Expert Systems With Applications 174 (2021) 114744

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A new hybrid ensemble model with voting-based outlier detection and balanced sampling for credit scoring

Wenyu Zhang, Dongqi Yang, Shuai Zhang *

School of Information Management and Artificial Intelligence, Zhejiang University of Finance and Economics, Hangzhou 310018, China

ARTICLE INFO

Keywords:
Machine learning
Ensemble modeling
Outlier detection
Balanced sampling
Credit scoring

ABSTRACT

The credit scoring system has been revolutionized with the development of the financial system and has received increasing attention from the academia and industry. Artificial intelligence technology has reshaped credit scoring through predictive classification. In this study, a new hybrid ensemble model with voting-based outlier detection and balanced sampling is proposed to achieve superior predictive power for credit scoring. To avoid noise-filled data from misleading the classifier training, a new voting-based outlier detection method is proposed to enhance the classic outlier detection algorithms with the weighted voting mechanism and boost the outlier scores into the training set to form an outlier-adapted training set. To reduce the information loss caused by under-sampling when dealing with imbalanced data, a new bagging-based balanced sampling method is proposed to enhance the traditional under-sampling methods with the bagging strategy to obtain a balanced training set. To further improve the performance of the proposed model, a stacking-based ensemble modeling method is proposed to first perform parametrical optimization and then construct the stacking-based multi-stage ensemble model. Five datasets from the UC Irvine machine learning repository and five evaluation indicators were adopted to evaluate the model performance. The experimental results indicate the superior performance of the proposed model and prove its robustness and effectiveness.

1. Introduction

If the real economy were to be compared to muscles in the human body, the financial system would be akin to blood vessels as it provides liquidity support to the real economy. In the financial system, credit is more valuable than gold. A high probability of default (PD) will shake the market confidence in the financial system, which will ultimately lead to its structural collapse (Peihani, 2016). Therefore, controlling PD by developing a good credit scoring model through predictive classification essentially defends the financial system from financial risks (Zhang, He, & Zhang, 2018). If the PD of borrowers is predicted correctly through the credit scoring model, the losses of lenders could be minimized in a fiercely competitive market. Decades of credit scoring research have resulted in some classic prediction models (e.g., Altman, 1968; Ohlson, 1980) that have been developed to measure PD and financial status. With the efficient-market hypothesis (Fama, 1976) questioned in the increasingly complex economic environment, the classic credit scoring models are incapable of accurately predicting the PD in the real world. The rapid development and application of artificial intelligence technology, in particular, machine learning, has brought dawn to the above dilemma (Kirkos, 2015). However, the datasets collected from the real world for machine learning mostly contain noise-filled imbalanced data, which affect the performance of credit scoring models. Therefore, researchers have reached a consensus on the need to develop effective feature processing methods to improve the quality of the datasets for credit scoring models (Appiah, Chizema, & Arthur, 2015).

Noise or outliers are erroneous or biased data that affect the performance of credit scoring models (Wei, Yang, Zhang, & Zhang, 2019). Outliers would mislead the classifier training in the credit scoring model, thereby reducing the predictive accuracy of the model. Some classic outlier detection algorithms include the elliptic envelope algorithm (EE; Rousseeuw & Driessen, 1999), local outlier factor algorithm (LOF; Breunig, Kriegel, Ng, & Sander, 2000), one-class support vector machine algorithm (OCSVM; Manevitz & Yousef, 2001), and isolation forest algorithm (IF; Liu, Ting, & Zhou, 2008a). However, it is difficult to justify the selection of a single outlier detection algorithm. Therefore, combining the respective power of these classic outlier detection algorithms is worth exploring and a topic of interest in this study.

* Corresponding author.
E-mail address: zhangshuai@zufe.edu.cn (S. Zhang).

https://doi.org/10.1016/j.eswa.2021.114744
Received 10 August 2020; Received in revised form 2 December 2020; Accepted 13 February 2021
Available online 20 February 2021

On the other hand, data imbalance is widely present in real world datasets. The relative absence of minority class sample points would cause the classifier to learn less about their features, producing severe classifier bias against the minority class samples. Under-sampling methods, such as random under-sampling (Yen & Lee, 2006), BalanceCascade (Liu, Wu, & Zhou, 2008), and random under-sampling boosting (Seiffert, Khoshgoftaar, Van Hulse, & Napolitano, 2010) have been widely used to achieve data balance by deleting some majority class sample points, which would lead to information loss. Therefore, it is necessary to develop an effective balanced sampling method to reduce information loss.

Owing to their powerful performance, ensemble learning methods have been widely used in credit scoring models, such as gradient boosting decision tree (GBDT; Friedman, 2001), random forest (RF; Breiman, 2001), adaptive boosting (AdaBoost; Freund & Schapire, 1996), and extreme gradient boosting (XGBoost; Chen & Guestrin, 2016).

In this study, a new hybrid ensemble model with voting-based outlier detection and balanced sampling is proposed to achieve good predictive power for credit scoring. First, a new voting-based outlier detection method is proposed to enhance the classic outlier detection algorithms by integrating the outlier scores through the weighted voting mechanism (Schapire, 1990), and the outlier scores are boosted into the training set to form an outlier-adapted training set. Subsequently, a new bagging-based balanced sampling method is proposed to enhance the traditional under-sampling methods with the bagging strategy (Breiman, 1999) by splitting the sample points into several parallelized subsets, performing random under-sampling, and obtaining a balanced training set. Further, a stacking-based ensemble modeling method with parametrical optimization is proposed so that the parameters of the selected base classifiers are optimized adaptively. The optimized base classifiers are constructed as a stacking-based (Wolpert, 1992) multi-stage ensemble model. Finally, the trained ensemble model is applied to predict the test set, and the predicted result is returned with the soft voting mechanism (Littlestone & Warmuth, 1994).

The rest of the paper is organized as follows: in Section 2, previous literature related to the model is introduced; in Section 3, the proposed model is detailed; in Section 4, the experimental datasets, evaluation indicators, and parameter settings are introduced; the experimental results are analyzed in Section 5; in Section 6, conclusions are drawn and issues are discussed that should be noted in future research.

2. Related work

In this study, the proposed model mainly involves three technologies: outlier detection, balanced sampling, and ensemble modeling. These three aspects being important fields of credit scoring and machine learning, have attracted the attention of various scholars. In this section, representative works are reviewed.

2.1. Outlier detection

Outliers are generally thought to affect the predictive performance of machine learning models. Some classic outlier detection algorithms have previously been proposed to detect outliers in datasets. For example, Yu, Wang, Zhang, Wang, and Huang (2016) proposed an outlier detection method based on under-sampling with averaging, which could detect outliers in case of little or significant noises in output observations, and provide much better parameter estimation of models compared with that based on raw data.

In contrast to classic outlier detection algorithms, outlier detection algorithms based on ensemble learning technology (e.g., IF) have recently become a popular research field. In our previous work, Wei et al. (2019) proposed an IF-based noise detection approach to reduce the possibility of model overfitting; In our previous work, Zhang et al. (2020) proposed a bagging-based local outlier factor algorithm to identify the outliers and subsequently boost them back into the training set to form the outlier-adapted training set to enhance the outlier adaptability of base classifiers. However, there are different types of outliers contained in the same dataset generally, and any single outlier detection algorithm can only detect certain types of outliers contained in the dataset, while neglecting the other types of outliers, which would mislead the classifier training in the models. Besides, it is difficult for credit scoring models to adopt any proper outlier detection algorithm to deal with the different types of outliers contained in the unknown datasets in the real world. This necessitates the combination of the respective powers of different outlier detection algorithms.

The weighted voting mechanism (Schapire, 1990) has been considered as a solution to combine the respective power of different machine learning algorithms. For example, Alzubi, Alzubi, Tedmori, Rashaideh, and Almomani (2018) proposed a consensus-based combining method to adjust the weights iteratively after comparing all the classifiers' outputs, converging all the weights to a final set so that the combined output could reach the consensus and utilize the respective power of different classifiers. Inspired by the work of Alzubi et al. on voting-based classifier integration, this study proposes a new voting-based outlier detection method to enhance the classic outlier detection algorithms by integrating the outlier scores through the weighted voting mechanism, and boost the outlier scores into the training set to form an outlier-adapted training set. This could realize the respective advantages of different outlier detection algorithms.

