International Journal of Forecasting 39 (2023) 1593–1614

Contents lists available at ScienceDirect

International Journal of Forecasting

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

Tree-based heterogeneous cascade ensemble model for credit scoring☆

Wanan Liu [a], Hong Fan [a,*], Meng Xia [b]

[a] Glorious Sun School of Business and Management, Donghua University, Shanghai 200051, China
[b] College of Information Science and Technology, Donghua University, Shanghai 201620, China

ARTICLE INFO

Keywords:
Credit scoring
Ensemble algorithm
Heterogeneous deep forest
Weighted voting mechanism
Interpretability

ABSTRACT

Credit scoring is an important tool to guard against commercial risks for banks and lending companies and provides good conditions for the construction of individual personal credit. Ensemble algorithms have shown appealing progress for the improvement of credit scoring. In this study, to meet the challenge of large-scale credit scoring, we propose a heterogeneous deep forest model (Heter-DF), which is established based on considerations ranging from base learner selection, encouragement of the diversity of base learners, and ensemble strategies, for credit scoring. Heter-DF is designed as a scalable cascading framework that can increase its complexity with the scale of the credit dataset. Moreover, each level of Heter-DF is built by multiple heterogeneous tree-based ensembled base learners, avoiding the homogeneous prediction of the ensemble framework. In addition, a weighted voting mechanism is introduced to highlight important information and suppress irrelevant features, making Heter-DF a robust model for credit scoring. Experimental results on four credit scoring datasets and six evaluation metrics show that the cascading framework a good choice for the ensemble of tree-based base learners. A comparison among homogeneous ensembles and heterogeneous ensembles further demonstrates the effectiveness of Heter-DF. Experiments on different training sets indicate that Heter-DF is a scalable framework which not only deals with large-scale credit scoring but also satisfies the condition where small-scale credit scoring is desirable. Finally, based on the good interpretability of a tree-based structure, the global interpretation of Heter-DF is preliminarily explored.

1. Introduction

With the rapid growth of the financial market and the fast transformation of the public consumption concept, as well as the continuous enrichment of credit products, credit risk has become a critical and dynamic risk to commercial banks. Therefore, effectively reducing the risk of default is a major concern for financial risk management. The update of the Basel Capital Accord II (Huang & Thomas, 2015) provided guidelines for banks to forecast the probability of default (PD) based on historical information by establishing an internal rating method (Xia, Zhao, He, Li, & Yang, 2021). As one of the major concerns influencing credit risk, accurate estimations of the PD have become the basis of credit risk management. Credit scoring has been recognized as a powerful tool for financial institutions to learn from the worldwide financial crisis.

Banks determine whether to issue a loan, the loan amount, and the transaction strategy by assessing the credit scores of applicants to mitigate credit risks. With

☆ This document is the result of a research project funded by the National Natural Science Foundation of China (71971054), and the Natural Science Foundation of Shanghai, China (19ZR1402100).
* Corresponding author.
E-mail addresses: liuwanan@hotmail.com (W. Liu), hongfan@dhu.edu.cn (H. Fan), shom@mail.dhu.edu.cn (M. Xia).

https://doi.org/10.1016/j.ijforecast.2022.07.007
0169-2070/

the development of computational resources and the advancement of artificial intelligence (AI) theories, the development of credit scoring models has progressed from statistical-based credit scoring algorithms to intelligent machine learning (ML)-based credit scoring approaches. To achieve high-performance identification of good and bad applicants, great effort has been devoted to developing ML-based credit scoring algorithms.

Linear discriminant analysis (LDA) (Altman, 1968; Lee, Chiu, Lu, & Chen, 2002) and logistic regression (LR) (Bahnsen, Aouada, & Ottersten, 2014; Sohn, Kim, & Yoon, 2016) are the two dominant models for early-stage credit scoring and have been widely adopted. While risk managers are interested in statistical credit scoring due to its simplification and credibility, conventional statistical-based approaches are driven by the belief that credit datasets are linearly separable. Therefore, many researchers focus on intelligent ML-based credit score scoring models. Maldonado, Pérez, and Bravo (2017) modeled credit scoring as a reject inference process, and applied a support vector machine (SVM) to minimize the credit scoring error. Bequé and Lessmann (2017) comprehensively balanced the efficiency and accuracy of credit scoring algorithms and developed an efficient neural network (NN) algorithm, called an extreme learning machine, to improve the efficiency and performance of credit scoring. Sohn and Kim (2012) employed a decision tree (DT), which can be regarded as an alternative to complex credit scoring models, for the credit scoring of start-up firms. Many effective and intelligent credit scoring models have been developed, including NNs (Tsai & Wu, 2008; West, 2000), *k*-nearest neighbors (KNNs) (Henley et al., 1997), SVMs, (Huang, Chen, & Wang, 2007; Maldonado, Bravo, López, & Pérez, 2017), and naive Bayesian (NB) models (Hu & Ansell, 2007).

According to the "no free lunch" (Wolpert & Macready, 1997) theorem, a single ML-based credit scoring model is not optimal for complex nonlinear credit datasets from different banks, due to significant variations in the size of data, reported variable details, and data structure. A good solution for addressing this limitation is to integrate multiple weak learners into a stronger one for accurate credit scoring, which is the key to ensemble learning-based credit scoring.

A variety of literature has witnessed the significant improvement of ensemble credit scoring models compared with individual "weak" learners. To achieve effective integration, individual learners are encouraged to be accurate and diverse in ensemble models. Here, "accurate" is related to the robustness of base learners, and "diverse" implies that one base learner should be different from another in the training pattern and parameter settings. According to the distinction among base learners on training patterns and parameter settings, ensemble credit scoring approaches can be divided into homogeneous ensemble and heterogeneous ensemble methods. A homogeneous ensemble consists of individuals having a single-type base learning algorithm, while a heterogeneous ensemble is a group of algorithms that are ensembled by base learners that have different training patterns.

Abellán and Castellano (2017) focused on the selection and optimization of the base learner of ensemble credit scoring methods. The experimental results show that base learner selection is a vital process for ensemble credit scoring models. Moreover, their study revealed that the credal decision tree is the optimal base learner for the model ensemble. He, Zhang, and Zhang (2018) solved the imbalanced credit scoring problem with a cascade cost-sensitive boosted tree. In their study, a weighted extreme gradient boosting (XGBoost) is introduced as the base learner to enhance the robustness of the ensemble model, and a particle swarm optimization (PSO) algorithm is employed to finetune the hyperparameters of the cascade framework. Xiao, Xiao, and Wang (2016) integrated ensembling and clustering techniques to enrich the diversity of base learners, thereby driving the ensembled model to make diversified prediction results. Furthermore, a weighted voting mechanism is borrowed to integrate the prediction results of the base learners. Abellán and Mantas (2014) improved the bagging strategy and integrated multiple DTs for bankruptcy prediction and credit scoring. The results show that bagging-like ensemble algorithms are good choices to encourage the diversity of base learners to further reduce the credit scoring error. Wang, Ma, Huang, and Xu (2012) designed two dual ensemble algorithms – random space (RS)-bagging DT, and bagging-RS DT – to deal with noisy credit scoring datasets. Their findings showed that the proposed dual ensemble tree can be used for credit scoring as an alternative to random forests (RF), bagging DTs, and rotation forests. Feng, Xiao, Zhong, Dong, and Qiu (2019) introduced a dynamic Markov chain to model the classification ability of each base learner, and computed different weights for the combination of the base learner, making the ensemble approach an adaptive model for credit scoring. A boosted DT is proposed for credit scoring in Xia, Liu, Li, and Liu (2017)'s work, and the hyperparameters of the boosted DT are finetuned based on a Bayesian optimization technique. Furthermore, the interpretability of the proposed boosted decision tree is preliminarily explored. Many other ensemble learning studies have contributed to improving the robustness of credit scoring. These works include Liu, Fan, and Xia (2021), Niu, Zhang, Liu, and Li (2020), Shen, Zhao, Li, Li, and Meng (2019).

The above ensemble algorithms enhance the robustness of credit scoring models from the perspective of the heterogeneous ensemble. However, according to the conception of ensemble learning, the base learners should be robust and different. The same type of classification algorithm has a consistent bias in the prediction of credit risk, which reduces the diversity of the output of the base learner and makes it incapable of dealing with complex credit scoring datasets. In contrast, heterogeneous ensemble approaches integrate different base classifiers, which enables diversified predictions and enhances their adaptability to different credit datasets.

Lessmann, Baesens, Seow, and Thomas (2015) updated Baesens et al. (2003)'s work and explored PD modeling on eight real credit datasets and 41 credit scoring models. They found that heterogeneous ensemble algorithms are better at predicting credit risk than homogeneous ensemble methods, but that both outperform industry-standard LR. Nalić, Martinović, and Žagar (2020) focused

on the optimization of the combination mechanism of heterogeneous base learners and the feature selection process. In their study, the advantages and disadvantages of five feature selection strategies are compared and discussed. Further, four base learners (a generalized linear model (GLM), DT, SVM, and NB) are ensembled, and a new voting mechanism, named if-any, is proposed to integrate the output of base learners. Papouskova and Hajek (2019) proposed a two-stage heterogeneous ensemble approach for PD and exposure-at-default (EAD) modeling. An undersampling strategy is introduced in the first stage to improve the performance of predicting PD. Their results support the conclusion that heterogeneous ensemble approaches are an effective solution for predicting the credit risk of consumers. Xia, Liu, Da, and Xie (2018) designed a bagging-stacking heterogeneous framework for credit scoring. A Gaussian process classifier (GPA), SVM, RF, and XGBoost are first trained with a bagging strategy that selectively determines the optimal base classifier. Then, a stacking ensemble is considered to further integrate the optimal meta-classifier into a highly integrated framework to improve the performance of credit scoring. Feng, Xiao, Zhong, Qiu, and Dong (2018) proposed a dynamic heterogeneous ensemble strategy which takes into account the different misclassification costs between good applicants and bad applicants, and calculated the soft probability for the selection and combination of base learners. The comparison results on 10 datasets and seven evaluation metrics showed the stronger predictive ability of the proposed method. To improve the performance of credit scoring, many other heterogeneous ensemble models have been developed. These studies include De Bock, Coussement, and Lessmann (2020), Xia, Zhao, He, Li, and Niu (2020), Zhou, Li, Wang, Ding, and Xia (2019).

The key to the success of ensemble learning depends on the following three aspects: the robustness of the base learners, the diversity of the base learners, and the ensemble strategy. Based on the above considerations, this study is motivated by the following questions:

(1) Existing ensemble credit scoring models focus on the ensemble of individual weak learners. Ensemble models are robust credit scoring approaches compared with simple classifiers. Would the credit scoring performance be enhanced by further integrating advanced ensembled classifiers?
(2) Heterogeneous ensembles enhance the performance of credit scoring by integrating different types of base learners. However, weak learners for heterogeneous ensembles harm the performance of credit scoring. How can this be avoided?
(3) How can we effectively integrate heterogeneous ensembled base learners to gain a better credit scoring performance?

To answer these questions, we propose a tree-based heterogeneous cascade ensemble model (heterogeneous deep forest, Heter-DF) for credit scoring. First, we consider the robust tree-based ensemble algorithm as the base learner, making Heter-DF an "ensemble in ensemble" credit scoring algorithm. Secondly, three different tree-based ensemble models are employed for heterogeneous ensembling, which increases the diversity of base learners. Thirdly, a weighted ensemble mechanism is introduced to enhance the prediction results of stronger base learners. Finally, a cascade framework is borrowed to further improve the performance of ensemble learning-based credit scoring.

2. Heterogeneous deep forest-based credit scoring

2.1. Problem formulation

Credit scoring is a crucial tool for banks and lending institutions when determining whether or not to issue loans to prospective borrowers. Such decisions often have a considerable impact on the financial situation of borrowers. The update of the Basel Accord further highlights the importance of credit scoring for the credit risk management of banks and other financial organizations. Credit scoring seeks to forecast a borrower's probability of default (PD) by taking into account crucial factors such as repayment history, loan type, credit history, total debt, and the external economic climate. The estimated PD is further utilized to establish whether a person has satisfied the conditions for receiving credit. As a result, the credit scoring method is often considered a PD predictor. Generally, an ML-based credit scoring system that serves to discriminate between risky loans and non-risky loans can be modeled as a binary classification problem. Suppose we represent the credit features of each credit sample as a vector $\mathbf{x} = (x_1, x_2, \ldots, x_M)^T$ with M financial variables that cover a borrower's credit information and loan attributes as well as external economic conditions. We define $y_i \in \{0, 1\}$ as the actual state that denotes whether the loan is timely paid, where 0 represents a loan payment cycle that has been complete, while 1 denotes that a loan is at risk of defaulting in its payment cycle.

Given a training credit set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, N is the number of credit samples in a training set. An ML-based credit scoring realizes the PD prediction by finding a powerful predictor that minimizes the empirical risk:

$$F^* = \underset{F}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, F(\mathbf{x}_i)), \tag{1}$$

where $\mathcal{L}$ is the loss function. Next, a translation of the PD into a decision on whether to issue a loan to a borrower is performed by a probability threshold.

$$\hat{y}_i = sgn(F(\mathbf{x}_i)) = \begin{cases} 0, & F(\mathbf{x}_i) < \tau \\ 1, & F(\mathbf{x}_i) > \tau \end{cases}, \tag{2}$$

where $sgn(\cdot)$ is a threshold function, and τ is a probability threshold that transforms the PD into the state of a loan sample. In this study, since imbalanced credit scoring is not involved, we follow a general case and set $\tau = 0.5$.

