

[30] R. Henckaerts, K. Antonio, M.P. Côté, When stakes are high: balancing accuracy and transparency with model-agnostic interpretable data-driven surrogates, Expert Syst. Appl. 202 (2022) 117230.
[31] N.C. Hsieh, L.P. Hung, A data driven ensemble classifier for credit scoring analysis, Expert Syst. Appl. 37 (2010) 534–545.
[32] S.C. Huang, C.F. Wu, C.C. Chiou, M.C. Lin, Intelligent fintech data mining by advanced deep learning approaches, Comput. Econ. 59 (2021) 1407–2422.
[33] C. Hung, J.H. Chen, A selective ensemble based on expected probabilities for bankruptcy prediction, Expert Syst. Appl. 36 (2009) 5297–5303.
[34] W.N. Ismail, H.A. Alsalamah, E.A. Mohamed, Ga-stacking: a new stacking-based ensemble learning method to forecast the Covid-19 outbreak, Comput. Mater. Contin. 74 (2022) 3945–3976.
[35] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.Y. Liu, Lightgbm: a highly efficient gradient boosting decision tree, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2017.
[36] S. Lessmann, B. Baesens, H.V. Seow, L.C. Thomas, Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research, Eur. J. Oper. Res. 247 (2015) 124–136.
[37] W. Li, S. Ding, Y. Chen, S. Yang, Heterogeneous ensemble for default prediction of peer-to-peer lending in China, IEEE Access 6 (2018) 54396–54406.
[38] W. Li, S. Ding, H. Wang, Y. Chen, S. Yang, Heterogeneous ensemble learning with feature engineering for default prediction in peer-to-peer lending in China, World Wide Web 23 (2020) 23–45.
[39] W. Liu, H. Fan, M. Xia, Tree-based heterogeneous cascade ensemble model for credit scoring, Int. J. Forecast. 39 (2023) 1593–1614.
[40] S.M. Lundberg, G. Erion, H. Chen, A. DeGrave, J.M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.I. Lee, From local explanations to global understanding with explainable AI for trees, Nat. Mach. Intell. 2 (2020) 2522–5839.
[41] S.M. Lundberg, S.I. Lee, A unified approach to interpreting model predictions, In I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 4765–4774.
[42] A. Markov, Z. Seleznyova, V. Lapshin, Credit scoring methods: latest trends and points to consider, J. Finance Data Sci. 8 (2022) 180–201.
[43] C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, Lulu Enterprises, 2020.
[44] V. Moscato, A. Picariello, G. Sperlí, A benchmark of machine learning approaches for credit score prediction, Expert Syst. Appl. 165 (2021) 113986.
[45] P.B. Nemenyi, Distribution-Free Multiple Comparisons (Ph.D. thesis). Princeton University., 1963.
[46] H.H. Nguyen, J. Viviani, S.B. Jabeur, Bankruptcy prediction using machine learning and Shapley additive explanations, Rev. Quant. Finance Account. (2023).
[47] N.V. Chawla, K.W. Bowyer, L. Hall, Smote: synthetic minority over-sampling technique, Artif. Intell. 16 (2002) 321–357.
[48] C. O'Neil, Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy, Crown Publishing Group, New York, United States, 2016.
[49] D.K. Padhi, N. Padhy, A.K. Bhoi, J. Shafi, M.F. Ijaz, A fusion framework for forecasting financial market direction using enhanced ensemble models and technical indicators, Mathematics 9 (2021).
[50] R. Poli, W.B. Langdon, N.F. McPhee, A Field Guide to Genetic Programming, Lulu Enterprises, UK Ltd, 2008.
[51] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, A. Gulin, Catboost: unbiased boosting with categorical features, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2018.
[52] S.S.C.G. Ribeiro, "why should i trust you?": explaining the predictions of any classifier, in: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[53] M.T. Ribeiro, S. Singh, C. Guestrin, Anchors: high-precision model-agnostic explanations, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI Press, 2018, pp. 1527–1535.