2.2. Balanced sampling methods

Datasets from the real world are generally imbalanced, whereas many classifiers have a basic assumption that the data distribution is balanced (Sun, Wong, & Kamel, 2009). Therefore, when imbalanced datasets are directly used for classifier training, it is likely that this ideal assumption cannot be satisfied. The sampling methods, including over-sampling and under-sampling, have been recognized to transform an imbalanced dataset into a balanced dataset so that classifier training in the models can be free from imbalance bias. As an over-sampling method, the synthetic minority oversampling technique (SMOTE; Chawla et al., 2002) algorithm was proposed to increase the minority class sample points, thus enhancing the balance of data distribution. Kim, Jo, and Shin (2016) examined the effectiveness of a hybrid ensemble method to balance the proportion between minority and majority classes by combining the clustering technique and under-sampling.

The under-sampling method based on ensemble learning, for example, the BalanceCascade algorithm (Liu et al., 2008b) has been proven to be an effective method for solving the problem of data imbalance. In our previous work, He, Zhang, and Zhang (2018) extended the BalanceCascade algorithm to generate adjustable balanced subsets based on the imbalance ratios of the training data, which reduced the negative effect of imbalanced data and improved the comprehensive performance of the predictive model. However, the traditional under-sampling methods achieve data balance by removing the majority sample points in the data, which would inevitably cause information loss.

To reduce the information loss caused by the traditional under-sampling methods and handle the time complexity, a new bagging-based balanced sampling method is proposed to enhance the traditional under-sampling methods with the bagging strategy to obtain a balanced training set by splitting the sample points into several parallelized subsets and perform random under-sampling in these subsets, which are subsequently used to train the base classifiers.

2.3. Ensemble methods

In credit scoring, ensemble learning methods usually have better prediction performance than a single classifier. Bagging, boosting

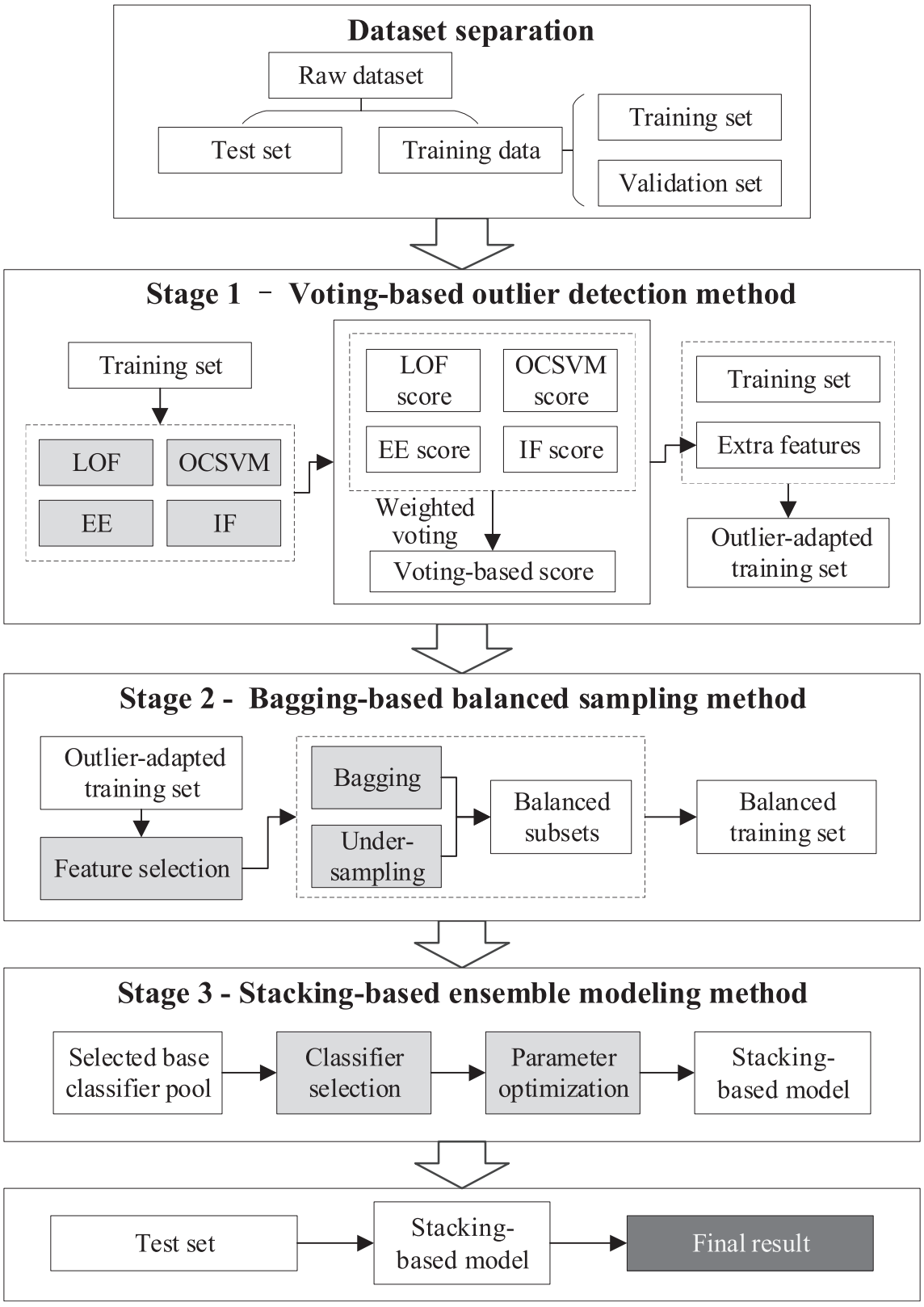

Fig. 1. Framework of the proposed model.

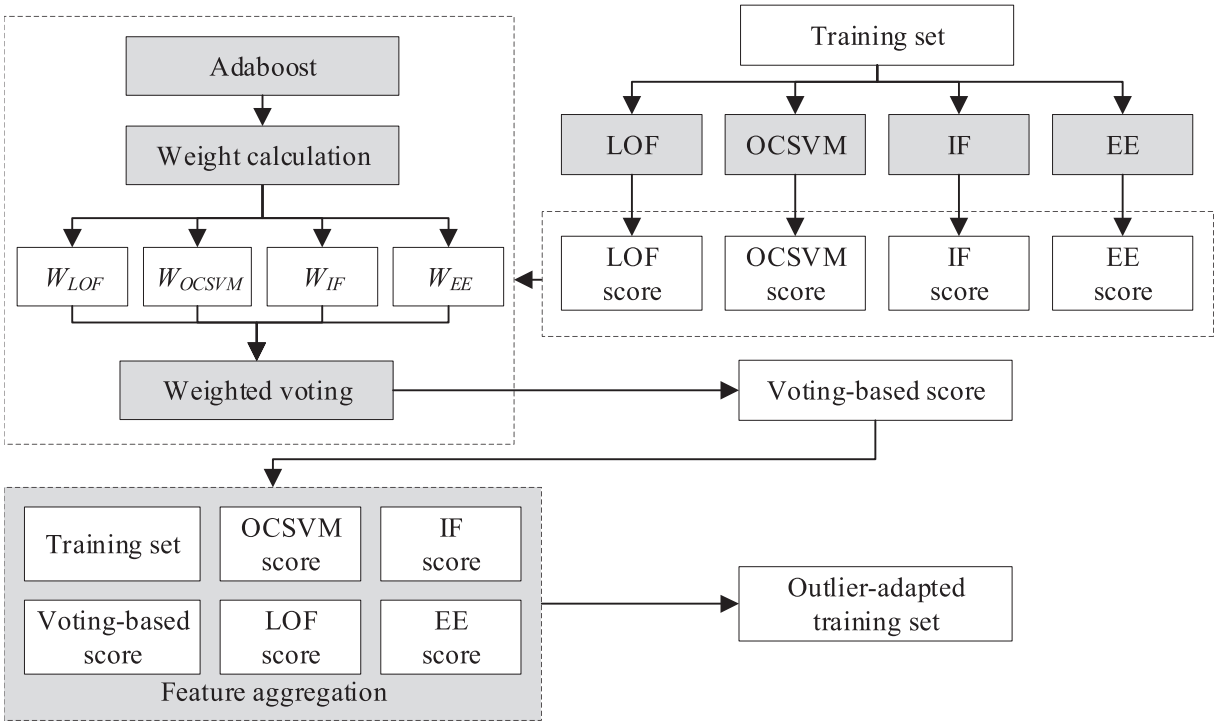

Fig. 2. Schematic diagram of the voting-based outlier detection method.