To enhance the predictive performance of credit scoring, ensemble learning algorithms combine multiple base learners into a robust hypothesis, realizing accurate credit scoring by optimizing the predictive bias and variance. According to the integration strategy, ensemble approaches can be divided into three groups: bagging-based ensemble methods, boosting ensemble algorithms, and stacking

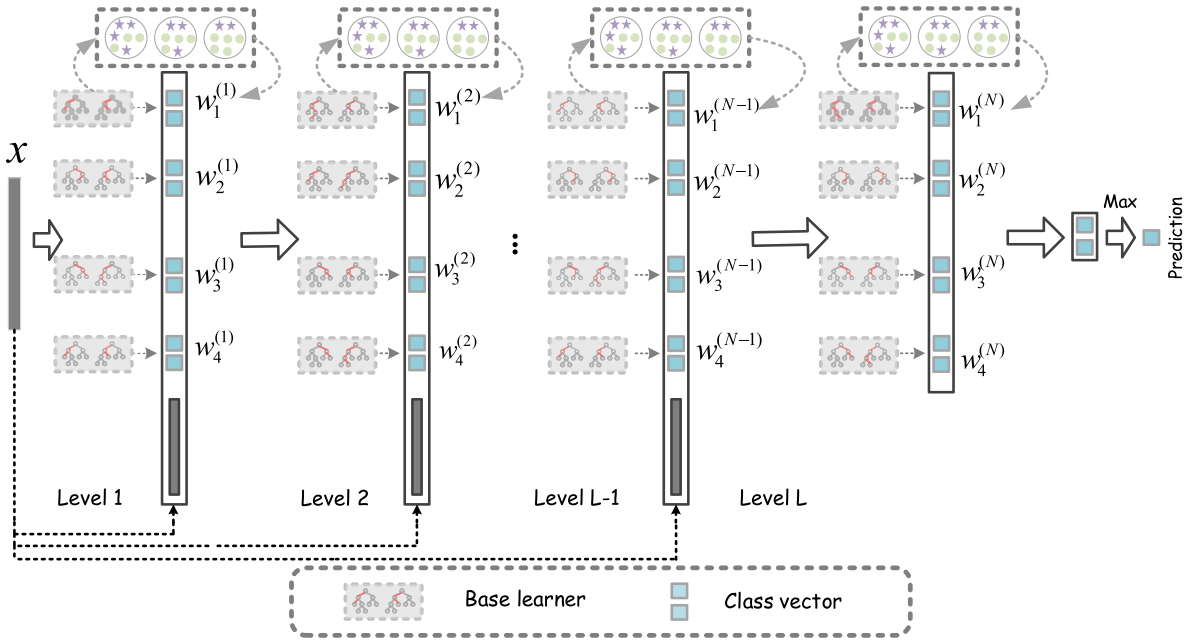


Fig. 1. Structure of Heter-DF.

ensemble methods. Bagging-based ensemble algorithms and stacking ensemble methods reduce the PD prediction error by optimizing the predictive variance, while boosting ensemble approaches iteratively add base learners to reduce the predictive bias. Based on that, in the context of PD modeling, we establish a heterogeneous deep forest cascade model that regards the bagging-based ensemble method, RF, and the boosting ensemble approaches XGBoost (Bequé & Lessmann, 2017) and lightweight gradient boosting machine (LightGBM) (Ke et al., 2017) as base learners to implement F.

2.2. Heterogeneous deep forest

Fig. 1 illustrates the structure of Heter-DF. To answer the third question described in Section 1, as shown in Fig. 1, Heter-DF is designed as a hierarchical cascade architecture that allows us to process information layer-by-layer. Each layer of Heter-DF is composed of multiple tree-based base learners. RF, which is ensembled based on a bagging strategy, is employed as the base learner for cascading in the original concept of deep forests (Zhou & Feng, 2017). To address the first issue described in Section 1, borrowed from the framework of deep forests, each layer of Heter-DF is integrated by multiple strong ensembled base learners. Therefore, Heter-DF can be regarded as an "ensemble of the ensemble" algorithm. Diversity is vital to ensemble learning. Considering both the robustness of base learners and the interpretability of Heter-DF, as shown in Fig. 1, each level of Heter-DF is established by four different tree-based ensembled base learners, where RF, completely RF, XGBoost, and LightGBM are employed to enhance the diversity of Heter-DF. Moreover, to emphasize important features, a boosting strategy is introduced.

Specifically, as shown in Fig. 1, given a training set S, we start training Heter-DF from the first level, and the input of the next level is the concatenation of the output of the previous level and original credit features. Suppose that Heter-DF has L levels, each level is constructed by four base learners, including three heterogeneous tree ensemble algorithms: RF, XGBoost, and LightGBM. An important conception behind Heter-DF is that each DT can generate a class distribution for an individual input credit instance $\mathbf{x}_i$. The class distribution estimates the probabilities that indicate the possibilities that a DT predicted the credit sample into a good applicant and a bad applicant. In the DT, the probabilities are represented by calculating the percentage of examples of different classes on leaf nodes. Therefore, the probability vector generated by each base learner in Heter-DF is the combined result of multiple DTs. RF and completely RF are a group of tree-based approaches that ensemble DTs in a parallel fashion. Their class vectors are generated from averaging the scores of all DTs (see Fig. 2). In comparison, XGBoost and LightGBM are approaches that optimize the tree-based ensemble structure in a boosting way, and their final class vector for a credit sample is the addition of the leaf node scores of all the DTs. Appendix discusses the heterogeneity of RF, XGBoost, and LightGBM in terms of integration methods, optimization patterns, and their internal DT growth strategies (see Fig. 3 and Fig. 4).

According to Zhou and Feng (2017), the class distribution forms a class vector, which is considered as the transformed or augmented features, and the concatenation with the original vector for the cascading of the next level of Heter-DF. Assuming that the original vector is $\mathbf{x}_i$, $p_{t,i,c}$ represents the probability of the ith sample generated by the tth DT in the RF on the cth class. In the case of the RF as the base learner, the cth element of the

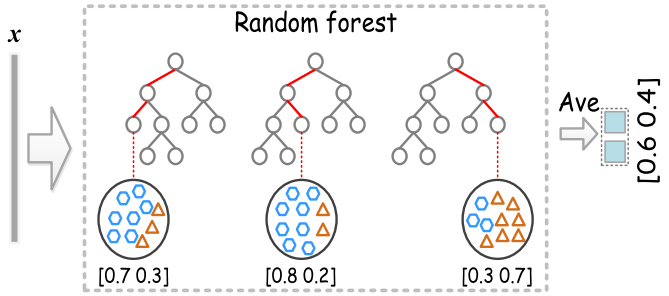


Fig. 2. A simplified random forest.

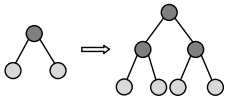


Fig. 3. Level-wise growth of a DT.

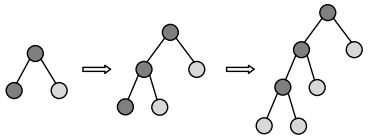


Fig. 4. Leaf-wise growth of a DT.

class vector **V** generated by the RF can be expressed as:

$$V_{i,c} = \frac{1}{T}\sum_{t=1}^{T} p_{t,i,c}, \tag{3}$$

where T is the number of DTs in the tree-based ensemble model. Accordingly, the cth element of the class vector **V** generated by an additive ensemble model such as XGBoost and LightGBM can be calculated as follows (see Appendix):

$$V_{i,c} = \sum_{t=1}^{T} p_{t,i,c}. \tag{4}$$

In the credit scoring task based on Heter-DF, the prediction of good and bad applicants can be modeled as a classification process. Therefore, the class vector of the ith sample obtained from the $(l-1)$-th layer of Heter-DF is expressed as $\mathbf{V}_i^{(l-1)} = [V_{i,0}^{(l-1)}, V_{i,1}^{(l-1)}]$. Then the input vector of the lth cascade level of Heter-DF can be expressed as:

$$\mathbf{x}_i^{(l)} = [\mathbf{x}_i^{(l-1)}, \mathbf{V}_{i,1}^{(l-1)}, \ldots, \mathbf{V}_{i,4}^{(l-1)}]. \tag{5}$$

The input feature of the lth layer is the result of the fusion of the transformed features of the $(l-1)$-th layer of Heter-DF and the original feature **x**. $\mathbf{V}_{i,n_b}^{(l-1)}$ represents the class probability vector for the ith credit sample generated by the n_b-th base learner in the $(l-1)$-th layer.

In Eq. (5), the class vectors produced by multiple base learners in the $(l-1)$-th level of Heter-DF are assigned with equal weights $w_{n_b} = 1$. However, the paralleled base learners are complementary to each other on performance, and the class vector generated by the weak base learner may harm the overall performance of the lth level of Heter-DF. A good solution to address this issue is to assign small weights to weak learners and increase the weight of strong learners, driving the base learners in the lth level of Heter-DF to focus on learning important augmented features and suppressing irrelevant augmented features. The essence of the weighting operation on base learners is to further minimize the error between the predicted result and the real label of each level of Heter-DF. Therefore, in each level of Heter-DF, we denote the augmented probabilistic features in the form of a weighted summation:

$$V_{n_b,i,c} = p_{n_b,i,c} w_{n_b}, \tag{6}$$

where w_{n_b} is the weight assigned to the augmented probabilistic feature produced by the n_b-th base learner. It should be noted that different weights are assigned to the class vectors predicted by different base learners. Therefore, the weights in this study can be described as:

$$\mathbf{w} = [w_1, w_2, w_3, w_4]^T, \tag{7}$$

which follows the condition

$$\sum_{n_b=1}^{4} w_{n_b} = 1, w_{n_b} > 0. \tag{8}$$

By weighting the predictive information of the base learners in each level of Heter-DF, the importance of strong learners in the parallel training of the same cascade level is highlighted. Therefore, calculating the weights for the base learners is a central issue for Heter-DF. Referring to the operation of Pang, Ting, Zhao, and Zhou (2018), Zhou and Feng (2017), to avoid overfitting, K-fold cross-validation (Jiang & Wang, 2017; Moreno-Torres, Sáez, & Herrera, 2012) is performed in each level of Heter-DF to generate a class probability vector. That is, each instance is used for $K-1$ times training to generate $K-1$ class vectors, and the final class vector generated by the n_b-th base learner is the average result of the $K-1$ cross-validated probabilistic predictions. After extending to a new level, the performance of the current level is estimated by calculating the average validation AUC scores (see Section 4) on $K-1$ validation sets (Xia, Tian, Zhang, Xu, & Zhang, 2020), since the AUC is a preferred metric in the credit scoring domain. If there is no significant performance gain, the training process terminated; otherwise, cascading continues. Based on the above analysis, the number of levels of Heter-DF is self-determined. In contrast to

Algorithm 1 Training pseudo-code of Heter-DF.

Input: Training set S, the parameter of K for cross-validation, layer index $l = 1$, and AUC score $\zeta^{(0)} = 0$

Output: Heterogeneous cascade forest model $F(x)$

if layer $l = 1$ **then**
  $S_1^{(l)}, S_2^{(l)}, ..., S_{k-1}^{(l)}, S_k^{(l)} \leftarrow split(S^{(l)})$ $\{S^{(1)} = S\}$
else
  $S^{(l)} = concat(\mathbf{V}^{(l-1)}, S^{(l-1)})$
  $S_1^{(l)}, S_2^{(l)}, ..., S_{k-1}^{(l)}, S_k^{(l)} = split(S^{(l)})$
end if
for $n_b \leftarrow 1$ to 4 **do**
  for $j \leftarrow 1$ to K **do**
    Fit base classifier $f_{n_b}^{(l)}$ for the l-th layer of Heter-DF with in-samples set $S^{(l)} \setminus S_j^{(l)}$
    $\mathbf{V}_{n_b,j}^{(l)} \leftarrow f_{n_b}^{(l)}(S \setminus S_j)$
    $\mathbf{V}_{n_b,j}^{(l,val)} \leftarrow f_{n_b}^{(l)}(S_j)$
  end for
  $A_{n_b}^{(l)} \leftarrow Accuracy(\mathbf{V}_{n_b}^{(l,val)}, y)$ $\{\mathbf{V}_{n_b}^{(l,val)} \leftarrow \mathbf{V}_{n_b,1}^{(l,val)} \cup \mathbf{V}_{n_b,2}^{(l,val)} \cup ... \cup \mathbf{V}_{n_b,K-1}^{(l,val)} \cup \mathbf{V}_{n_b,K}^{(l,val)}\}$
  $\zeta_{n_b}^{(l)} = AUC(\mathbf{V}_{n_b}^{(l,val)}, y)$
end for
Compute weight $w_{n_b}^{(l)} \leftarrow \frac{A_{n_b}^{(l)}}{\sum_{j=1}^{4} A_j^{(l)}}$ for the n_b-th base learner, $n_b = \{1, 2, 3, 4\}$
$\zeta^{(l)} \leftarrow \frac{1}{4}\sum_{i=1}^{4} \zeta_i^{(l)}$
Update $\mathbf{V}_{n_b}^{(l,val)} \leftarrow w_{n_b}^{(l)} \mathbf{V}_{n_b}^{(l,val)}$
$\mathbf{V}^{(l)} = concat(\mathbf{V}_1^{(l,val)}, ..., \mathbf{V}_4^{(l,val)})$
if $\zeta^{(l)} > \zeta^{(l-1)}$ and early_stopping is False **then**
  $l = l + 1$
  Repeat Step 1 to Step 19
else
  Return $V_C^{(L)} = \sum_{n_b=1}^{4} x_{n_b,C}^{(l)} w^{(l)}{}_{n_b} / [\mathbf{V}^{(l)}\mathbf{w}]$
end if

most ensemble models and deep NNs (Xia et al., 2020) with pre-determined model complexity, Heter-DF is an adaptive model which determines its complexity by terminating training. The self-adapted structure of Heter-DF makes it a good choice for modeling datasets of different scales. In the $(l-1)$-th level of Heter-DF, we assign a larger weight to the important augmented feature produced by the $(l-1)$-th base learners to encourage the fitting of the lth level of Heter-DF. Different base learners at the same level may provide different probabilistic predictions. To enhance the effective prediction and suppress the irrelevant fitting, the introduction of weight operators for base learners should be based on considerations regarding the computational burden. Naturally, the K-fold cross-validation process not only provides us with a comparison standard for building a new layer of Heter-DF but also offers a solution for calculating the weights for base learners. Suppose that the K-fold cross-validation of the n_b-th base learner in layer l is expressed as $A_{n_b}^{(l)}$. Then the weight of the n_b-th base learner in layer l can be further denoted as $w_{n_b}^{(l)} \leftarrow \frac{A_{n_b}^{(l)}}{\sum_{j=1}^{4} A_j^{(l)}}$, where $A_{n_b}^{(l)}$ is the K-fold cross-validation AUC of the n_b-th base learner of the lth level of Heter-DF. Algorithm 1 presents the training pseudo-code of Heter-DF, which includes the weighting process for each base learner of Heter-DF.