[54] C. Rudin, J. Radin, Why are we using black box models in AI when we don't need to? a lesson from an explainable AI competition, Harv. Data Sci. Rev. 1 (2019) 3–7.
[55] G. Sagastabeitia, J. Doncel, J. Aguilar, A.F. Anta, J. Ramirez, Covid-19 seroprevalence estimation and forecasting in the USA from ensemble machine learning models using a stacking strategy, Expert Syst. Appl. 258 (2024) 124930.
[56] T. Shi, L. Meng, L. Deng, J. Li, Explainable models for predicting crab weight based on genetic programming, Ecol. Inform. 88 (2025) 103131.
[57] Y. Takefuji, Beyond Xgboost and SHAP: unveiling true feature importance, J. Hazard. Mater. 488 (2025) 137382.
[58] I. Tomek, Two modifications of CNN, IEEE Trans. Syst. Man Cybern. SMC-6 (1976) 769–772.
[59] E. Tosetti, V. Vinciotti, A computationally efficient correlated mixed probit model for credit risk inference, J. R. Stat. Soc. Ser. C Appl. Stat. 68 (2019) 1183–1204.
[60] C.F. Tsai, Feature selection in bankruptcy prediction, Knowl.-Based Syst. 22 (2009) 120–127.
[61] S. Wang, X. Zhang, Research on credit default prediction model based on tabnet-stacking, Entropy 26 (2024) 861.
[62] D. Wood, T. Mu, A. Webb, H. Reeve, M. Luján, G. Brown, A unified theory of diversity in ensemble learning, J. Mach. Learn. Res. 24 (2023) 1–49.
[63] Y. Xia, C. Liu, B. Da, F. Xie, A novel heterogeneous ensemble credit scoring model based on bstacking approach, Expert Syst. Appl. 93 (2018) 182–199.
[64] D. Yang, B. Xiao, M. Cao, H. Shen, A new hybrid credit scoring ensemble model with feature enhancement and soft voting weight optimization, Expert Syst. Appl. 238 (2024) 122101.
[65] W. Yin, B. Kirkulak-Uludag, D. Zhu, Z. Zhou, Stacking ensemble method for personal credit risk assessment in peer-to-peer lending, Appl. Soft Comput. 142 (2023) 110302.
[66] W. Zhang, D. Yang, S. Zhang, A new hybrid ensemble model with voting-based outlier detection and balanced sampling for credit scoring, Expert Syst. Appl. 174 (2021) 114744.
[67] X. Zhang, L. Yu, Consumer credit risk assessment: a review from the state-of-the-art classification algorithms, data traits, and learning methods, Expert Syst. Appl. 237 (2024) 121484.
[68] G.Y. Zhao, K. Ohsu, H. Kasmanhadi Saputra, T. Okada, J. Suzuki, Y. Kuwahara, M. Fujita, Enhancing interpretability of tree-based models for downstream salinity prediction: decomposing feature importance using the Shapley additive explanation approach, Results In Engineering 23 (2024a) 102373.
[69] Z. Zhao, T. Cui, S. Ding, J. Li, A.G. Bellotti, Resampling techniques study on class imbalance problem in credit risk prediction, Mathematics 12 (2024b).
[70] Y. Zhen, X. Zhu, An ensemble learning approach based on tabnet and machine learning models for cheating detection in educational tests, Educ. Psychol. Meas. 84 4 (2023) 780–809.
[71] L. Zhou, H. Fujita, H. Ding, R. Ma, Credit risk modeling on data with two timestamps in peer-to-peer lending by gradient boosting, Appl. Soft Comput. 110 (2021) 107672.
[72] Z.H. Zhou, Ensemble Methods: Foundations and Algorithms, Chapman and Hall/CRC, New York, 2012.
[73] Y. Zhu, Y. Hu, Q. Liu, H. Liu, C. Ma, J. Yin, A hybrid approach for predicting corporate financial risk: integrating Smote-Enn and Ngboost, IEEE Access 11 (2023) 111106–111125.