(Schapire, 1999), and stacking (Wolpert, 1992) are the mainstream research directions in ensemble learning. Among them, stacking has been widely adopted as an ensemble method with a flexible structure and high robustness to integrate the prediction results of base classifiers. For example, Fedorova, Gilenko, and Dovzhenko (2013) employed the stacking-based combinations of innovative learning algorithms to construct an effective ensemble model for bankruptcy prediction for Russian manufacturing companies. In stacking-based models, parametrical optimization significantly enhances the predictive performance of base classifiers, thereby enhancing the performance of the stacking-based models. Vukovic, Delibasic, Uzelac, and Suknovic (2012) applied the genetic algorithm to search for the optimal parameters to improve the model performance. Xia, Liu, Li, and Liu (2017) developed a hyper-parameter optimization technique to adaptively tune the hyper-

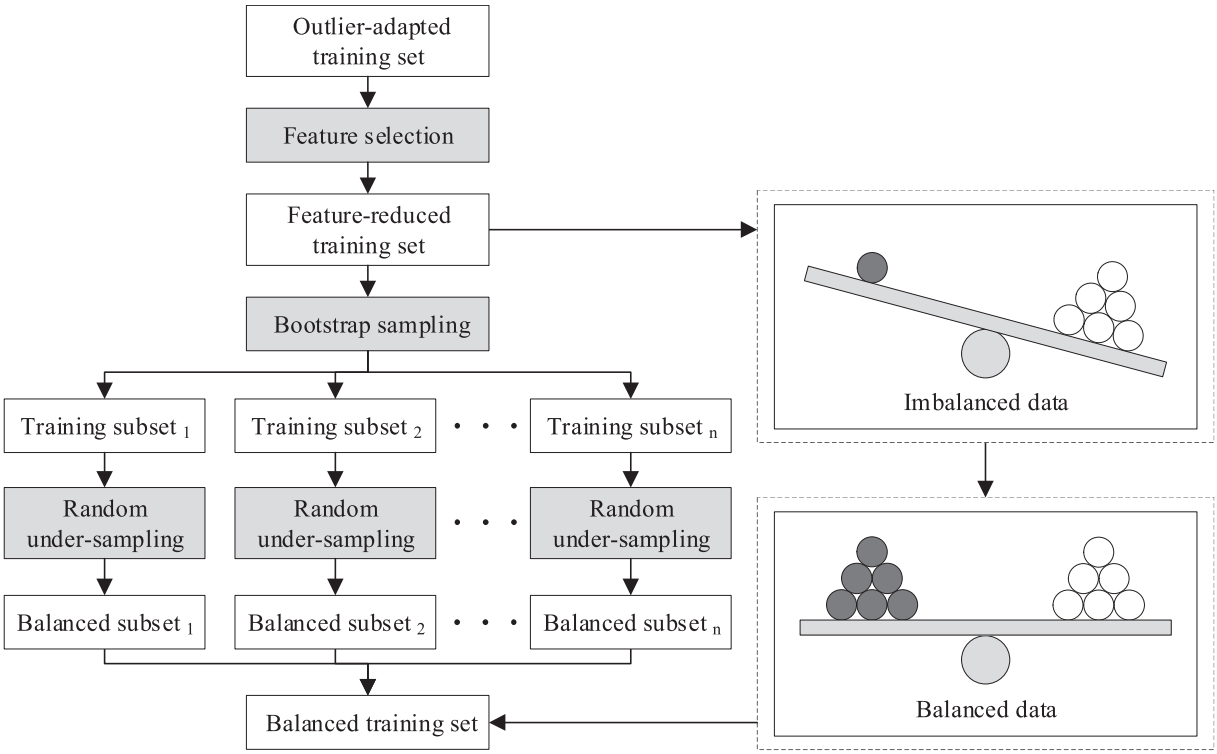

Fig. 3. Schematic diagram of the bagging-based balanced sampling method.

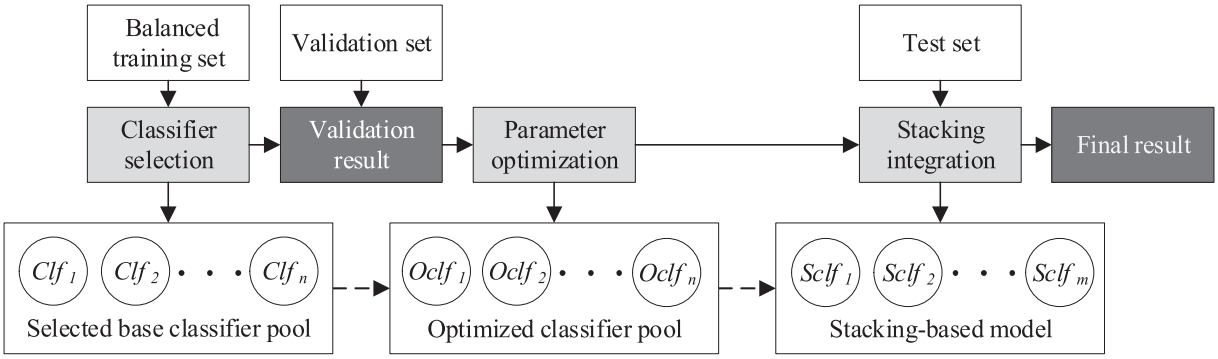

Fig. 4. Schematic diagram of the stacking-based ensemble modeling method.

Table 1
Description of the datasets.

Dataset	Sample size	Positive samples	Negative samples	Dimension of the input features (numerical/ categorical)
Australian	690	307	383	15 (6/9)
German	1000	700	300	21 (7/14)
Japanese	690	383	307	16 (5/11)
Taiwan	30,000	6636	23,364	24 (15/9)
Creator	35,960	14,001	21,959	61 (59/2)

parameters of classifiers to improve the model performance. In our previous work, He et al. (2018) developed a stacking-based ensemble model using a particle swarm optimization algorithm for parameter optimization of the base classifiers and proved the superiority of the proposed model when compared to other baseline ensemble learning algorithms in most evaluation measures on different datasets.

Stacking-based models consist of multiple base classifiers. However, tuning the parameters of all base classifiers using a fixed parametrical optimization technique leads to difficulty in balancing the performance of base classifiers, which affects the overall performance of the ensemble model. Therefore, this study proposes a stacking-based ensemble modeling method with self-adapted parametrical optimization based on the hyperparameter optimization framework (Komer, Bergstra, & Eliasmith, 2014) to optimize the parameters of the selected base classifiers adaptively to enhance the predictive performance of the ensemble model.

3. Modeling

In this work, a new hybrid ensemble model with voting-based outlier detection and balanced sampling for credit scoring is proposed. The raw dataset is separated into a training set, validation set, and test set. The framework is shown in Fig. 1. The proposed model consists of three main stages: voting-based outlier detection, bagging-based balanced sampling, and stacking-based ensemble modeling. The details of the three stages are illustrated in the following subsections.

3.1. Voting-based outlier detection method

Datasets from the real world generally involve noise-filled data, which have a high risk of misleading the classifier training. Four classic outlier detection algorithms, that is, EE, LOF, OCSVM, and IF, have been widely employed to handle the noise in the datasets. As depicted in Fig. 2, in the proposed voting-based outlier detection method, the outlier

Table 2
Best parameters of different models.