Tree-based ensemble base learners such as RF are strong classifiers based on a bagging algorithm, which enriches the diversity of each DT by resampling the training set, making RF an ensemble that achieves low-variance predictions. From the perspective of generalization error optimization, gradient boosting decision trees (GBDTs) are another group of ensemble approaches that are distinguished from RF in each iteration. GBDTs keep the same training features while iteratively and step-wisely reducing the prediction bias through multiple weak DTs. The optimization mechanism makes GBDT an ensemble solution that has a low-bias predictive result. In this study, considering the advantages of bagging-type ensemble methods that effectively control variance and boosting-type ensemble algorithms that minimize bias, heterogeneous tree-based ensembled base learners are employed to jointly optimize the overall generalization error.

2.3. Interpretability of Heter-DF

In Heter-DF, four tree-based ensemble approaches – RF, completely RF, XGBoost, and LightGBM – are considered as the base learners. The importance of tree-based ensemble features is calculated as follows:

$$I_i^{(Gini)} = \sum_{k=1}^{K} \sum_{\phi=1}^{\Phi} \Delta_{k,i}^{(Gini)} 1_{k,i}(v_\phi = i), \tag{9}$$

In the realization of random forests, the Gini impurity is considered a feature selection criterion for node splitting. In Eq. (9), K is the number of DTs in an RF; $\Delta_{k,i}^{(Gini)}$ measures the Gini impurity change after the split of feature i in the kth DT of RF; $1_{k,i}(v_\phi = i)$ is a binary function that judges whether the ith feature participates in the growth of the kth DT; and Φ is a set that includes all features that participate in the growth of a DT. Furthermore, we scale feature importance into [0, 1] by the following equation:

$$I_i = \frac{I_i^{(Gini)}}{\sum_{m=1}^{M} I_m^{(Gini)}}, \tag{10}$$

where M represents the number of input features of RF. To encourage the diversity of Heter-DF, two GBDTs are further employed as the base learners of each level of Heter-DF. Unlike bagging-based ensembles, GBDT is an additive ensemble method. The feature importance scores of GBDT are obtained by averaging the feature importance calculated by all DTs, which can be expressed as:

$$I_i^2(F) = \frac{1}{T} \sum_{t=1}^{T} I_i^2(h_t), \tag{11}$$

where $I_i^2(F)$ is the function for calculating the importance score for the ith feature, and T is the number of DTs in

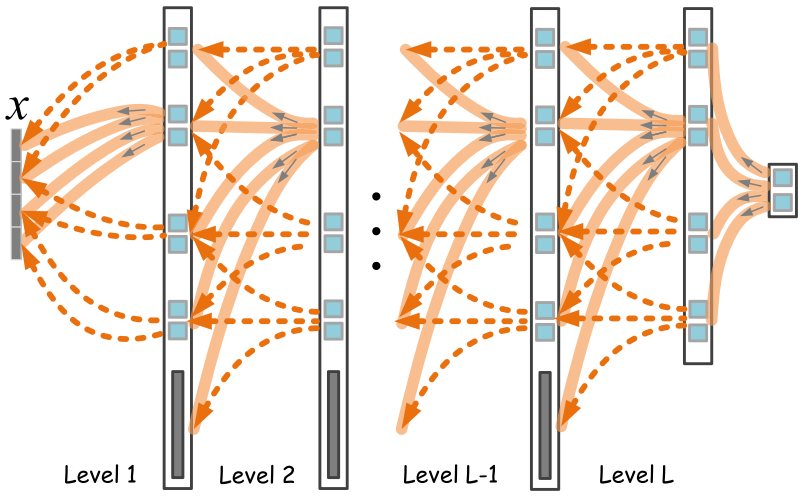


Fig. 5. Importance scores back-propagation mechanism of Heter-DF.

GBDT. The contribution $I_i^2(h_t)$ of the tth DT h_t of the ith input feature is computed by:

$$I_i^2(h) = \sum_{\phi=1}^{\Phi} \Delta_i^2 1_i(v_\phi = i), \tag{12}$$

where Δ_i^2 is the performance improvement of feature i in the splitting of node j, which specifically denotes the change of the Friedman mean square error (MSE) in GBDT. As noted above regarding the description in RF, $1_i(v_\phi = i)$ is a binary function that indicates whether the feature i participates in node splitting during the growth of the DT.

In this study, we utilize RFs and GBDTs as the base learners for higher integration. Based on the advantage that the RF and GBDT algorithms are tree-based ensemble models that are self-interpretable, we further explore the global interpretable mechanism of the Heter-DF. As shown in Fig. 1, from the perspective of feature embedding, the ith level of Heter-DF completes the transformation of old features produced by the $(l-1)$-th cascade level of Heter-DF. The concatenation of original features with the transformed features not only strengthens the learning process but also converts the process of feature transformation into a process of feature enhancement. Based on the structure of the cascading, the final prediction result is the weighted average results of the final level of Heter-DF, which is composed of four tree-based ensemble base learners. Suppose the number of layers of Heter-DF is L, according to the interpretable mechanism of different ensembles in the Lth layer, we can get feature importance scores of the Lth level of Heter-DF, which can be expressed as:

$$I_i^{(L)} = \frac{\sum_{n_b=1}^{4} I_{n_b,i}^{(L)} w_{n_b}}{\sum_{i=1}^{M}\sum_{n_b=1}^{4} I_{n_b,i}^{(L)} w_{n_b}}. \tag{13}$$

Cascading is an approach that involves sequential ensembling, similar to the learning mechanism of NNs, where information is processed layer-by-layer and the output of the previous layer is taken as the input of the current layer. The contribution of the input of the Lth layer to the output of Heter-DF is determined by calculating the feature importance scores. Note that the probabilistic prediction of the lth level is generated based on the original features and augmented features of the $l-1$-th level of Heter-DF. According to the sequential ensemble mechanism, the $(l-1)$-th layer indirectly participates in the final prediction of Heter-DF. Therefore, all the base learners before the $(L-1)$-th level of Heter-DF can be regarded as the indirect contributors to the prediction of Heter-DF. Therefore, we can get the importance score of any layer of Heter-DF based on a back-calculation mechanism, which is illustrated in Fig. 5:

$$I_i^{(l-1)} = I_i^{*(l-1)} \times \sum_{j=M+1}^{M+8} I_j^{(l)}, i \in [1, M], l \in [2, L], \tag{14}$$

where $I_i^{*(l-1)}$ represents the contribution of the i-th feature of the $(l-1)$-th level to the generation of the augmented feature of the lth layer. This is the same as the calculation of the feature importance scores on the Lth level of Heter-DF, which can be expressed as $I_i^{(l-1)} = \frac{\sum_{n_b=1}^{4} I_{n_b,i}^{(l-1)} w_{n_b}}{\sum_{i=1}^{M}\sum_{n_b=1}^{4} I_{n_b,i}^{(l-1)} w_{n_b}}$. $I_i^{(l-1)}$ indicates the indirect contribution of the ith original feature of the $(l-1)$-th layer to the final prediction. $\sum_{j=M+1}^{M+8} I_j^{(l)}$ calculates the sum of the indirect contributions of the augmented features of the lth level to the final prediction result. Combining Fig. 5 and Eq. (14), it can be seen that the indirect contribution of the $(l-1)$-th level is determined based on the augmented feature importance scores of the lth level. Therefore, we can get the indirect contributions of the original features on the different levels of Heter-DF through a back-calculation mechanism. Based on the above analysis, the final feature contribution is the sum of the indirect feature contributions calculated by all the levels previous to the Lth level and the direct contributions of the original features calculated by the Lth level of Heter-DF:

$$I_i = \sum_{l=1}^{L} I_{l,i}. \tag{15}$$

3. Experimental settings

3.1. Credit scoring datasets

The Shandong public money dataset was published by a competition to encourage enterprises, social organizations, and individuals to actively explore the innovative application of big data, accelerate the development and utilization of public data resources, form a normalized docking mechanism between data suppliers and data demanders, and promote the development of the digital economy and information technology. The goal of this project is to establish an accurate risk control model based on real scenes and practical applications, using basic personal identity information, personal housing provident fund deposits, and loan data information, to predict whether the user will be overdue repayment. The Shandong dataset is accessible at http://data.sd.gov.cn/cmpt/cmptDetail.html?id=26.

The Fannie dataset is a mortgage dataset. The Federal National Mortgage Association, normally recognized as Fannie Mae, is a United States government-sponsored publicly traded company. The aim of Fannie Mae is to expand the secondary mortgage market by securitizing mortgages as mortgage-backed securities, thereby enabling borrowers to recycle their investments in further loans and reducing reliance on local savings to increase the number of borrowers in the mortgage market. Fannie Mae released a quarterly single-family fixed-rate mortgage dataset on its website. The data are divided into acquisition data and performance data, grouped by quarter from 2000 to 2020. The acquisition data contain static mortgage information regarding when the mortgage was issued and delivered to Fannie Mae. The performance file contains monthly performance data for each loan, from acquisition to the current status in the previous quarter. The Fannie Mae dataset can be accessed at https://capitalmarkets.fanniemae.com/credit-risk-transfer/single-family-credit-risk-transfer/

The Give dataset, normally known as the Give Me Some Credit dataset, is a credit dataset to evaluate whether a consumer has good credit. Banks play an important role in the market economy. They decide who can get financing and what conditions support or undermine investment decisions. For the market and society to function, individuals and enterprises need to gain credit. Credit scoring models, which estimate the PD, are a group of algorithms used by banks to decide whether they should issue loans. The Give dataset is issued on the Kaggle platform https://www.kaggle.com/c/GiveMeSomeCredit. The Kaggle competition encourages participants to improve their credit scores by predicting the likelihood that someone will experience financial distress in the next two years.

The BankFear dataset is released by Indessa Bank. Indessa Bank has underperformed in the last three quarters. Their non-performing assets have reached a peak, and investors have lost confidence. As a result, its share price fell by 20% in the last quarter. After careful analysis, it was found that most of the non-performing assets were contributed by defaulting borrowers. Based on data collected over the years, the bank decided to design an ML-based approach to find these defaulters and develop plans to reduce them. Indessa Bank utilized a large amount of information from investors to approve its loans. For example, if any customer applies for a $20,000 loan, the investor will perform due diligence on the requested loan application. In this challenge, we aim to develop an effective algorithm to help Indessa Bank predict the PD of a potential borrower. The BankFear dataset was collected from https://www.kaggle.com/codenamekash/bank-fears-loanliness. Detailed information regarding the four credit scoring datasets is presented in Table 1.

Table 1
Details of the four credit datasets.

Dataset	#Samples	#Features	Good/Bad
Shandong	40,000	19	37,243/2757
Give	150,000	10	139,974/10,026
Fannie	369,335	18	340,458/28,877
BankFear	532,428	32	406,601/125,827

3.2. Credit scoring models

In this study, a large number of credit scoring models are introduced for comparison: classical statistical-based credit scoring methods, LDA and LR; simple ML-based credit scoring models, SVM, KNN, NN, and DT; and tree-based ensemble credit scoring models, RF, GBDT, XGBoost, and LightGBM. Moreover, some highly advanced ensemble credit scoring models are designed for further comparison.

To finetune the hyperparameters of Heter-DF, we split each credit scoring dataset into a training set, a validation set, and a test set, with a ratio of 80%:10%:10%. For a comprehensive comparison, since Heter-DF is a heterogeneous model that is ensembled by four types of tree-based ensemble frameworks, we first finetuned the hyperparameters of base learners for homogeneous cascade models. The optimal hyperparameter setting for base learners in Heter-DF is inherited from the hyperparameters of the base learners in homogeneous cascade models. Since Heter-DF is a tree-based highly ensembled framework, the tree-based structure is robust and does not rely on many parameters. Thus, we only finetuned the vital hyperparameters for homogeneous cascade models and Heter-DF, as presented in Table 2.

4. Experimental results

4.1. Heter-DF vs. classical ML-based methods

In this study, graphical analysis is first provided for preliminary comparison. The receiver operating characteristic (ROC) curve (Mandrekar, 2010), whose x-axis represents the false positive rate (FPR) and whose y coordinate denotes the true positive rate (TPR), is a popular graphical metric that has been widely applied for assessing the performance of credit scoring. The FPR is the ratio of samples incorrectly classified as bad applicants where the real labels are good applicants: $FPR = \frac{FP}{FP + TN}$ (see

Table 2
Hyperparameter settings for cascade models.

Algorithm	Parameter	Range	Stride
Cascade RF	Number of base learners for RF	[500,2000]	100
	Maximum depth of each DT	{5,10,15,30,None}	
	Minimal number of samples to split at each splitting node	[500,1000]	100
	Minimal number of samples at each leaf node	[30,100]	10
Cascade XGBoost	Number of iterations	[500,1000]	25
	Learning rate which controls the optimization step	[1e−3,5e−1]	
	Maximum depth of each DT in XGBoost	[5,15]	1
Cascade LightGBM	Number of iterations	[500,1000]	25
	Learning rate	[1e−3,5e−1]	
	Maximum depth of each DT	[5,15]	1
	Number of leaves of each DT	[20,2^max_depth]	5

Table 3
Confusion matrix of credit scoring results: TP is the number of bad applicants that are accurately predicted; FN represents the number of bad applicants that are incorrectly predicted; FP denotes the number of good applicants that are wrongly classified; TN counts the number of accurately predicted good applicants.

Confusion matrix		Prediction	
		Bad	Good
Label	Bad	TP	FN
	Good	FP	TN

Table 3). The TPR is the ratio of samples correctly classified as bad applicants where the labels are bad applicants as well: $TPR = \frac{TP}{TP + FN}$.