Base classifiers	Parameters	Australian	German	Japanese	Taiwan	Creator
XGBoost	learning_rate	0.05	0.06	0.05	0.05	0.05
	max_depth	19	7	13	18	5
	min_child_weight	6	2	1	4	1
	n_estimators	376	432	308	410	306
	subsample	0.5	0.9	0.9	0.5	0.9
GBDT	learning_rate	0.06	0.06	0.05	0.06	0.05
	max_depth	12	24	18	10	23
	n_estimators	526	572	452	438	510
	subsample	0.55	0.59	0.59	0.55	0.59
	loss	deviance	deviance	deviance	deviance	deviance
	presort	auto	auto	auto	auto	auto
AdaBoost	base_estimator	DT	DT	DT	DT	DT
	max_depth	9	9	16	14	11
	min_samples_split	25	20	16	35	28
	algorithm	SAMME	SAMME	SAMME	SAMME	SAMME
RF	n_estimators	261	241	354	361	121
	criterion	gini	gini	gini	gini	gini
	learning_rate	0.05	0.05	0.05	0.06	0.05
LR	penalty	L2	L2	L2	L2	L2
	tol	0.0001	0.0001	0.0001	0.0001	0.0001
Bagging	n_estimators	156	188	264	202	358
	max_samples	0.8	0.7	0.8	0.8	0.9
ExtraTree	n_estimators	128	156	184	254	386
	criterion	gini	gini	gini	gini	gini
	max_depth	11	16	13	26	8

Table 3
Baseline results.

Dataset	Base classifiers	ACC	AUC	F-score	Brier score	KS
Australian	XGBoost	0.87681	0.94425	0.85470	0.08344	0.33594
	GBDT	0.89130	0.94142	0.86957	0.08398	0.32813
	AdaBoost	0.87681	0.92334	0.85217	0.23611	0.33594
	RF	0.85507	0.91605	0.81818	0.10428	0.30469
	LR	0.87681	0.93902	0.85950	0.09382	0.40625
	Bagging	0.84783	0.92008	0.81416	0.10688	0.33594
	ExtraTree	0.86232	0.91420	0.82883	0.10348	0.33594
German	XGBoost	0.77000	0.82943	0.84321	0.15799	0.27564
	GBDT	0.75500	0.81974	0.83916	0.16090	0.24176
	AdaBoost	0.74500	0.78365	0.81455	0.24263	0.27564
	RF	0.72000	0.74476	0.80282	0.19380	0.08978
	LR	0.77500	0.82264	0.84642	0.15986	0.09357
	Bagging	0.70500	0.75217	0.78229	0.18475	0.27564
	ExtraTree	0.78000	0.81612	0.84058	0.15705	0.10642
Japanese	XGBoost	0.85507	0.94762	0.86111	0.08899	0.32000
	GBDT	0.88406	0.94730	0.88889	0.08775	0.36000
	AdaBoost	0.81884	0.93460	0.82759	0.23307	0.32667
	RF	0.86957	0.89937	0.88000	0.12058	0.42000
	LR	0.84783	0.90667	0.85906	0.12495	0.40667
	Bagging	0.86957	0.92889	0.87500	0.09746	0.31333
	ExtraTree	0.85507	0.92772	0.86301	0.10449	0.40000
Taiwan	XGBoost	0.81283	0.75758	0.45775	0.14199	0.12285
	GBDT	0.81867	0.78065	0.46562	0.13552	0.14157
	AdaBoost	0.81800	0.77598	0.44612	0.20313	0.14906
	RF	0.81550	0.76552	0.45700	0.13787	0.12397
	LR	0.80767	0.72560	0.38672	0.14527	0.13411
	Bagging	0.80383	0.73755	0.41964	0.15215	0.10300
	ExtraTree	0.81083	0.76090	0.44444	0.14058	0.12360
Creator	XGBoost	0.99708	0.99762	0.99633	0.00298	0.38281
	GBDT	0.99625	0.99792	0.99516	0.00360	0.37500
	AdaBoost	0.99680	0.99780	0.99598	0.17194	0.39062
	RF	0.99708	0.99735	0.99633	0.00305	0.38281
	LR	0.98165	0.99695	0.97688	0.01498	0.38281
	Bagging	0.99708	0.99797	0.99633	0.00325	0.38281
	ExtraTree	0.99458	0.99818	0.99320	0.00951	0.37500

scores of the training set (i.e., *LOF score*, *OCSVM score*, *IF score*, and *EE score*), which are obtained by the four classic outlier detection algorithms (i.e., *LOF*, *OCSVM*, *IF*, and *EE*) respectively were standardized and normalized to the same order of magnitude. Then the four individual outlier scores are used as features to train an Adaboost classifier (Freund & Schapire, 1996), which is subsequently validated with the predictive label of the validation set. The feature importance values, i.e., weights of the four individual outlier scores will be calculated by the Adaboost algorithm, which are represented as W_{LOF}, W_{OCSVM}, W_{IF}, and W_{EE}, respectively. The individual outlier score obtained from a certain outlier detection algorithm that is less relevant in context with the dataset will be assigned with a smaller weight. A voting-based outlier score is obtained by integrating the four individual outlier scores through weighted voting. The four individual outlier scores and the voting-based outlier score are then aggregated as extra features so that the respective power of different outlier detection algorithms can be leveraged for classifier training. They are boosted into the training set as five extra features to form an outlier-adapted training set to enhance the outlier adaptability of base classifiers.

3.2. Bagging-based balanced sampling method

The features of the outlier-adapted training set are not all correlated with the credit; therefore, only salient features need to be extracted through feature selection to form the feature-reduced training set. As depicted in Fig. 3, the feature selection approach based on the logistic regression penalty term (Andrew, 2004) is utilized to extract salient features from the outlier-adapted training set and obtain the feature-reduced training set. Because the feature-reduced training set may be an imbalanced dataset, it needs to be transformed into a balanced dataset through an under-sampling method. However, the traditional under-sampling methods achieve the data balance by simply removing the majority class sample points, which would inevitably cause information loss. Meanwhile, the time complexity of the under-sampling methods is also considerable. Therefore, a bagging-based balanced sampling method is proposed to enhance the traditional under-sampling methods using the bagging strategy. It performs parallelized under-sampling to divide the feature-reduced training set into several training subsets (e.g., *Training subset*$_1$, *Training subset*$_2$, etc.) through the bootstrap sampling technique of bagging. Random under-sampling is performed to transform the imbalanced training subsets into balanced training subsets (e.g., *Balanced subset* $_1$, *Balanced subset* $_2$, etc.). The balanced subsets are then aggregated into a balanced training set. Therefore, the proposed balanced sampling method not only handles time complexity but also reduces information loss.

3.3. Stacking-based ensemble modeling method

The balanced training set is used to train and validate the candidate base classifiers according to the area under the ROC curve (AUC; Hanley & McNeil, 1982) performance. Referring to Fig. 4, the base classifiers that perform better (e.g., Clf_1, Clf_2, …, Clf_n) are selected to form a selected base classifier pool. The parametrical optimization of the selected base classifiers could effectively enhance the performance of the classifiers. The hyperparameter optimization framework (Hyperopt; Komer et al., 2014) with self-adapted parametrical optimization is applied to optimize the parameters of the selected base classifiers adaptively based on the validation result, and obtain the optimized classifier pool that contains the optimized classifiers (e.g., $Oclf_1$, $Oclf_2$, …, $Oclf_n$). The optimized classifiers are permuted and combined to form

Table 4
Performance evaluation of the voting-based outlier detection method.

Dataset	Base classifiers	ACC	AUC	F-score	Brier score	KS
Australian	XGBoost	**0.89855**	**0.95035**	**0.88136**	**0.08237**	**0.37500**
	GBDT	**0.90580**	**0.94490**	**0.89076**	0.08447	**0.34375**
	AdaBoost	0.86232	**0.94578**	0.83761	**0.23292**	**0.35156**
	RF	**0.85507**	**0.92672**	**0.82143**	**0.10138**	**0.33594**
	LR	**0.88406**	**0.94142**	**0.86667**	**0.09241**	**0.42188**
	Bagging	**0.89130**	**0.93630**	**0.86726**	**0.08935**	0.29688
	ExtraTree	**0.86957**	**0.92030**	**0.83636**	**0.09877**	**0.35156**
German	XGBoost	**0.77500**	**0.82977**	**0.86986**	**0.15603**	**0.38258**
	GBDT	**0.77000**	**0.83378**	**0.85813**	**0.15248**	**0.42045**
	AdaBoost	**0.74500**	**0.80281**	**0.82230**	**0.24183**	**0.38258**
	RF	**0.75500**	**0.75635**	**0.82437**	**0.18275**	**0.44697**
	LR	**0.77500**	**0.83567**	0.81928	**0.15411**	**0.44697**
	Bagging	**0.79000**	**0.78855**	**0.84672**	**0.16805**	**0.37879**
	ExtraTree	0.73500	0.75986	0.80727	0.18555	**0.42803**
Japanese	XGBoost	**0.87681**	0.94751	**0.88112**	**0.08826**	**0.36667**
	GBDT	0.86957	**0.95101**	0.87500	**0.08458**	**0.36000**
	AdaBoost	**0.89130**	**0.94011**	**0.89655**	0.23554	**0.36000**
	RF	**0.87681**	**0.92487**	**0.88435**	**0.11391**	0.41333
	LR	**0.85507**	**0.90899**	**0.86486**	**0.12373**	**0.41333**
	Bagging	**0.87681**	**0.94550**	**0.88112**	**0.08297**	**0.37333**
	ExtraTree	**0.88406**	**0.94635**	**0.89041**	**0.09239**	**0.40667**
Taiwan	XGBoost	**0.81717**	**0.76573**	**0.45987**	**0.13934**	**0.14772**
	GBDT	**0.81867**	**0.78102**	0.46351	**0.13540**	**0.17053**
	AdaBoost	0.81667	**0.77840**	**0.45672**	**0.19974**	**0.15931**
	RF	**0.81583**	**0.76690**	**0.45974**	**0.13770**	**0.15370**
	LR	**0.80917**	**0.75381**	0.34132	**0.14103**	**0.15468**
	Bagging	**0.80500**	0.72769	**0.41991**	**0.15065**	**0.13500**
	ExtraTree	**0.81550**	**0.76319**	**0.46184**	**0.13936**	**0.16156**
Creator	XGBoost	**0.99778**	**0.99795**	**0.99714**	**0.00227**	**0.42734**
	GBDT	**0.99652**	0.99714	**0.99562**	**0.00345**	**0.49875**
	AdaBoost	**0.99778**	**0.99797**	**0.99714**	**0.16711**	**0.46800**
	RF	**0.99778**	**0.99836**	**0.99714**	**0.00240**	**0.42734**
	LR	**0.98331**	**0.99699**	**0.97937**	0.01522	**0.46800**
	Bagging	**0.99764**	**0.99839**	**0.99696**	**0.00270**	**0.49875**
	ExtraTree	**0.99513**	**0.99820**	**0.99376**	**0.00868**	**0.42734**

Note: significant values were boldfaced.

m candidate classifier ensembles (e.g., $Sclf_1$, $Sclf_2$, …, $Sclf_m$). The candidate classifier ensembles can be composed of several optimized base classifiers through exhaustive search. The total number of candidate classifier ensembles is represented as $m = \sum_{i=2}^{n} C_n^i$. The predicted results of the candidate classifier ensembles are tested on the test set, and the best-performed classifier ensemble is selected as the proposed model.

4. Experiment

4.1. Dataset description and data preprocessing

Four datasets from the UC Irvine (UCI) machine learning repository, that is, the Australian, German, Japanese (Asuncion & Newman, 2007), and Taiwan (Yeh & Lien, 2009) were adopted for the current study. Furthermore, Creator dataset was also adopted in the current study, which was published by a Chinese digital government services provider named Creator Information Technology Co., Ltd[1] in 2019. The Creator dataset includes the financial information of 35,960 Chinese companies. The details of these datasets are listed in Table 1.

The Australian credit dataset contains 690 samples, of which 307 are positive and 383 are negative. The dimension of the input features, including the class label, is 15, with six attributes being numerical and nine being categorical. The other datasets are self-explainable in Table 1 likewise. Basic data preprocessing approaches, including standardization, normalization, dummy coding, and correlation analysis, are applied to process these raw datasets. For example, numerical features are standardized and normalized by removing the mean and unit variance. Dummy coding was applied to transform continuous input variables into several dichotomous features. Correlation analysis is applied to remove one of the two features with a correlation larger than 0.97 (He et al., 2018; Wei et al., 2019).

4.2. Evaluation indicators

In this study, five evaluation indicators were adopted: accuracy (ACC; Stehman, 1997), AUC, F-score, Brier score (BRIER, 1950), and Kolmogorov–Smirnov rate (KS; Hodges, 1958). The evaluation indicators are determined by true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values. The predictive accuracy is defined in Eq. (1).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Precision is defined in Eq. (2), recall is defined in Eq. (3), and the F-score is defined in Eq. (4). The F-score is the harmonic average of precision and recall; its best value is one and its worst value is zero.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

[1] http://www.chinacreator.com/cn/

Table 5
Performance evaluation of the IF vs. the voting-based outlier detection method.

Dataset	Base classifiers	ACC	AUC	F-score	Brier score	KS
Australian	XGBoost	0.86957	0.94664	0.85246	0.08692	0.34375
	GBDT	0.85507	0.93002	0.83607	0.08756	0.33594
	AdaBoost	0.86130	0.93769	0.82189	0.25095	0.32281
	RF	**0.86957**	0.92611	**0.83246**	0.13557	0.32031
	LR	0.87754	0.92558	0.83188	**0.07279**	0.42188
	Bagging	0.84783	**0.94249**	0.82353	0.09232	0.28906
	ExtraTree	0.85130	0.90576	0.81805	0.09957	**0.35938**
German	XGBoost	0.74500	0.76811	0.82230	0.18502	0.33333
	GBDT	0.75000	0.78237	0.83301	0.15827	0.37847
	AdaBoost	0.72500	0.78113	0.79324	0.26411	0.37153
	RF	0.74500	**0.79421**	0.80469	0.19906	0.40972
	LR	0.76500	0.81262	0.79417	0.17741	0.42014
	Bagging	0.75500	0.78398	0.82437	0.18265	0.35417
	ExtraTree	**0.80000**	**0.79105**	0.79842	0.18967	0.41667
Japanese	XGBoost	0.84783	0.92254	0.85106	0.11593	0.32667
	GBDT	0.86232	0.94952	0.86525	0.08981	0.36000
	AdaBoost	0.86957	0.92847	0.87671	**0.22236**	0.35333
	RF	0.85580	0.90407	0.86156	0.14378	0.41333
	LR	0.84957	0.89111	**0.88312**	**0.11148**	0.41333
	Bagging	0.86130	0.90698	0.87510	0.09167	0.37333
	ExtraTree	0.85029	0.92857	**0.92715**	0.11797	**0.43333**
Taiwan	XGBoost	0.81667	0.74701	0.44962	0.15929	0.14211
	GBDT	0.81867	0.75168	0.44404	0.16524	0.16941
	AdaBoost	**0.81700**	0.75888	0.45590	0.20235	0.15707
	RF	0.80185	0.74798	0.43800	0.14754	0.15071
	LR	0.80817	0.74884	0.32001	0.14148	0.15093
	Bagging	0.80400	**0.72866**	0.40012	0.15154	**0.13650**
	ExtraTree	0.81267	0.76214	0.45331	0.13985	0.15856
Creator	XGBoost	0.99708	0.99764	0.99633	0.00295	0.41303
	GBDT	0.99499	0.99528	0.99369	0.00475	0.49544
	AdaBoost	0.99680	0.99762	0.99598	0.16856	0.41610
	RF	0.99708	0.99749	0.99633	0.00302	0.40183
	LR	**0.98345**	0.98702	0.96954	**0.01488**	0.45872
	Bagging	0.99708	0.99801	0.99633	0.00312	0.39153
	ExtraTree	0.99458	0.99776	0.99320	**0.00812**	0.38643

Note: significant values were boldfaced.