Fig. 6 shows the ROC curves of various credit scoring models on the four credit scoring datasets: Fig. 6(a) shows the ROC curves on the Shandong dataset; Fig. 6(b) shows an ROC curve comparison of the various credit scoring models on the Give dataset; Fig. 6(c) shows ROC curves of the credit scoring models on the Fannie dataset; and Fig. 6(d) compares the ROCs of multiple credit scoring models on the BankFear dataset. As can be seen from Fig. 6, Heter-DF has the largest area under the ROC curve (AUC) on the Shandong dataset. The AUC of boosting-type ensemble models is significantly larger than that of other single ML-based credit scoring ROCs and bagging-type ensemble models such as RF. Furthermore, as can be seen from Fig. 6(a), the AUC of the ensemble algorithms is significantly larger than that of a single ML-based credit scoring model, which further verifies the effectiveness of the ensemble strategy. LR and LDA, the classical statistical models for early-stage credit scoring, have low modeling AUCs on the Shandong dataset, which indicates the poor robustness of these two algorithms on the Shandong dataset. The largest AUC of Heter-DF in Fig. 6(a) demonstrates that Heter-DF is the best choice for the credit scoring for the Shandong dataset compared with other mainstream credit scoring models. As can be seen from Fig. 6(b), Heter-DF, which has been further integrated based on the ensemble model, improves the comprehensive performance of credit scoring on the Give dataset. The AUC of Heter-DF is the largest among ML-based credit scoring models and statistical methods. Similarly, compared to individual ML-based credit scoring classifiers, ensemble credit scoring approaches significantly improve the performance. As can be seen from Fig. 6(c), LightGBM and XGBoost, which have AUC values close to those for Heter-DF, outperform other ensemble approaches such as RF, AdaBoost, and GBDT on the Fannie dataset, indicating that boosting-type ensemble approaches are good solutions for accurate credit scoring. From the ROC curves of various credit scoring algorithms on the BankFear dataset shown in Fig. 6(d), it can be seen that Heter-DF as well as XGBoost, LightGBM, and GBDT are relatively close, and the AUCs of these ensemble algorithms are larger than the AUC of other credit scoring models. From the fine-grained comparison, it can be seen that Heter-DF is superior to LightGBM and XGBoost, which further improves the comprehensive performance of the advanced ensemble algorithms on the BankFear dataset. Further, the AUCs of NN, RF, and AdaBoost are relatively close, and KNN has the worst performance.

It also can be seen from Fig. 6 that the AUCs of all the credit scoring methods on the Shandong dataset and the BankFear dataset are larger than those of the other two datasets, Fannie and Give. This result is determined by the complexity of credit scoring datasets. In addition, from the comparison of Fig. 6(b), Fig. 6(c), and Fig. 6(d), it can be seen that the performance of the KNN on the Give, Fannie, and BankFear datasets is relatively poor. This is because of the imbalanced problem of credit datasets. KNN is an algorithm that predicts good and bad applicants by searching for the k-nearest neighbors, while the samples labeled as good applicants are the dominant class in the imbalanced credit datasets.

To quantitatively evaluate the results of credit scoring, several credit scoring evaluation metrics were selected for further comparison: the accuracy, AUC, precision score, recall score, F1 score, and Brier loss. Accuracy measures the predictive ability of the credit scoring model to discriminate good from bad applicants:

$$acc = \frac{TP + TN}{TP + FN + FP + TN}. \tag{16}$$

AUC assesses how much better the prediction made by the model is than a random guess.

The Type-I error indicates the proportion of samples whose real state is "good" that are incorrectly predicted:

$$e - I = \frac{FP}{FP + TN}. \tag{17}$$

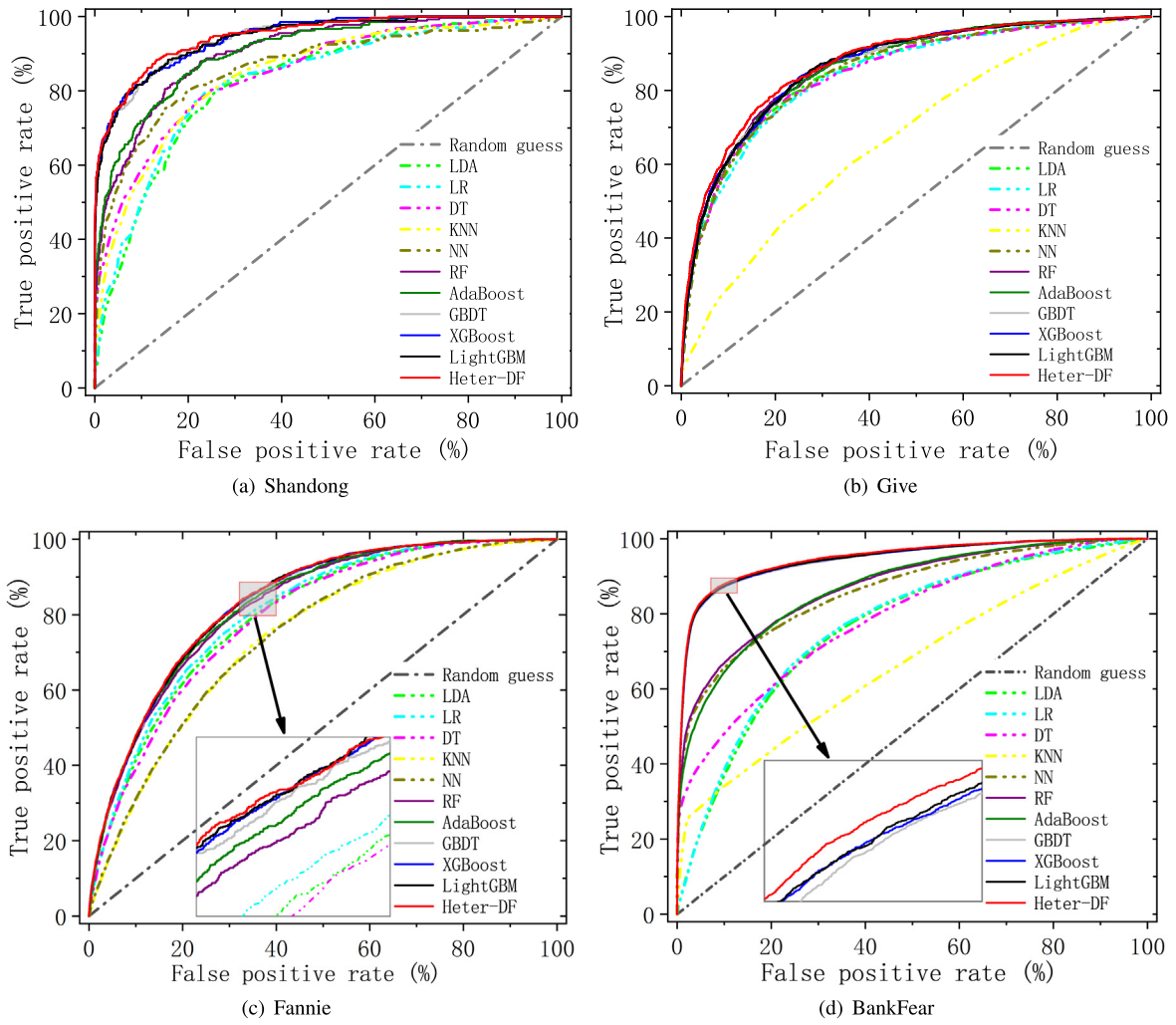


Fig. 6. ROC curve comparison of various credit scoring models on four credit scoring datasets.

The Type-II error measures the misclassification ratio of bad applicants:

$$e - II = \frac{FN}{TP + FN}. \tag{18}$$

The Brier loss score describes the average probabilistic prediction error:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2. \tag{19}$$

The Hmeasure is an imbalance measure that takes into account the different costs between the misclassification of good applicants and bad applicants. It was first proposed by Hand (2009). The Hmeasure estimates the cost weight function for credit datasets with different class distributions through the beta distribution. The cost weight function can be expressed as:

$$u(\upsilon) \triangleq f(\upsilon; \alpha, \beta) = \frac{\upsilon^{\alpha-1}(1-\upsilon)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}du}, \tag{20}$$

where υ represents the cost ratio, and α, β is the parameter of the beta distribution, which obeys $\alpha > 1$, $\beta > 1$. From Eq. (20), we assign a unified cost weight distribution to all credit scoring models, and the calculation of the misclassification cost does not rely on the classifier but rather on the credit scoring dataset. Given the cost weight distribution function, we can calculate the HMeasure value from the perspective of loss:

$$\begin{aligned} HM &= 1 - \frac{L_{\alpha,\beta}}{L_{\max}} \\ &= 1 - \frac{\int Q(T(\upsilon); b, \upsilon)u_{\alpha,\beta}(\upsilon)d\upsilon}{\pi_0\int_0^{\pi_1}\upsilon u_{\alpha,\beta}(\upsilon)d\upsilon + \pi_1\int_{\pi_1}^1(1-\upsilon)u_{\alpha,\beta}(\upsilon)d\upsilon}, \end{aligned} \tag{21}$$

where $Q(T(\upsilon); b, \upsilon)$ is the total loss of the cost pair (b, υ), π_0 and π_1 are the prior possibilities of class 0 and class 1, respectively, $L_{\alpha,\beta}$ denotes the general losses on the weight

Table 4
Comparison of the performance of credit scoring models on the Shandong dataset.

Algorithm	Accuracy	AUC	Type-I error	Type-II error	BS	Hmeasure
LR	.93575	.83025	.00348	.91386	.05314	.36367
LDA	.933	.82830	.00830	.88764	.05527	.34424
DT	.947	.85095	**0**	.79401	.04553	.40474
KNN	.93825	.84830	.00027	.91760	.05497	.38717
NN	.94275	.86556	.01741	.61423	.04690	.47193
AdaBoost	.9525	.90971	.00348	.66292	.24516	.53798
RF	.94675	.90677	.00107	.78277	.04251	.52476
GBDT	.9675	.94514	.00482	.41948	.02813	.65172
XGBoost	.9675	.94594	.00375	**.43446**	.02813	.65868
LightGBM	.96425	.94424	.00536	.46068	.02899	.66093
Heter-DF	**.96825**	**.94864**	.00214	.44570	**.02807**	**.66884**

distribution $u_{\alpha,\beta}$, and $L_{\max}$ represents the maximum loss for the balanced class distribution.

Based on the consideration that each layer of the Heter-DF is a structure of parallel implementation, decision-making between the base learners of each cascade layer is independent. In the design of Heter-DF, we did not finetune all the hyperparameters of each cascading layer, because RF, XGBoost, and LightGBM are essentially robust ensemble algorithms. Instead, in the cascading stage, we borrowed some parameter settings from the finetuned cascade RF, cascade XGBoost, and cascade LightGBM. For example, only the parameters of min_samples_split and min_samples_leaf, the number of DTs in the RF, and the depth of each DT were optimized for cascade RF; in the cascading layer which consists of XGBoost, we learned from the setting of parameter values such as the learning_rate, min_samples_split, and min_samples_leaf in the parameter finetuning stage, and we only finetuned the number of iterations and the maximum depth of each DTs for each XGBoost-based base learners. In the optimization of cascade LightGBM, we similarly fixed some parameters of LightGBM-based base learners such as the subsample on the training set, colsample_bytree, etc., and only some vital parameters, such as the number of DTs and the number of leaves, were finetuned. The hyperparameter settings for the cascade RF, cascade XGBoost, and cascade LightGBM are shown in Table 2. The optimal base learners were further selected for the heterogeneous ensemble of Heter-DF.

Table 4 shows the performance of various credit scoring models on the Shandong dataset. It can be seen in Table 4 that Heter-DF achieves the best results in terms of the accuracy, AUC, BS, and Hmeasure, showing its potential for modeling credit scoring. In addition, as can be seen from Table 4, compared to classic statistical-based credit scoring models, ML-based models show great promise for credit scoring. From the performance of DT on Type-II error, it can be seen that a single DT algorithm is not good at identifying "bad" applicants. Such defects can be solved by an ensemble strategy. Moreover, compared to individual DT, the performance of the tree-based ensemble model on Type-I error slightly improved. Fundamentally, however, solving the problem of the high error in the identification of "bad" applicants should take into account the different costs of misclassifying "bad" applicants and applicants with good credit scores. The same issue also appeared in KNN-based credit scoring. Compared with bagging-based ensemble approaches, boosting-type ensemble methods such as GBDT, XGBoost, and LightGBM show more robust credit scoring results, and their AUCs significantly improved. With cascading, the performance of Heter-DF on the Shandong dataset was further improved.

Table 5 compares the performance of various credit scoring models on the Give dataset. As can be seen from Table 5, LR has the optimal Type-II error, but its Type-I error is the largest among the credit scoring algorithms, making its AUC and other metrics on the Give dataset unsatisfactory. Compared with LR, another classic statistical-based model LDA shows the opposite prediction result. Its Type-I error is lower than that of LR, but its Type-II error is much smaller than that of LR. That is to say, if we pay more attention to the profit of credit scoring, LR can be regarded as a better choice than LDA. If our goal is to minimize the cost of misrecognizing a "bad" applicant, LDA is a better choice than LR. Similar to the performance on the Shandong dataset, DT has the lowest Type-I error, but it has the highest Type-II error. Thus, DT more accurately predicts "bad" applicants than "good" applicants. The bias prediction results make DT has a low Hmeasure score. Further, AdaBoost and GBDT have a better AUC than RF, which indicates that boosting-ensemble approaches are better candidates for credit scoring than bagging-type ensemble algorithms. Moreover, as can be seen from the comparison on the AUC and BS values of AdaBoost, although the AUC of AdaBoost is better than that of RF, its value in BS is much larger. BS measures the average prediction error of AdaBoost on the Give dataset. The larger BS value of AdaBoost implies a larger average error between the predicted probability and the true state. Since the calculation of AUC is related to the ordering of "bad" samples, we conclude that the contradiction of AdaBoost's high AUC and high BS is that the probabilistic predictions of AdaBoost on most "bad" samples tend to have a probability of 0.5.

Table 6 compares the performance of the models on the larger-scale credit scoring dataset BankFear. Similar to the results on the Shandong and Give datasets, LDA and LR (which are based on the assumption that credit datasets are linear) performed poorly on the large-scale credit dataset. In addition, on the relatively complex BankFear dataset, due to the low complexity of a single DT, the AUC of DT is the lowest. Compared with other individual ML-based credit scoring classifiers for the BankFear dataset, NN gets the best AUC due to its ability to learn the nonlinear relationship between variables. Moreover, as can be

Table 5
Performance comparison of credit scoring models on the Give dataset.

Algorithm	Accuracy	AUC	Type-I error	Type-II error	BS	Hmeasure
LDA	.93533	.84455	.00914	.84453	.05209	.38729
LR	.9356	.85071	.01785	**.71815**	.05626	.39984
DT	.9338	.67329	**0**	.99599	.05999	.10127
KNN	.93747	.84774	.00978	.80341	.05021	.40199
NN	.934	.85075	.01300	.81043	.05140	.39727
AdaBoost	.93607	.86387	.01221	.79037	.24587	.42090
RF	.93733	.86583	.00821	.82748	.0491	.42801
GBDT	.93713	.86581	.00971	.80441	.04916	.42696
XGBoost	**.94088**	.86640	.00971	.80943	.04917	.42874
LightGBM	.93687	.86649	.01035	.80441	.04926	.42647
Heter-DF	.9408	**.87404**	.00813	.78659	**.04712**	**.45055**

Table 6
Performance comparison of credit scoring models on the BankFear dataset.

Algorithm	Accuracy	AUC	Type-I error	Type-II error	BS	Hmeasure
LDA	.77778	.77015	.05620	.75754	.15081	.24348
LR	.77584	.76439	.01050	.80542	.15294	.42585
DT	.80417	.66441	.02896	.73461	.15556	.16523
KNN	.82590	.78777	.02707	.64829	.13155	.30124
NN	.86320	.86218	.03733	.45755	.10611	.45858
AdaBoost	.83848	.87040	.00881	.65392	.24804	.45362
RF	.83756	.87341	**.00618**	.66630	.11457	.47617
GBDT	.91946	.94357	.02724	.25238	.06437	.70394
XGBoost	.91879	.94464	.02544	.26103	.06466	.70788
LightGBM	.92003	.94542	.02532	.25619	.06418	.70937
Heter-DF	**.92234**	**.94818**	.02570	**.24481**	**.06249**	**.71614**

Table 7
Comparison of the performance of credit scoring models on the Fannie dataset.

Algorithm	Accuracy	AUC	Type-I error	Type-II error	BS	Hmeasure
LDA	.92162	.80615	.00238	.97303	.06468	.27233
LR	.92135	.79862	.00264	.97337	.06513	.25647
DT	.92170	.74122	**0**	1	.06816	.16336
KNN	.92151	.79196	.00176	.98167	.06539	.24444
NN	.88517	.74332	.05552	**.81293**	.09056	.16738
AdaBoost	.92156	.82720	.00426	.95159	.24614	.31241
RF	.92262	.82303	.00091	.97752	.06284	.30340
GBDT	.92470	.83198	.00375	.94212	.06100	.32033
XGBoost	**.92481**	.83218	.00349	.94389	.06093	.32078
LightGBM	.92441	.83240	.00425	.93100	.06095	.32050
Heter-DF	.92441	**.83367**	.00387	.94460	**.06080**	**.32342**

seen from Table 6, NN well balances the Type-I error and Type-II error compared with other individual ML-based credit scoring models. Compared with RF, XGBoost, and LightGBM, Heter-DF, which is an ensemble of the above three tree-based ensemble models, significantly improved in terms of the AUC. In this study, Heter-DF reached the best performance, except in terms of the Type-I error metric. This outstanding advantage shows the potential of Heter-DF for credit scoring. Furthermore, the performance improvement of the ensemble models on various metrics further verifies the effectiveness of the ensemble mechanism.

Table 7 shows the performance of various credit scoring models on the Fannie dataset. As shown in Table 7, similar to the performance on the previous datasets, a single DT has biased predictions in the identification of good and bad applicants, leading to its Type-I error being 0 and its Type-II error being 1. Compared with the performance on other datasets, the comprehensive performance of NN on the Fannie dataset is poor. The main reason for this is that the Fannie dataset is highly imbalanced, and the cross-entropy loss function-based NN cannot effectively process the imbalanced credit scoring dataset. Compared with the single classifiers, the AUCs of the ensemble classifiers on the Fannie dataset improved. Similar to the previous conclusion, boosting-type ensemble models can be regarded as more effective credit scoring models compared with bagging-based ensemble models. Compared with existing credit scoring models, Heter-DF scored highest in terms of the AUC, BS, and Hmeasure. This indicates that Heter-DF is a good choice for reducing the prediction error and improving the performance on an imbalanced credit scoring dataset.

4.2. Heter-DF vs. deep cascade ensemble models

To further verify the effectiveness of the heterogeneous ensemble framework, we compared the performance of Heter-DF to some deep cascade ensemble models, including homogeneous deep cascade ensemble models and heterogeneous deep ensemble approaches. In the

Table 8
Performance comparison of deep cascade ensemble models.

Base learner		Homogeneous deep cascade ensemble						Heterogeneous deep cascade ensemble			
		DT	LDA	LR	RF	XGB	LGB	Heter-M1	Heter-M2	Heter-M3	Heter-DF
Shandong	Accuracy	.95775	.95175	.95925	.95675	.966	.9655	.9585	.95775	.9645	**.96825**
	AUC	.91364	.88010	.90037	.9363	.94671	.94786	.91678	.93725	.94302	**.94864**
	e-I	.00428	**.00003**	.00054	.00107	.00509	.00375	.00080	.00107	.00402	.00214
	e-II	.57678	.72285	.60300	.63296	.4382	**.04644**	.61049	.61798	.47565	.44569
	BS	.03603	.07697	.07198	.03454	.02866	.02844	.03603	.03393	.02927	**.02807**
	HM	.57845	.49907	.55701	.63584	.65865	.66789	.58464	.64061	.65019	**.66884**
Give	Accuracy	.93693	.93867	.93867	.9384	.9368	.9377	.93947	.93987	.9392	**.9408**
	AUC	.85409	.85392	.85185	.86699	.86638	.86697	.86829	.87204	.87280	**.87404**
	e-I	.00878	.00685	.00685	**.00621**	.01057	.00928	.00713	.00721	.00942	.00813
	e-II	.83638	.83740	.83740	.83852	.80241	.80642	.82114	.81402	.79268	**.78659**
	BS	.04933	.05031	.05031	.04888	.04927	.04919	.04772	.04750	.04764	**.04712**
	HM	.41122	.41804	.41857	.42646	.42861	.42854	.43859	.44694	.44574	**.45055**
BankFear	Accuracy	.92413	.92376	.92378	.83921	.92202	.92208	**.92419**	.84344	.92223	.92234
	AUC	.79116	.81136	.80864	.8855	.9474	.947	.81737	.88810	.94746	**.9481**
	e-I	.00108	.00003	**.000029**	.0063	.02565	.02580	.00032	.00647	.01970	.0257
	e-II	.98189	.99964	.99929	.65892	.24632	.24560	.99041	.63939	.25618	**.24481**
	BS	.06402	.06292	.06350	.11154	.06303	.0626	**.06226**	.10986	.06275	.06249
	HM	.24228	.27818	.27337	.49704	.71342	.71411	.28973	.51587	.71368	**.71614**
Fannie	Accuracy	.92405	.92424	.92359	.92318	.92392	.92454	.92419	.92462	**.92500**	.92441
	AUC	.79602	.81332	.80162	.81727	.83276	.83317	.81764	.81908	.83232	**.83367**
	e-I	.00114	.00023	.00191	**.00006**	.00332	.00399	.00384	.00053	.00311	.00387
	e-II	.98260	.99077	.97905	.996	.94797	**.94141**	.99290	.98224	.94602	.9446
	BS	.06348	.06243	.06384	.06291	.06094	.0609	.06218	.06187	.06089	**.06080**
	HM	.25645	.28137	.25953	.29096	**.3240**	.32235	.29129	.29450	.32049	.32342

heterogeneous deep cascade ensemble set, since large-scale credit scoring is involved in this study, we employed efficient ML-based individual classifiers (DT, LDA, and LR), as well as advanced ensemble approaches (RF, XGBoost, and LightGBM), as base learners for the homogeneous ensemble. Correspondingly, the heterogeneous deep cascade ensemble library covered deep frameworks that were ensembled by different weak base learners DT-LR-LDA-EDT (EDT refers to extremely randomized trees; we abbreviate this model as Heter-M1), hybrid heterogeneous deep framework that were level-wisely established based on DT-LR-RF-ERF (see Appendix A.1; we abbreviate this model as Heter-M2) and DT-RF-ERF-XGB (Heter-M3), and the robust deep tree-based heterogeneous ensemble model Heter-DF.

Table 8 compares the performance of homogeneous cascade ensemble models and Heter-DF. From the comparison of the AUC in Table 8, it can be seen that – compared with homogeneous cascade ensemble models such as deep cascade random forests (CD-RF), deep cascade XGBoost (DC-XGBoost), and deep cascade LightGBM (DC-LightGBM) – Heter-DF improves the AUC score on the four credit datasets. Combining Tables 4–8, it can be seen from comparing classical credit scoring approaches and their homogeneous ensemble modifications that the adaptive deep structure for ML-based classifiers is beneficial for their performance improvement on credit scoring task, further demonstrating the effectiveness of the cascading ensemble framework. Moreover, it can also be seen from the intra-group comparisons of homogeneous deep cascade ensemble models and heterogeneous deep cascade ensemble models that embedding a strong classifier, such as RF, XGBoost, or LightGBM, into the cascade framework shows a more significant performance improvement. Among the comparisons between heterogeneous deep cascade ensemble models, Heter-DF, which is an ensemble of four advanced tree-based heterogeneous ensemble models and adaptively grows its structure in a weighted manner, achieves optimal AUC scores and superior BS on the four datasets. These results demonstrate that Heter-DF can be a good choice for the probabilistic prediction of good and bad applicants.

The deep cascade ensemble framework, which adaptively increases its depth according to the complexity of credit datasets, may result in varied structures with different base learner ensembles. Therefore, to analyze the marginal effect of Heter-DF, we report the level-wise training cost and prediction cost of deep cascade ensemble models for credit scoring, which is shown in Table 9. All the results are averaged from 30 runs. In Table 9, to compare the computational costs of DC-RF, DC-XGBoost, DC-LightGBM, and heterogeneous deep cascade ensembles, we set the number of DTs in RF, XGBoost, and LightGBM to 500 to establish the tree-based deep cascade ensemble models. The training and prediction costs of the deep cascade ensemble models that the advanced tree-based ensemble base learners participated in are further counted and averaged from 30 runs. As can be seen from Table 9, the computation cost of the deep cascade ensemble models is highly related to the base learner selection. Simple base learners such as DT, LDA, and LR for the deep cascade ensemble models have advantages in training and prediction on credit scoring datasets, yet the performance improvement is not as significant as for the deep cascade ensemble models that are ensembled by tree-based advanced ensemble approaches. On the other hand, deep cascade ensemble models that are established based on advanced tree-based ensemble algorithms have better predictive performance at credit scoring tasks, while their computational cost is not as competitive as deep cascade ensemble models that are ensembled by

Table 9
Computational cost comparison of deep cascade ensemble models for credit scoring (unit: s).

Ensemble model		Homogeneous deep cascade ensemble						Heterogeneous deep cascade ensemble			
Base learner		DT	LDA	LR	RF	XGB	LGB	Heter-M1	Heter-M2	Heter-M3	Heter-DF
Training	Shandong	1.497	1.998	2.523	42.914	23.896	20.894	1.804	26.912	24.050	29.029
	Give	3.230	3.247	15.466	179.269	149.453	120.674	6.852	90.190	123.128	159.782
	BankFear	11.337	18.500	58.318	1130.445	807.880	600.895	24.598	832.382	773.038	860.299
	Fannie	5.215	7.760	25.089	495.369	300.789	250.586	10.820	229.282	300.460	380.237
Prediction	Shandong	0.024	0.069	0.096	1.256	0.165	0.156	0.047	0.524	0.659	0.683
	Give	0.068	0.141	0.122	0.902	0.137	0.110	0.0910	0.527	0.398	0.420
	BankFear	0.243	0.285	0.303	2.683	0.705	0.589	0.243	1.203	1.427	1.502
	Fannie	0.159	0.223	0.229	1.806	0.464	0.387	0.169	1.293	1.002	1.239

simple ML-based predictors. This motivates us to prioritize the performance-complexity dilemma of Heter-DF in our future work (see Section 5).

4.3. Statistical comparison between Heter-DF and other credit scoring models

To detect whether Heter-DF statistically outperforms existing credit scoring models, we first perform a Friedman test (Abellán & Castellano, 2017; Feng et al., 2018). The Friedman test is a non-parametric significance test that computes its statistics based on the ranks of classifiers:

$$\chi_F^2 = \frac{12D}{N_c(N_c+1)}\Big[\sum_{n_c=1}^{N_c} AvR_{n_c}^2 - \frac{N_c(N_c+1)^2}{4}\Big], \tag{22}$$

where N_c is the number of classifiers; D denotes the number of credit datasets; AvR_{n_c} calculates the average rank of the n_c-th classifier over D datasets, which can be computed by $AvR_{n_c} = \frac{1}{D}\sum_{d=1}^{D} r_{d,n_c}$; and r_{d,n_c} is the average rank of the jth classifier on the dth credit dataset cross the four comprehensive credit scoring metrics. Specifically, r_{d,n_c} can be defined as $r_{d,n_c} = \frac{1}{4}(r_{d,n_c,\text{Acc}} + r_{d,n_c,\text{AUC}} + r_{d,n_c,\text{BS}} + r_{d,n_c,\text{HM}})$, where $r_{d,n_c,\text{Acc}}$, $r_{d,n_c,\text{AUC}}$, $r_{d,n_c,\text{BS}}$, and $r_{d,n_c,\text{HM}}$ represent the accuracy ranking, AUC ranking, BS ranking, and HM ranking, respectively, of the jth classifier on the dth credit dataset. If the Friedman statistic is smaller than a critical value, we accept the null hypothesis that there is no significant difference among credit scoring approaches. Otherwise, we reject the null hypothesis and perform a post hoc procedure, namely, the Nemenyi test (Junior, Nardini, Renso, Trani, & Macedo, 2020; Xiao, Wang, Chen, Xie, & Huang, 2021; Zhang, Yang, & Zhang, 2021), to investigate the significant differences among the existing benchmarks and Heter-DF. The conclusion that there is a significant difference between credit scoring baselines and Heter-DF holds when the average rank differs by at least a critical difference (CD) level. The critical difference is computed by:

$$CD_\alpha = q_\alpha\sqrt{\frac{N_c(N_c+1)}{12D}} \tag{23}$$

To investigate at what confidence level the proposed Heter-DF outperforms other imbalanced credit scoring algorithms, as well as the advanced balanced ensemble techniques LightGBM and XGBoost, we perform a Friedman test to determine whether there is a statistically