AUC is an indicator typically used in binary classification tasks, which is defined as the area enclosed by the ROC curve and coordinate axis. The value of this area is less than one. A classification model with a higher AUC value has a better classification ability than one with a lower value.

The Brier score, which measures the mean squared difference between the predicted probability and the actual label, is also adopted. The Brier score can be regarded as a loss function. The lower the Brier score, the better the predictive performance of the model.

The KS rate is used to measure the ability of a binary classification model to classify positive and negative samples. The higher the KS value, the better the performance of the model (He et al., 2018).

4.3. Experimental parameter settings

The raw dataset was divided as follows: 20% of the total data was used as the test set; the remaining 80% was used as training data, which was further separated into two parts: 80% was used as the training set and 20% was used as the validation set. The data preprocessing approaches were imported from the Python module "sklearn." In the proposed voting-based outlier detection method, classic outlier detection algorithms were imported from the Python module "sklearn." The default parameters of the classic outlier detection algorithms were adopted. In the proposed bagging-based balanced sampling method, the random under-sampling algorithm with default parameters was imported from the Python module "imbalanced-learn." In the proposed stacking-based ensemble modeling method, default parameters of the base classifiers were adopted before parametrical optimization via the Hyperopt framework. The base classifier XGBoost was imported from the Python module "xgboost"; the base classifiers GBDT, Adaboost, RF, Bagging (Breiman, 1996), extremely randomized trees (ExtraTree; Geurts, Ernst, & Wehenkel, 2006), and logistic regression (LR; Fan, Chang, Hsieh, Wang, & Lin, 2008) were imported from the Python module "sklearn." The Hyperopt framework was imported from the Python module "hyperopt." The base classifiers and their best corresponding parameter values on each dataset during ten times running are shown in Table 2.

5. Experimental analysis

In this study, seven base classifiers, that is, XGBoost, GBDT, Adaboost, RF, LR, Bagging, and ExtraTree, were evaluated and four base classifiers were selected for ensemble operation according to their superior performance on AUC indicator for each dataset. Five evaluation indicators, ACC, AUC, F-score, Brier score, and KS were adopted to evaluate the model performance of the baseline classifiers and ensemble models. To enhance the robustness of the experiments, each experiment was performed ten times and the average values were calculated for the validation evaluation. All the experiments were run on a PC running Python Version 3.7 with a 3.2 GHz Intel CORE i5 processor, 12 GB of RAM, and Microsoft Windows 10 operating system.

5.1. Baseline results

To verify the performance of the proposed model, the baseline results of each dataset were evaluated using five evaluation indicators. As shown in Table 3, seven base classifiers were applied, and the Australian, German, Japanese, Taiwan, and Creator credit datasets were

Table 6
Performance evaluation of the bagging-based balanced sampling method.

Dataset	Base classifiers	ACC	AUC	F-score	Brier score	KS
Australian	XGBoost	**0.90580**	0.94882	**0.88889**	**0.08063**	**0.51449**
	GBDT	**0.90580**	**0.94730**	0.88136	**0.07880**	**0.51668**
	AdaBoost	**0.86957**	0.93902	**0.85246**	**0.09624**	**0.51087**
	RF	**0.86957**	**0.93293**	**0.84746**	**0.10114**	**0.51449**
	LR	**0.89855**	**0.94382**	**0.87931**	**0.08284**	**0.50725**
	Bagging	**0.89130**	0.92334	**0.86957**	**0.08398**	**0.46800**
	ExtraTree	**0.86957**	**0.92944**	**0.85246**	**0.08759**	**0.50362**
German	XGBoost	**0.77500**	**0.83567**	**0.86993**	**0.15411**	**0.42045**
	GBDT	**0.77500**	**0.83623**	**0.86207**	0.16300	**0.44697**
	AdaBoost	**0.76500**	**0.83378**	**0.84321**	**0.16242**	**0.39773**
	RF	0.75000	**0.78715**	0.82394	**0.17130**	**0.45076**
	LR	**0.77500**	**0.83612**	**0.82353**	0.16577	0.44318
	Bagging	**0.79500**	**0.82787**	**0.85813**	**0.16241**	**0.39773**
	ExtraTree	**0.74500**	**0.80281**	**0.82230**	0.24183	**0.45076**
Japanese	XGBoost	**0.87681**	**0.95069**	**0.89362**	**0.08736**	**0.50725**
	GBDT	**0.89855**	**0.95111**	**0.88571**	0.08695	**0.49875**
	AdaBoost	**0.89130**	**0.94434**	0.89362	**0.08393**	**0.46800**
	RF	**0.87681**	**0.94434**	**0.88732**	**0.09522**	**0.49875**
	LR	**0.87681**	**0.95026**	**0.87943**	**0.08640**	**0.51449**
	Bagging	**0.89855**	**0.95005**	**0.89041**	0.08883	**0.51668**
	ExtraTree	**0.88406**	**0.94751**	0.88571	**0.08765**	**0.50000**
Taiwan	XGBoost	**0.81717**	**0.77349**	**0.48181**	**0.13823**	**0.15056**
	GBDT	**0.82167**	**0.79044**	**0.48159**	**0.13282**	**0.17315**
	AdaBoost	**0.82150**	**0.78812**	**0.47934**	0.20507	**0.16866**
	RF	**0.81650**	**0.77693**	**0.47346**	**0.13580**	**0.15482**
	LR	**0.81033**	0.73390	**0.36707**	0.14417	**0.17861**
	Bagging	**0.80700**	**0.73496**	**0.44910**	0.15194	0.13426
	ExtraTree	0.81517	**0.76916**	**0.47515**	**0.13763**	**0.16305**
Creator	XGBoost	**0.99778**	**0.99823**	**0.99714**	**0.00224**	**0.50725**
	GBDT	**0.99666**	**0.99714**	**0.99570**	**0.00324**	**0.50362**
	AdaBoost	**0.99778**	**0.99804**	**0.99714**	0.17107	**0.51449**
	RF	**0.99778**	0.99788	**0.99714**	**0.00238**	**0.52150**
	LR	**0.98512**	**0.99732**	**0.98118**	**0.01432**	**0.51449**
	Bagging	**0.99778**	0.99817	**0.99714**	**0.00265**	**0.50725**
	ExtraTree	0.99499	**0.99855**	0.99359	**0.00840**	**0.50000**

Note: significant values were boldfaced.

Table 7
Selected base classifiers for each dataset.

Dataset	XGBoost	GBDT	Adaboost	RF	LR	Bagging	ExtraTree
Australian	●	●	●		●		
German	●	●	●		●		
Japanese	●	●			●	●	
Taiwan	●	●	●	●			
Creator	●		●			●	●

Table 8
Composition of the best-performed classifier ensemble corresponding to the five datasets.

Dataset	The proposed model	XGBoost	GBDT	Adaboost	RF	LR	Bagging
Australian	*Eclf* 1	●				●	
German	*Eclf* 2		●	●			
Japanese	*Eclf* 3	●				●	●
Taiwan	*Eclf 4*	●	●		●		
Creator	*Eclf 5*	●		●			●

represented as “Australian”, “German”, “Japanese”, “Taiwan”, and “Creator”, respectively.

5.2. Performance evaluation of the voting-based outlier detection method

To prove the effectiveness of the proposed voting-based outlier detection method on the datasets, five evaluation indicators were adopted to evaluate the performance, as shown in Table 4. In the same dataset, the values of the evaluation indicators are shown in bold if the base classifiers performed better or the same after the voting-based outlier detection method was applied. It shows that the performance of most of the five evaluation indicators for each dataset was improved after the voting-based outlier detection method was performed, which indicates that the proposed voting-based outlier detection method could enhance the outlier adaptability of the base classifiers.

To prove the superiority of the voting-based outlier detection method over classic outlier detection algorithms (i.e., EE, LOF, OCSVM, and IF), the classic outlier detection algorithms are also applied on the same

Table 9
Final performance of the proposed model on five datasets.

Dataset	ACC	AUC	F-score	Brier score	KS
Australian	**0.90580**	0.94839	**0.88889**	**0.07635**	**0.52267**
German	**0.79500**	**0.83846**	0.82591	**0.15248**	**0.46276**
Japanese	**0.89855**	**0.95302**	**0.90141**	**0.08273**	**0.52150**
Taiwan	**0.82267**	**0.79373**	**0.48473**	0.13806	**0.17929**
Creator	**0.99778**	0.99843	**0.99714**	**0.00224**	**0.52237**

Note: significant values were boldfaced.

dataset. As an illustrative example, the performance evaluation of the IF vs. the voting-based outlier detection method is shown in Table 5. In Table 5, the values of five evaluation indicators are shown in bold if the base classifiers perform better by employing IF than by employing the voting-based outlier detection method. It shows that IF performs worse than the voting-based outlier detection method on most of the five evaluation indicators. The experiment also shows that the other three classic outlier detection algorithms (i.e., EE, LOF, and OCSVM) perform worse than the voting-based outlier detection method, and their performance evaluation results have been uploaded to Figshare (https://doi.org/10.6084/m9.figshare.13317641) due to space limit of the paper.

5.3. Performance evaluation of the bagging-based balanced sampling method

To prove the effectiveness of the proposed bagging-based balanced sampling method on the datasets, five evaluation indicators were adopted to evaluate the performance on the datasets. As shown in Table 6, in the same dataset, the values of the evaluation indicators are shown in bold if the base classifiers performed better or the same after the bagging-based balanced sampling method was applied. The performance on most of the five evaluation indicators for each dataset was improved after the bagging-based balanced sampling was performed, indicating the effectiveness of the bagging-based balanced sampling method.

5.4. Performance evaluation of the stacking-based ensemble modeling method

The base classifiers that perform better for each dataset after both voting-based outlier detection and bagging-based balanced sampling methods were performed, were selected to form a selected base classifier pool, as shown in Table 7.

Then, the parameters of the selected base classifiers were optimized using the Hyperopt framework. These optimized base classifiers made up the optimized classifier pool that was used for classifier permutation and combination to form candidate classifier ensembles. As shown in Table 8, the best-performed classifier ensembles corresponding to the five datasets are represented as *Eclf* 1, *Eclf* 2, *and Eclf* 3, respectively. For example, *Eclf* 1 is an ensemble of XGBoost and LR.

Table 9 presents the final performance of the proposed ensemble model. In the same dataset, the values of the evaluation indicators are shown in bold if the proposed ensemble model performs better than or the same as the best-performed base classifier after both voting-based outlier detection and bagging-based balanced sampling methods were performed. This shows that the performance of the proposed ensemble model was superior to most of the evaluation indicators for each dataset. This indicates that the stacking-based ensemble modeling method could strengthen the predictive power of the proposed model, and therefore could predict the PD of borrowers so that lenders can minimize losses in the fiercely competitive market.

5.5. Performance comparison between the proposed ensemble model and benchmark ensemble models

A performance comparison between the proposed ensemble model and benchmark ensemble models proposed by Ala'raj & Abbod (2016), Abellán and Castellano (2017), He et al. (2018), García, Marqués, and Sánchez (2019), Shen, Zhao, Li, Li, and Meng (2019), Zhang, He, and Zhang (2019), and Xiao et al. (2020) are presented in Table 10. The Australian and German credit datasets were used by all the above seven benchmark ensemble models for model evaluation. The Japanese credit dataset has been used by the benchmark ensemble models proposed by Ala'raj and Abbod (2016), Abellán and Castellano (2017), He et al. (2018), García et al. (2019), and Zhang et al. (2019) for model evaluation. As shown in Table 10, in the same dataset, the values of evaluation

Table 10
Performance comparison between the proposed model and benchmark ensemble models.

Dataset	Benchmark ensemble models	ACC	AUC	F-score	Brier score	KS
Australian	Ala'raj & Abbod (2016)	0.87980	0.94040	/	0.09200	/
	Abellán and Castellano (2017)	0.86810	0.93210	/	/	/
	He et al. (2018)	/	0.93404	0.85020	/	0.76953
	García et al. (2019)	/	0.93600	/	/	/
	Shen et al. (2019)	/	0.93880	/	/	/
	Zhang et al. (2019)	0.87540	0.93700	/	0.09380	/
	Xiao et al. (2020)	0.86890	0.91280	/	/	/
	The proposed model	**0.90580**	**0.94839**	**0.88889**	**0.07635**	0.52267
German	Ala'raj & Abbod (2016)	0.77720	0.80230	/	0.15770	/
	Abellán and Castellano (2017)	0.77400	0.79370	/	/	/
	He et al. (2018)	/	0.80021	0.84439	/	0.49321
	García et al. (2019)	/	0.79400	/	/	/
	Shen et al. (2019)	/	0.81020	/	/	/
	Zhang et al. (2019)	0.76820	0.80290	/	0.16030	/
	Xiao et al. (2020)	0.73760	0.75610	/	/	/
	The proposed model	**0.79500**	**0.83846**	0.82591	**0.15248**	0.46276
Japanese	Ala'raj & Abbod (2016)	0.87880	0.93280		0.09460	/
	Abellán and Castellano (2017)	0.87750	0.93490	/	/	/
	He et al. (2018)	/	0.93058	0.87004	/	0.75942
	García et al. (2019)	/	0.93600	/	/	/
	Zhang et al. (2019)	0.87200	0.93870	/	0.09470	/
	The proposed model	**0.89855**	**0.95302**	**0.90141**	**0.08273**	0.52150

Note: significant values were boldfaced; "/" indicates that the corresponding evaluation indicators were not presented in the previous work.

Table 11
Significance test results of the classifier ranking with Friedman test.

Method	Classifier	ACC	AUC	F-score	Brier score	KS	AvgRank
Baseline	XGBoost	13.70	11.00	12.70	9.40	18.90	13.14
	GBDT	10.00	9.90	9.70	8.30	18.70	11.32
	AdaBoost	14.20	14.90	17.00	21.60	16.60	16.86
	RF	16.20	19.00	16.00	14.40	17.90	16.70
	LR	16.30	17.50	16.20	15.00	16.00	16.20
	Bagging	18.60	17.30	18.10	16.40	19.30	17.94
	ExtraTree	15.30	15.00	16.80	13.60	18.50	15.84
Voting-based outlier detection method	XGBoost	7.30	8.30	6.70	6.40	13.70	8.48
	GBDT	9.90	7.80	9.00	5.70	11.00	8.68
	AdaBoost	10.90	10.20	10.80	20.70	12.70	13.06
	RF	12.40	14.00	11.90	12.20	11.50	12.40
	LR	15.10	14.40	17.80	13.30	9.20	13.96
	Bagging	10.50	13.00	10.90	10.20	15.50	12.02
	ExtraTree	14.40	13.80	13.90	14.40	10.80	13.46
Bagging-based balancedsampling method	XGBoost	6.70	4.90	2.70	4.80	6.90	5.20
	GBDT	5.60	5.90	6.80	6.20	4.70	5.84
	AdaBoost	8.10	8.70	6.80	14.20	6.60	8.88
	RF	11.50	13.10	9.70	10.60	4.70	9.92
	LR	12.30	10.80	15.40	11.00	4.30	10.76
	Bagging	7.20	12.10	7.40	10.50	8.80	9.20
	ExtraTree	14.50	9.80	12.50	11.50	5.70	10.80
Stacking-based ensemble modeling method	The proposed model	**2.30**	**1.60**	**2.20**	**2.60**	**1.00**	**2.34**
Statistics of the Friedman test		83.75	82.44	83.09	73.75	53.19	
Chi-square critical value		9.49	9.49	9.49	9.49	9.49	
p-value		2.79E-17	5.31E-17	3.85E-17	3.67E-15	7.77E-11	

Table 12
Significance test results of the AUC ranking with the Friedman test on the proposed model and the benchmark models.