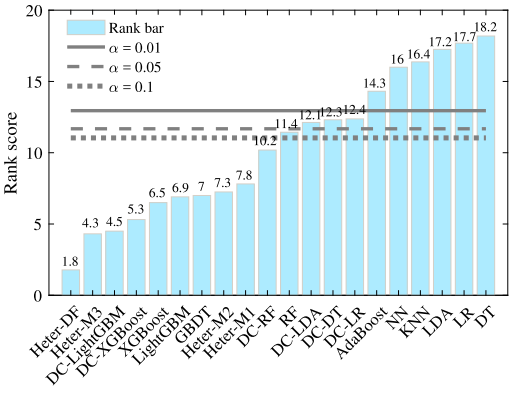
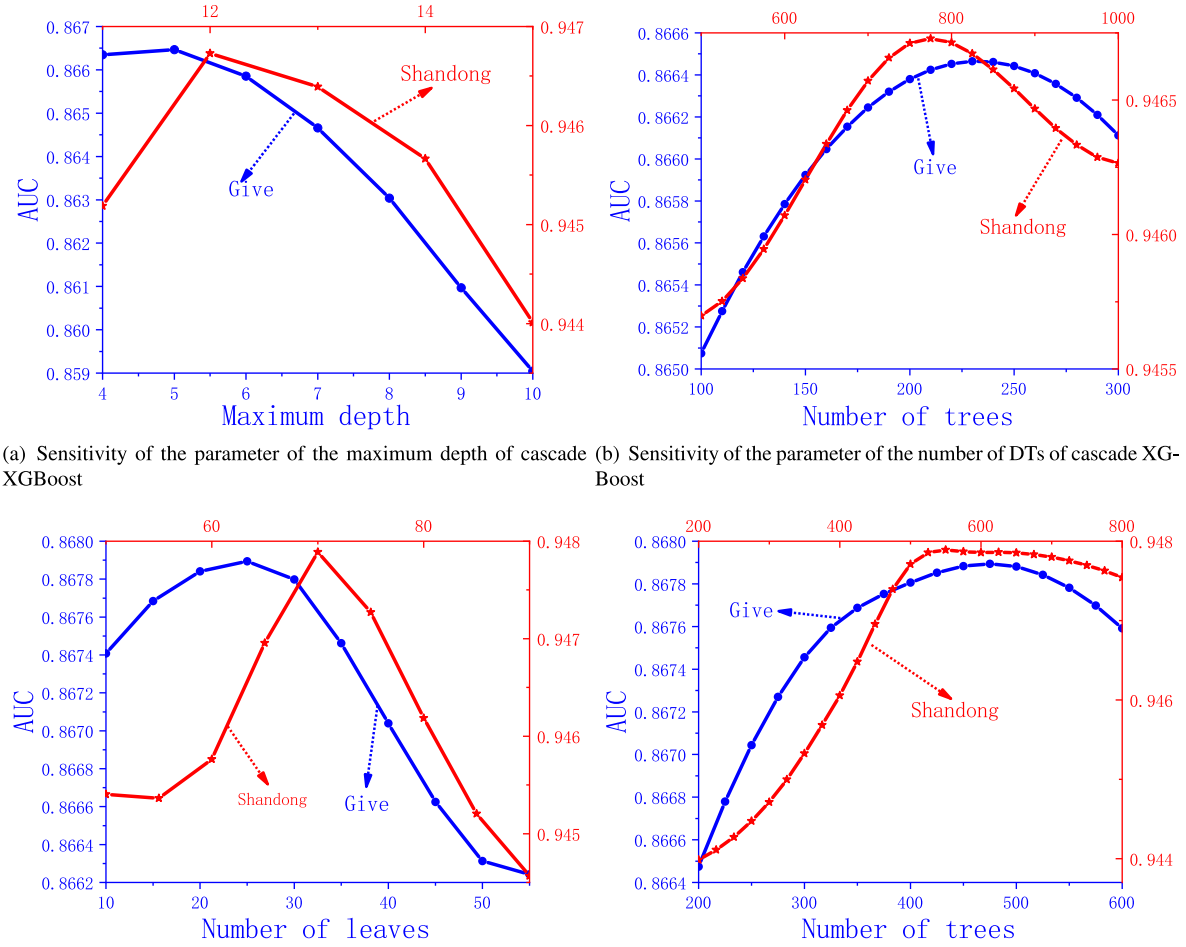


Fig. 7. Average rankings of credit scoring algorithms and CDs for the Nemenyi test: the horizontal lines represent the addition values of difference CDs and the lowest ranking of the credit scoring models.

significant difference between these credit scoring algorithms. To see the statistically significant ranking of various credit scoring algorithms, we first follow (Xia et al., 2018)'s work and calculate the Friedman statistics based on the average ranking of the accuracy score, AUC, BS, and HM, which are $\chi_{\text{ACC}} = 9.27$, $\chi_{\text{AUC}} = 9.16$, $\chi_{e2} = 9.32$, and $\chi_{\text{Gmean}} = 8.61$. All reject the null hypothesis with 99% confidence (i.e. there is no statistically significant difference among credit scoring models). Next, a post hoc procedure, the Nemenyi test, is conducted for a pairwise comparison. According to Eq. (23), the critical values $q_{0.01} = 3.97$, $q_{0.05} = 3.52$, and $q_{0.1} = 3.29$ at significance levels of $\alpha = 0.01$, $\alpha = 0.05$, and $\alpha = 0.1$, respectively, are first computed from a studentized range distribution. The critical differences at significance levels $\alpha = 0.01$, $\alpha = 0.05$, and $\alpha = 0.1$ are further calculated as $CD_{0.01} = 11.17$, $CD_{0.05} = 9.90$, and $CD_{0.1} = 9.26$, respectively, according to Eq. (23).

Fig. 7 shows the average ranks of credit scoring approaches and CDs computed under different significance levels for the Nemenyi test. As seen in Fig. 7, the dominant average ranking of LightGBM and XGBoost demonstrates the superiority of the tree-based heterogeneous cascade ensemble framework for credit scoring, which ranks first with an average ranking score of 1.8. If we regard Heter-DF as a baseline, Heter-DF outperforms AdaBoost, NN, KNN, LDA, LR, and DT at a significance level of $\alpha = 0.01$.



(a) Sensitivity of the parameter of the maximum depth of cascade XGBoost

(b) Sensitivity of the parameter of the number of DTs of cascade XGBoost

(c) Sensitivity of the parameter of the number of leaves of cascade LightGBM

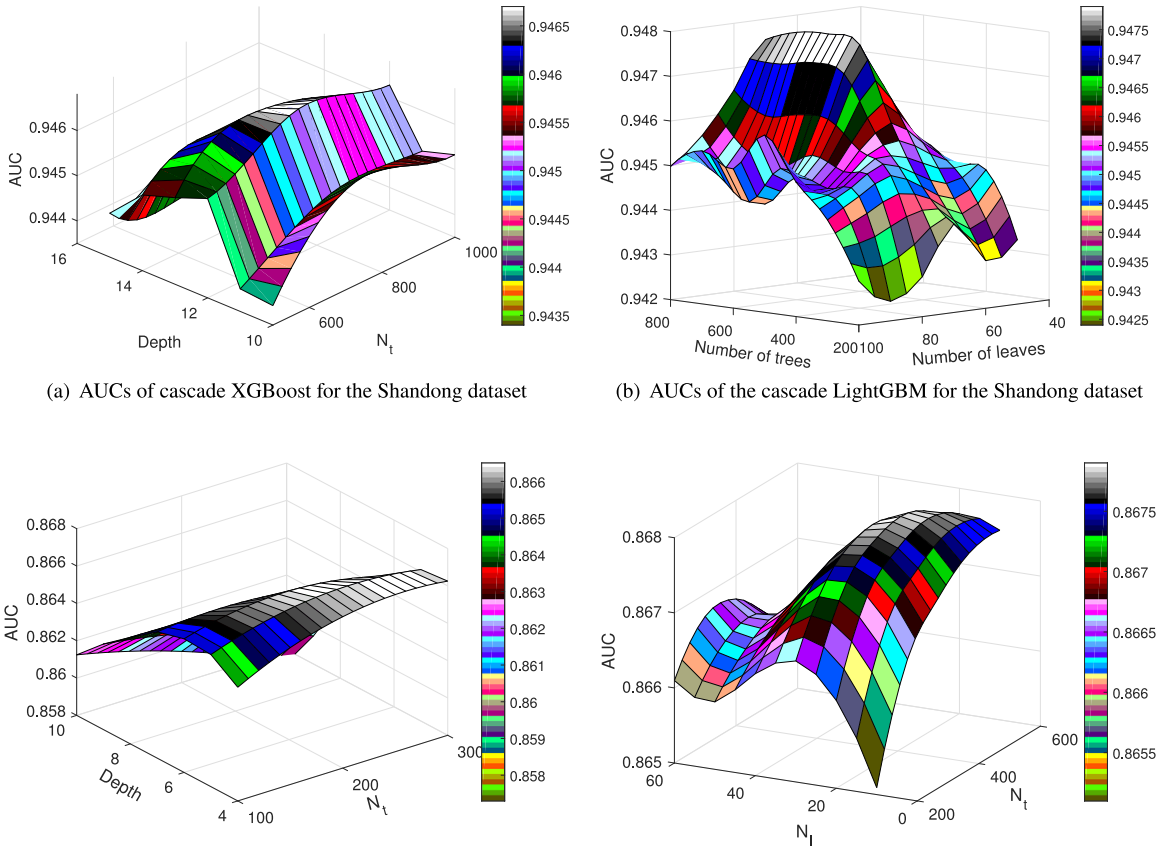
(d) Sensitivity of the parameter of the number of DTs of cascade LightGBM

Fig. 8. Sensitivity map of vital parameters of cascade XGBoost and LightGBM on the Give and Shandong datasets.

It can also be seen from the significance level $\alpha = 0.05$ that Heter-DF is superior to deep cascade ensemble models that are homogeneously integrated by weak learners, including DC-LDA, DC-DT, and DC-LR. Moreover, we conclude that Heter-DF can be a better solution for precise credit scoring compared with RF with a confidence of 90%. The statistically higher ranking of Heter-DF and Heter-M3 compared with DC-LightGBM and DC-XGBoost indicates that the heterogeneous ensemble is a robust option compared with a homogeneous ensemble. The leading ranking of both homogeneous cascade integration and heterogeneous cascade ensemble compared to individual ML-based classifiers further validates the advantages of the cascade structure.

4.4. Sensitivity analysis of Heter-DF

The good performance of Heter-DF benefits not only from the ability of adaptive cascading but also from the heterogeneous structure and boosting strategy implemented by K-fold cross-validation for each base learner. As shown in Algorithm 1, K-fold cross-validation is further performed to highlight the important augmented features for the construction of the cascade layer. This is because each base learner is essentially a tree-based ensemble algorithm that is robust and has fewer hyperparameters to be finetuned. As explained above, in this study, the parameters of the number of DTs in each base learner of the cascade ensemble models and the parameter of the maximum depth of RF and XGBoost, as well as the parameter of the number of leaves in LightGBM, are vitally finetuned. Based on the leading advantages of cascade XGBoost and cascade LightGBM compared to cascade forests, in Fig. 8, we focus on analyzing cascade XGBoost and cascade LightGBM on the Give dataset and Shandong dataset. In cascade XGBoost, we borrowed some parameter settings from the finetuned XGBoost and finetuned the parameter of the number of DTs and the maximum depth of each DT in each XGBoost. Similarly, in the optimization of cascade LightGBM, we only optimize the number of DTs and the number of leaves in each tree of LightGBM and fix other parameter settings such that they are the same as the finetuned LightGBM.



(a) AUCs of cascade XGBoost for the Shandong dataset

(b) AUCs of the cascade LightGBM for the Shandong dataset

(c) AUCs of cascade XGBoost for the Give dataset

(d) AUCs of cascade LightGBM for the Give dataset

Fig. 9. AUC 3D plots for cascade ensemble models for the Shandong and Give datasets.

It can be seen from Fig. 8 that the optimal maximum depth of cascade XGBoost on the Give dataset is 5, and the optimal value on the Shandong dataset is 12. Below the optimal value, cascade XGBoost exhibits under-fitting. Excessively increasing the maximum depth of each tree in cascade XGBoost will easily lead to the risk of overfitting. The optimal value for the parameter of the number of leaves of the cascade LightGBM on the Give dataset is 25, and the optimal value on the Shandong dataset is 70. Similarly, because LightGBM is a leaf-wise boosted DT, an excessive or insufficient number of leaf nodes with cascade LightGBM will lead to large deviations from the optimal AUC value. Finally, from the parameter sensitivity map of the number of trees in cascade XGBoost and cascade LightGBM shown in Fig. 8(b) and Fig. 8(d), it can be seen that the parameter of the number of DTs is a dominant parameter for the performance optimization of cascade XGBoost and cascade LightGBM.

The optimal parameter combination of cascade XGBoost and cascade LightGBM is grid-searched from the value intervals in Table 2, and the searched AUCs are shown in Fig. 9. Fig. 9(a) show an AUC 3D plot of cascade XGBoost for the Shandong dataset; Fig. 9(b) shows an AUC map of cascade LightGBM for the Shandong dataset; Fig. 9(c) presents the finetuned AUC scores of cascade XGBoost on the Give dataset; and Fig. 9(d) depicts the AUCs of cascade LightGBM for the Give dataset.

Figs. 8 and 9 show that the optimal base learner in cascade XGBoost is jointly determined by the number of DTs and the parameter of maximal depth that controls the complexity of each base learner in cascade XGBoost. As can be seen from Figs. 8 and 9, excessively raising the number of DTs for XGBoost and the complexity of each DT will result in overfitting; too few DTs for XGBoost and shallower DTs will lead to the poor fitting of the cascade ensemble models. The optimal parameters of the base learners in Heter-DF inherit the settings of the homogeneous cascade ensembled models. Therefore, to determine the optimal parameter settings for base learners in Heter-DF, the parameters of the number of DTs in each base learner and the parameters controlling the complexities of the DTs should be carefully finetuned.

4.5. Performance of different ratios of training sets

Benefiting from the cascade structure and K-fold cross-validation, Heter-DF is designed as an adaptive cascading framework whose expressive power and complexity are driven by the scale of the data. Such characteristics imply that Heter-DF is a potential candidate with a competitive advantage in small-scale credit dataset modeling. To explore the scalability of the Heter-DF on small-scale credit datasets, we further compared the performance of Heter-DF under different ratios of training sets. Fig. 10 shows the performance of Heter-DF under different ratios of

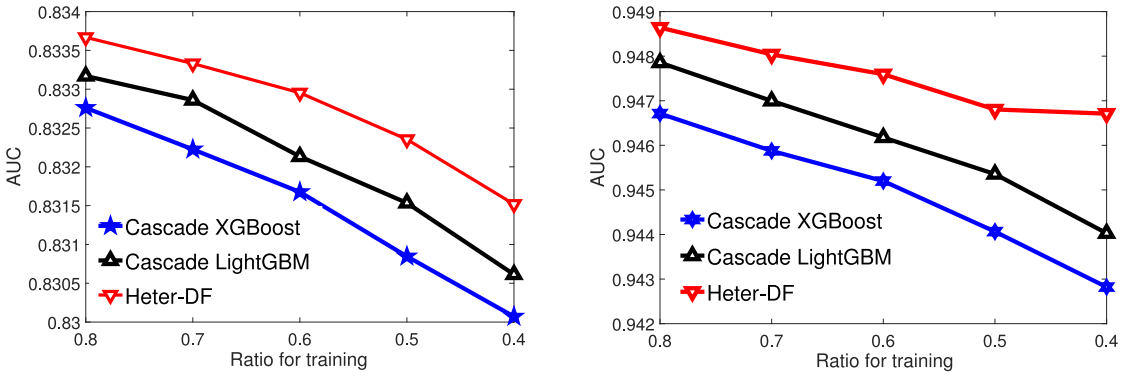


Fig. 10. AUC 3D plot for cascade ensemble models for the Shandong and Give datasets.

training sets on the Shandong dataset and Fannie dataset. Fig. 10(a) shows the test AUCs of cascade forests, cascade XGBoost, and cascade LightGBM under different training set ratios on the Fannie dataset. Fig. 10(b) shows the AUCs of cascade forests, cascade XGBoost, and cascade LightGBM under different training set ratios on the Shandong dataset.

It can be seen from Fig. 10 that even in the participation of only 40% of the training data, Heter-DF, cascade XGBoost, and cascade LightGBM can reach an AUC score of 0.83 on the Fannie dataset. Compared with the AUC under 80% training data, the AUC under 40% training data only decays by 0.024%. This advantage is more prominent on the Shandong dataset. Heter-DF borrows the parameter settings of the base learners from homogeneous cascade ensemble models, and each base learner is boosted to minimize the prediction error, which makes the heterogeneous framework more robust and scalable than homogeneous ensembles. As such, Heter-DF always gets a better AUC score than homogeneous ensembles such as cascade XGBoost and cascade LightGBM (Crone & Finlay, 2012) under different ratios of training data. In addition, it can be seen from the AUC decay of the cascade ensemble framework under different training set ratios that more data would bring better credit scores, and the reduction of a larger ratio of training data implies a faster AUC decays. In conclusion, the cascading framework ultimately expands its application where credit scoring training data are insufficient.

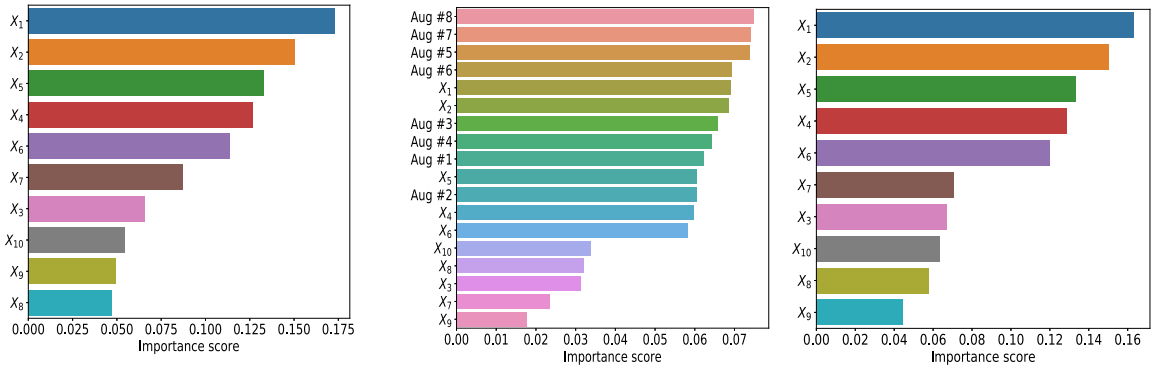
4.6. Interpretation of Heter-DF

Credit scoring models are a group of decision-making algorithms that assist lending companies or financial institutions in deciding whether to grant loans to loan applicants based on the prediction of the credit scoring model regarding whether the applicant is capable of repaying financial debts. An intelligent credit scoring model not only effectively improves the efficiency of loan approvals but also allows loan analysts to focus on a certain proportion of applications. This saves costs and reduces human subjectivity when assessing the risk of default. A large number of ML-based credit methods and AI-based approaches have shown great progress with credit scoring, but most ML-based credit scoring models are "black boxes" since they do not provide any explanation behind the decision. As a result, any financial expert is unlikely to be willing to trust the prediction of the model (Lasek & Gagolewski, 2020). The interpretability of the model has recently regained attention in emerging fields.

Based on the good interpretability of the tree algorithm, we further explore the interpretability of Heter-DF. According to the different ensemble mechanisms, the existing ensemble credit scoring models can be divided into sequential ensemble methods and parallel ensemble approaches. Heter-DF further integrates tree-based ensembled models in the manner of a cascade. Essentially, cascading can be regarded as a sequential ensemble. According to the structure of sequential prediction, the global interpretability of Heter-DF is explored in this study according to the description in Section 2.3. Fig. 11 shows the importance score ranking of each layer of Heter-DF, which is calculated according to the importance scores back-propagation mechanism discussed in Section 2.3.

Fig. 11(a) shows the feature importance of the first layer of Heter-DF for the Give dataset, and Fig. 11(b) shows the importance score of the second layer of Heter-DF on the Give dataset. Fig. 11(c) shows the contribution of input features for Heter-DF, calculated based on Fig. 11(a) and Fig. 11(b) according to Eqs. (9)–(15). The semantic information of the variables of the Give dataset is shown in Table 10.

It can be seen from Fig. 11(a) that x_1, x_2, and x_3 are the top three factors that drive the prediction of the first cascade layer of Heter-DF, where x_1 represents the ratio of the total amount of loans to the total amount of credit, x_2 is the age of the borrower, and x_5 contains the monthly income information of the borrower. After the feature transformation of the first level of Heter-DF, the second-level Heter-DF makes a prediction based on the output transformed feature vector of the first level of Heter-DF and the original features. Therefore, in the second layer of Heter-DF, 18 feature importance scores can be calculated, as shown in Fig. 11(b). As can be seen from Fig. 11(b), in the importance score ranking of the second layer of Heter-DF, the augmented feature #8, augmented feature #7, augmented feature #5, and augmented feature #6 play



(a) Importance score of 1st cascade layer of Heter-DF (b) Importance score of 2nd cascade layer of Heter-DF (c) Feature contributions of Heter-DF

Fig. 11. Importance score of each cascade layer of Heter-DF for the Give dataset.

Table 10
Semantic information of the features of the Give dataset.

Variable	Name	Description
x_1	RevolvingUtilizationOfUnsecuredLines	Recycling of unsecured loans: except for real estate and car loans, the total amount of credit card and personal credit line
x_2	Age	Age of borrower
x_3	NumberOfTime30–59DaysPastDueNotWorse	Overdue times in 30–59 days
x_4	DebtRatio	Debt ratio
x_5	MonthlyIncome	Monthly income
x_6	NumberOfOpenCreditLinesAndLoans	Loan amount
x_7	NumberOfTimes90DaysLate	90 days overdue: the number of times the borrower is overdue for 90 days or more
x_8	NumberRealEstateLoansOrLines	Number of real estate loans or lines: mortgage and real estate loans, including home equity lines of credit
x_9	NumberOfTime60–89DaysPastDueNotWorse	Number of times past due by 60–89 days but not worse
x_{10}	NumberOfDependents	The number of family members, excluding borrowers themselves

vital roles in the prediction of the second layer of Heter-DF. In other words, the prediction result of the last layer is largely determined by the augmented features produced by the previous layer. This result further demonstrates the effectiveness of the cascading ensemble mechanism. The essence of Heter-DF is a sequential ensemble model. According to the mechanism of the sequential ensemble, the prediction result is generated based on the input features of the last cascade layer, and the input features of the last layer are the concatenation of the raw input features and the output of the previous layer. In the Give dataset, the transformed features are generated based on the original features. Based on the good interpretability of the tree-based ensemble framework, the importance score of the last layer on the Give dataset is the direct contribution to the prediction of Heter-DF, while the importance score calculated in the first layer of Heter-DF can be regarded as the indirect contribution of Heter-DF for the Give dataset. Therefore, according to the importance score back-calculation mechanism described in Section 2.3, the contribution of original features to the prediction results of Heter-DF for the Give dataset can be further computed, as shown in Fig. 11(c). The feature importance ranking in Fig. 11(c) is consistent with the results in Fig. 11(a): x_1, x_2, and x_5 are still the main explanatory variables for Heter-DF's final prediction; the above three variables are important for processing loan applications. Therefore, more attention should be paid by decision-makers to such information when determining whether to issue a loan.

5. Conclusion

Issuing loans to potential borrowers is an important business for banks and lending institutions. Accurately distinguishing good from bad borrowers using historical repaying information and assessing their potential default risk can help banks and financial institutions increase profits and reduce unnecessary losses. Credit scoring is a useful tool for evaluating individual credit risk, and it has become a tool for banks and lending institutions to predict the potential default probability of customers. AI technology has pushed the performance of credit scoring to a higher level compared with statistical-based methods, providing new finance technology for high-performance discrimination between good and bad applicants. Ensemble approaches have been extensively developed by strengthening multiple weak learners into more robust learners. Based on the efficient modeling ability and intuitive interpretable mechanism of the tree algorithm, we proposed a tree-based heterogeneous cascade forest, called Heter-DF, for interpretable credit scoring. Heter-DF is a data-driven algorithm which increases the complexity of the model through a cascading structure to adapt to datasets of various scales. This feature of the proposed algorithm offers it good scalability and makes it suitable

for applications where the training credit data are insufficient. Furthermore, a bagging-type ensemble strategy and a boosting-type ensemble algorithm are used as the base model for further integration and to control the variance-bias tradeoff, thereby reducing the generalization error of credit scoring. A boosting strategy is introduced to minimize the prediction error of each cascade layer of Heter-DF, making the prediction results optimal. Finally, the inherent interpretability of the tree-based algorithm allows us to further explore the model's global interpretation capabilities. The exploration of the interpretability of algorithms is crucial to the task of credit scoring because the ultimate goal of the model's decision-making is to provide a reference for decision-makers. A credit scoring model that is interpretable can provide a direct and accurate decision-making reference for risk managers.

Although Heter-DF improves the performance of credit scoring, some vital issues need to be further addressed:

(1) Although the global self-interpretable mechanism of the algorithm was preliminarily explored in this study, the intrinsic local interpretation mechanism of Heter-DF is still relatively complex. Future work will consider analyzing the interpretation effect of the individual samples of the model from the perspective of a model-agnostic interpretation algorithm, and some efficient interpretation algorithms will be introduced, such as game theory-based explainable theories (Lundberg & Lee, 2017) and LIME (Tulio Ribeiro, Singh, & Guestrin, 2016).
(2) Heter-DF heterogeneously ensembles advanced ensemble methods (RF, ERF, XGBoost, and LightGBM) to boost the performance of credit scoring. The performance-complexity dilemma motivates us to turn future work in the direction of developing an efficient heterogeneous ensemble framework for credit scoring. Therefore, we will consider some feasible solutions that cover pruning technologies, such as (Wu, Liu, Xie, Chow, & Wei, 2021) and Bian and Chen (2021), to accelerate the training and prediction of Heter-DF.
(3) Imbalanced credit scoring is still a challenge. Therefore, some knowledge of imbalanced learning will be introduced into our future work to improve the accuracy of identifying bad applicants, and to assist banks in avoiding unnecessary losses. Some interesting works are in progress, such as the synthetic minority over-sampling (SMOTE) technique (Douzas, Bacao, & Last, 2018) and imbalanced DT (Akash, Kadir, Ali, & Shoyaib, 2019), and we will consider implementing a cost-aware loss function, such as the focal loss (Lin, Goyal, Girshick, He, & Dollár, 2017), into a boosting-type base learner for Heter-DF.

CRediT authorship contribution statement

Wanan Liu: Conceptualization, Methodology, Software, Investigation, Formal analysis, Resources, Writing – original draft.. **Hong Fan:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing. **Meng Xia:** Software, Investigation, Data curation, Visualization.

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.

Appendix. Heterogeneous base learners for Heter-DF

A.1. Random forest

The basic idea of the DT algorithm is to continually divide the subspace in the original feature space and to label each subspace (good/bad applicants). Therefore, given a test sample, each DT will generate a probability distribution based on the proportion of the training sample in the subspace where the sample is located. The probabilistic prediction of RF is then the average proportions of all DTs in the RF. Fig. 2 illustrates a simplified RF. As shown in Fig. 2, each test sample will find a path in each DT to determine which leaf nodes it should fall under. The probabilistic prediction of each DT on the test sample is the statistics on the proportion of samples that belong to a different class determined in the training stage. The probabilistic prediction of RF is in this study a class vector with two-dimensional elements, averaged from the proportions of all DTs. Ordinary RF and completely RF are two base learners employed in Heter-DF. Extremely RF (ERF) randomly selects one feature for node splitting. RF randomly selects $\sqrt{M}$ candidate features for node splitting according to splitting criteria such as the information gain and Gini index, where M is the number of features. In Zhou and Feng (2017), these two types of RFs are utilized as base models to enrich the diversity of the cascading framework.

A.2. GBDT models

GBDT is a boosting-type ensemble algorithm. The key to boosting ensemble approaches is to iteratively add the weak learner $f_i(x)$ (classification and regression DT) to an ensemble model, thereby step-wisely reducing the predictive error $L(F_t(x), y)$. According to the optimization principle of GBDT, $L(F_t(x), y) < L(F_{t-1}(x), y)$, and GBDT is an addictive model that is ensembled by multiple weak DTs:

$$F_T(x) = \sum_{t=1}^{T} \alpha_t f_t(x), \tag{24}$$

where $f_t(x)$ represents the weak learner of the tth iteration, and α_t is the weight coefficient for the tth DT.

Because boosting-type ensemble approaches increase the complexity of the model and thus reduce the predictive error, gradient boosting has become a popular approach in various data mining and machine learning competitions, such as the Kaggle platform. The goal of the tth DT is to further reduce the loss of the $(t-1)$-th ensembled model. Therefore, the learning objective of each DT can be converted into the fitting of the residual,

and the residual is approximated by the first derivative of the loss function, which can be expressed as:

$$g_t = \frac{\partial L(f_{t-1}(x_i), y)}{\partial f_{t-1}(x_i)}. \tag{25}$$

DTs are an efficient and easy-to-understand ML algorithm that can be used for classification and regression tasks. DTs have a tree structure composed of multiple nodes and edges (splitting paths). The process of training a DT is mainly divided into three steps: node splitting, decision tree growth, and pruning. Node splitting involves the process of feature selection. In each node's splitting, the best feature is selected to divide the feature subspace according to feature selection criteria such as the information gain and Gini index. The decision rule on the split path determines which leaf node the sample should fall under. Therefore, given a prediction sample, each DT would find the optimal decision path from top to bottom and make predictions based on the set of if-else selection rules. According to different strategies of node splitting, GBDT can be further extended to more advanced ensemble models such as XGBoost with level-wise growth for each DT and LightGBM with leaf-wise node splitting.

A.3. XGBoost

Fig. 3 shows the level-wise growth strategy of a DT. As shown in Fig. 3, each internal node involves a process of splitting. In the process of node splitting, all the nodes at the same level are participating in the growth of the current level of the DT. Therefore, in the node splitting of a DT in XGBoost, node selection is not involved, and all the internal nodes are split in parallel. The growth of the DT in traditional GBDT is expanded from the top layer to the bottom layer, and the non-leaf nodes of each layer participate in the process of splitting, which implies that each node splitting involves a process of traversing all the training data. Unlike the original GBDT, the advantage of level-wise growth for the DT is that the splitting of nodes at each level can be implemented in parallel, which accelerates the training of XGBoost. Correspondingly, level-wise learning may lead to the redundancy of splitting nodes, resulting in the splitting of a large number of irrelevant nodes, thus causing a certain burden on computing resources and memory devices.

XGBoost is an efficient implementation of GBDT that introduces a regularization function $\Omega(f)$ in the optimization process of the loss function to alleviate the overfitting problem of traditional GBDT. Therefore, based on the GBDT optimization objective, the objective function of XGBoost can be expressed as

$$\begin{aligned} &\min \{L(F_T(x), y) + \Omega(f) + \xi\} \\ &= \min \left\{ \sum_{i=1}^{N} (\hat{y}_i, y_i) + \sum_{t=1}^{T} \Omega(f_t) + \xi \right\}, \end{aligned} \tag{26}$$

where ξ is a constant term. In XGBoost, the complexity of each tree is determined by two parts: the number of leaf nodes, and the regularization term $\|s\|^2$. $\|s\|^2$ denotes that an L_2 regularization is performed on the leaf nodes s, which is equivalent to adding L_2 smoothing to the score of each leaf node to alleviate overfitting. Therefore, $\Omega(f)$ can be re-expressed as:

$$\Omega(f) = \gamma T_{\mathrm{L}} + \frac{1}{2}\lambda \sum_{i=1}^{T_{\mathrm{L}}} s^2, \tag{27}$$

where γ and λ are coefficient terms used to control the complexity of XGBoost, and T_{L} is the number of lead nodes. In the implementation of the original GBDT, the tth DT realizes the fitting of the residual of the $(t-1)$-th ensemble, which is approximated by the first-order negative gradient of the loss function. To approximate the optimization objection, XGBoost effectively balances the accuracy and complexity and transforms the fitting object from a one-order into a second-order Taylor expansion. Suppose the base learner of XGBoost is expressed as:

$$f_t(x) = s_{q(x)}, s \in R^{T_L}, q : R^d \to \{1, 2, \ldots, T_{\mathrm{L}}\}, \tag{28}$$

The value on these T_{L} leaf nodes forms a vector s with T_L dimension, and $q(x)$ is a function that maps the input from the root node to the leaf node. Naturally, $s_{q(x)}$ is the predicted score of the DT for sample x.

Therefore, according to the optimization objective of each DT in XGBoost, we can further calculate the score s of the leaf node by:

$$s_j^* = -\frac{G_j}{H_j + \lambda}, \tag{29}$$

where $G_j = \sum_{i \in \mathscr{I}_j} \frac{\partial L(f(\mathbf{x}_i), y_i)}{\partial f(\mathbf{x}_i)}$ represents the first-order partial derivative of the loss function, $H_j = \sum_{i \in \mathscr{I}_j} \frac{\partial^2 L(f(\mathbf{x}_i), y_i)}{\partial f(\mathbf{x}_i)^2}$ is the second-order partial derivative of the loss function, and $\mathscr{I}_j = \{i | q(\mathbf{x}_i) = j\}$ denotes sample $\mathbf{x}_i$ that falls under leaf node j.

In this way, the value of the loss function can be iteratively optimized through the fitting of DTs. Unlike the bagging-type ensemble mechanism that is illustrated in Fig. 2, we can further get the probabilistic prediction result of the addictive boosting-type ensemble model as the summation of all the predictions of DTs in XGBoost.

A.4. LightGBM

Although traditional boosting algorithms (such as GBDT and XGBoost) have good performance at predicting risky and not-risky loans, traditional boosting cannot meet the higher requirements in terms of efficiency and scalability when faced with high-dimensional and complex credit scoring environments. The main issue that hinders the practical application of the boosting ensemble-based approach is that GDBT involves scanning all the training samples at each node splitting. To improve the efficiency and extend the application to high-dimensional credit scoring, LightGBM introduces a leaf-wise growth strategy for each DT. The leaf-wise growth strategy of the DT is shown in Fig. 4. In the implementation of LightGBM, the following technologies have been further proposed to accelerate the training process. The first is the histogram bucketing technique. In the implementation of the histogram bucketing technique, the feature value is firstly converted into a bin value. That is, a piecewise

function is performed for each feature value, and different feature values are distributed into several different bins. Therefore, the feature value can be converted from a continuous value into some discrete bins. The second strategy is gradient-based one-side sampling (GOSS). Instead of taking all the samples into account for calculating the loss gradient, GOSS calculates the gradient of the loss function based on the samples for acceleration. The third strategy is exclusive feature bundling (EFB), which processes high-dimensional sparse features, thereby further improving the efficiency of the boosting model. Instead of considering all the features to get the best splitting node, EFB bundles some features to reduce the dimension of the feature and the cost of finding the best splitting node. For a mathematical description and understanding of LightGBM, a detailed introduction can be found in the official implementation of LightGBM.

References

Abellán, J., & Castellano, J. G. (2017). A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Systems with Applications*, *73*, 1–10.

Abellán, J., & Mantas, C. J. (2014). Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, *41*(8), 3825–3830.

Akash, P. S., Kadir, M. E., Ali, A. A., & Shoyaib, M. (2019). Inter-node Hellinger distance based decision tree. In *IJCAI* (pp. 1967–1973).

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, *23*(4), 589–609.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, *54*(6), 627–635.

Bahnsen, A. C., Aouada, D., & Ottersten, B. (2014). Example-dependent cost-sensitive logistic regression for credit scoring. In *2014 13th international conference on machine learning and applications* (pp. 263–269). IEEE.

Bequé, A., & Lessmann, S. (2017). Extreme learning machines for credit scoring: An empirical evaluation. *Expert Systems with Applications*, *86*, 42–53.

Bian, Y., & Chen, H. (2021). When does diversity help generalization in classification ensembles? *IEEE Transactions on Cybernetics*.

Crone, S. F., & Finlay, S. (2012). Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting*, *28*(1), 224–238.

De Bock, K. W., Coussement, K., & Lessmann, S. (2020). Cost-sensitive business failure prediction when misclassification costs are uncertain: A heterogeneous ensemble selection approach. *European Journal of Operational Research*, *285*(2), 612–630.

Douzas, G., Bacao, F., & Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*, *465*, 1–20.

Feng, X., Xiao, Z., Zhong, B., Dong, Y., & Qiu, J. (2019). Dynamic weighted ensemble classification for credit scoring using Markov Chain. *Applied Intelligence*, *49*(2), 555–568.

Feng, X., Xiao, Z., Zhong, B., Qiu, J., & Dong, Y. (2018). Dynamic ensemble classification for credit scoring using soft probability. *Applied Soft Computing*, *65*, 139–151.

Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning*, *77*(1), 103–123.

He, H., Zhang, W., & Zhang, S. (2018). A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications*, *98*, 105–117.

Henley, W., et al. (1997). Construction of a k-nearest-neighbour credit-scoring system. *IMA Journal of Management Mathematics*, *8*(4), 305–321.

Hu, Y. C., & Ansell, J. (2007). Measuring retail company performance using credit scoring techniques. *European Journal of Operational Research*, *183*(3), 1595–1606.

Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications*, *33*(4), 847–856.

Huang, B., & Thomas, L. C. (2015). The impact of Basel Accords on the lender's profitability under different pricing decisions. *Journal of the Operational Research Society*, *66*(11), 1826–1839.

Jiang, G., & Wang, W. (2017). Error estimation based on variance analysis of k-fold cross-validation. *Pattern Recognition*, *69*, 94–106.

Junior, L. M., Nardini, F. M., Renso, C., Trani, R., & Macedo, J. A. (2020). A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems. *Expert Systems with Applications*, *152*, Article 113351.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *30*, 3146–3154.

Lasek, J., & Gagolewski, M. (2020). Interpretable sports team rating models based on the gradient descent algorithm. *International Journal of Forecasting*, *37*, 1061–1071.

Lee, T. S., Chiu, C. C., Lu, C. J., & Chen, I. F. (2002). Credit scoring using the hybrid neural discriminant technique. *Expert Systems with Applications*, *23*(3), 245–254.

Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, *247*(1), 124–136.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988).

Liu, W., Fan, H., & Xia, M. (2021). Step-wise multi-grained augmented gradient boosting decision trees for credit scoring. *Engineering Applications of Artificial Intelligence*, *97*, Article 104036.

Lundberg, S., & Lee, S. I. (2017). A unified approach to interpreting model predictions. arXiv preprint arXiv:1705.07874.

Maldonado, S., Bravo, C., López, J., & Pérez, J. (2017). Integrated framework for profit-based feature selection and SVM classification in credit scoring. *Decision Support Systems*, *104*, 113–121.

Maldonado, S., Pérez, J., & Bravo, C. (2017). Cost-based feature selection for support vector machines: An application in credit scoring. *European Journal of Operational Research*, *261*(2), 656–665.

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, *5*(9), 1315–1316.

Moreno-Torres, J. G., Sáez, J. A., & Herrera, F. (2012). Study on the impact of partition-induced dataset shift on *k*-fold cross-validation. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(8), 1304–1312.

Nalić, J., Martinović, G., & Žagar, D. (2020). New hybrid data mining model for credit scoring based on feature selection algorithm and ensemble classifiers. *Advanced Engineering Informatics*, *45*, Article 101130.

Niu, K., Zhang, Z., Liu, Y., & Li, R. (2020). Resampling ensemble model based on data distribution for imbalanced credit risk evaluation in P2P lending. *Information Sciences*, *536*, 120–134.

Pang, M., Ting, K. M., Zhao, P., & Zhou, Z. H. (2018). Improving deep forest by confidence screening. In *2018 IEEE international conference on data mining* (pp. 1194–1199). IEEE.

Papouskova, M., & Hajek, P. (2019). Two-stage consumer credit risk modelling using heterogeneous ensemble learning. *Decision Support Systems*, *118*, 33–45.

Shen, F., Zhao, X., Li, Z., Li, K., & Meng, Z. (2019). A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation. *Physica A*, *526*, Article 121073.

Sohn, S. Y., & Kim, J. W. (2012). Decision tree-based technology credit scoring for start-up firms: Korean case. *Expert Systems with Applications*, *39*(4), 4007–4012.

Sohn, S. Y., Kim, D. H., & Yoon, J. H. (2016). Technology credit scoring model with fuzzy logistic regression. *Applied Soft Computing*, *43*, 150–158.

Tsai, C. F., & Wu, J. W. (2008). Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, *34*(4), 2639–2649.

Tulio Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. arXiv e-prints, arXiv–1602.

Wang, G., Ma, J., Huang, L., & Xu, K. (2012). Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems*, *26*, 61–68.

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research*, *27*(11–12), 1131–1152.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.

Wu, Y., Liu, L., Xie, Z., Chow, K. H., & Wei, W. (2021). Boosting ensemble accuracy by revisiting ensemble diversity metrics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16469–16477).

Xia, Y., Liu, C., Da, B., & Xie, F. (2018). A novel heterogeneous ensemble credit scoring model based on bstacking approach. *Expert Systems with Applications*, *93*, 182–199.

Xia, Y., Liu, C., Li, Y., & Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, *78*, 225–241.

Xia, M., Tian, N., Zhang, Y., Xu, Y., & Zhang, X. (2020). Dilated multi-scale cascade forest for satellite image classification. *International Journal of Remote Sensing*, *41*(20), 7779–7800.

Xia, M., Wang, K., Song, W., Chen, C., Li, Y., et al. (2020). Non-intrusive load disaggregation based on composite deep long short-term memory network. *Expert Systems with Applications*, *160*, Article 113669.

Xia, Y., Zhao, J., He, L., Li, Y., & Niu, M. (2020). A novel tree-based dynamic heterogeneous ensemble method for credit scoring. *Expert Systems with Applications*, *159*, Article 113615.

Xia, Y., Zhao, J., He, L., Li, Y., & Yang, X. (2021). Forecasting loss given default for peer-to-peer loans via heterogeneous stacking ensemble approach. *International Journal of Forecasting*, *37*(1590–1613).

Xiao, J., Wang, Y., Chen, J., Xie, L., & Huang, J. (2021). Impact of resampling methods and classification models on the imbalanced credit scoring problems. *Information Sciences*, *569*, 508–526.

Xiao, H., Xiao, Z., & Wang, Y. (2016). Ensemble classification based on supervised clustering for credit scoring. *Applied Soft Computing*, *43*, 73–86.

Zhang, W., Yang, D., & Zhang, S. (2021). A new hybrid ensemble model with voting-based outlier detection and balanced sampling for credit scoring. *Expert Systems with Applications*, *174*, Article 114744.

Zhou, Z. H., & Feng, J. (2017). Deep forest. arXiv preprint arXiv:1702.08835.

Zhou, J., Li, W., Wang, J., Ding, S., & Xia, C. (2019). Default prediction in P2P lending from high-dimensional data based on machine learning. *Physica A*, *534*, Article 122370.