Classifier	AUC
Ala'raj & Abbod (*Bclf 1*; 2016)	3.66667
Abellán & Castellano (*Bclf 2*; 2017)	6.00000
He et al. (*Bclf 3*; 2018)	5.66667
García et al. (*Bclf 4*; 2019)	4.66667
Shen et al. (*Bclf 5*; 2019)	4.16667
Zhang et al. (*Bclf 6*; 2019)	3.00000
Xiao et al. (*Bclf 7*; 2020)	7.83333
The proposed model (*Eclf*)	**1.00000**
Statistics of the Friedman test	7.31250
Chi-square critical value	5.99146
p-value	0.02583

indicators are shown in bold if the proposed ensemble model performs better than the benchmark ensemble models. The results indicate the superior performance of the proposed model.

5.6. Statistical test results

Table 11 shows the significance test results of the classifier ranking with the Friedman test (Friedman, 1940). The ranked classifiers include the base classifiers before and after the voting-based outlier detection and bagging-based balanced sampling are applied, and the proposed model. The Friedman test is a nonparametric statistical test to rank the classifiers based on predictive performance. According to Lessmann, Baesens, Seow, and Thomas (2015), the individual ranking and average ranking (AvgRank) were calculated. The ranking values of the proposed model were shown in bold. Table 11 also depicts the statistic, Chi-square critical value, and p-value on each indicator. The alpha value is 0.05, which indicates a 5% risk of concluding that a difference exists when there is no actual difference. The statistics of the Friedman test on each indicator are larger than the Chi-square critical value, and the p-value is less than the alpha value. So, the null hypothesis is rejected, which proves the superior performance of the proposed model.

Table 12 shows the significance test results of the AUC ranking with the Friedman test on the proposed model and the benchmark models. In Table 12, the proposed model is represented as *Eclf* and the benchmark models are represented as *Bclf 1* to *Bclf* 7, respectively. The individual AUC ranking values of the benchmark models were calculated according to Lessmann et al. (2015) and the ranking value of the proposed model was shown in bold. Table 12 also depicts the statistic, Chi-square critical value, and p-value of the Friedman test on AUC indicator. The alpha value is 0.05, the statistics of the Friedman test on the AUC indicator are larger than the Chi-square critical value, and the p-value is less than the alpha value. So, the null hypothesis is rejected, which proves the superior performance of the proposed model over the benchmark models.

Finally, the Nemenyi post hoc test (Nemenyi, 1963) was applied to compare the performance of the proposed model with those of the base classifiers and the benchmark models. The graphical presentation of the Nemenyi post hot test results is shown in Fig. 5 and the critical distance (CD) indicates the mean ranking score difference (Demšar, 2006). The more the position of the classifier on the coordinate axis is to the left, the better the performance of the classifier, and vice versa. The graphical presentation reveals that the proposed model (i.e. *Eclf*) is superior to all base classifiers and the benchmark models on evaluation indicators.

6. Conclusion & future work

In this study, a new hybrid ensemble model with voting-based outlier detection and balanced sampling for credit scoring is proposed, which improves the adaptability to outliers and imbalanced data to obtain superior predictive performance through two significant contributions. First, the classic outlier detection algorithms are enhanced by integrating the outlier scores through the weighted voting mechanism to form an outlier-adapted training set. Second, the traditional undersampling approach is enhanced by performing random undersampling on the parallelized subsets produced through the bagging strategy to obtain a balanced training set. Five datasets were adopted for model performance evaluation through five evaluation indicators: ACC, AUC, F-score, Brier score, and KS. The experimental results demonstrate the superior performance of the proposed model over the benchmark models.

Nowadays, artificial intelligence algorithms are ubiquitous and have achieved superior performance on various tasks ranging from data mining and pattern recognition (Thomas et al., 2019). As the predictive power of artificial intelligence algorithms increases, the requirements for their safe and responsible use should also be considered

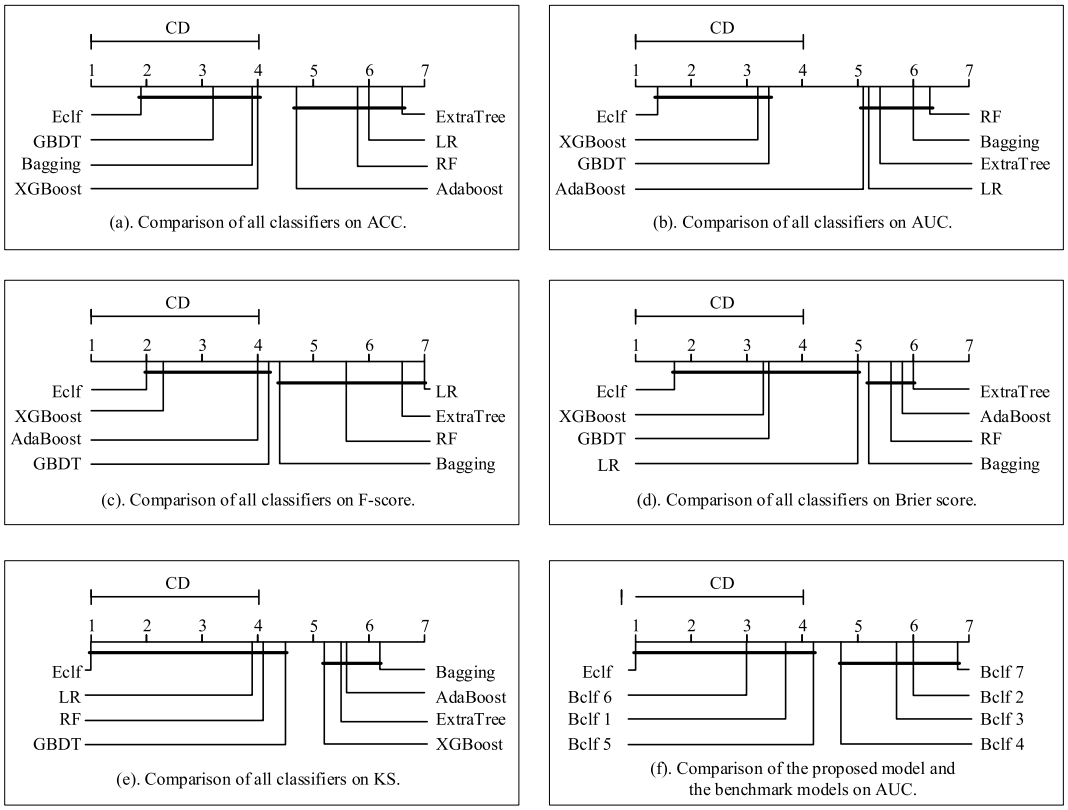

Fig. 5. Graphical presentation of the Nemenyi test results.

simultaneously. In future studies, the possible undesirable behaviors, and discriminations of artificial intelligence algorithms to human beings should be considered and prevented properly.

CRediT authorship contribution statement

Wenyu Zhang: Conceptualization, Methodology, Formal analysis, Writing - review & editing, Supervision, Funding acquisition. **Dongqi Yang:** Methodology, Formal analysis, Writing - original draft, Data curation, Software, Validation. **Shuai Zhang:** Conceptualization, Writing - review & editing, Supervision, Funding acquisition.

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.

Acknowledgements

The work has been supported by National Natural Science Foundation of China (No. 51875503, No. 51975512), Zhejiang Natural Science Foundation of China (No. LZ20E050001), Zhejiang Key R & D Project of China (No.2021C03153).

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. https://doi.org/10.1016/j.eswa.2016.12.020

Ala'raj, M., & Abbod, M. F. (2016). Classifiers consensus system approach for credit scoring. *Knowledge-Based Systems, 104*, 89–105. https://doi.org/10.1016/j.knosys.2016.04.013

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance, 23*(4), 589–609. https://doi.org/10.1111/j.1540-6261.1968.tb00843.x

Alzubi, O., Alzubi, J., Tedmori, S., Rashaideh, H., & Almomani, O. (2018). Consensus-based combining method for classifier ensembles. *International Arab Journal of Information Technology, 15*(1), 86–95.

Andrew, Y. N. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. In Proceedings of the 21st International Conference on Machine Learning, Banff, Alberta, Canada, pp. 78-86, July 4-8, 2004.

Appiah, K. O., Chizema, A., & Arthur, J. (2015). Predicting corporate failure: a systematic literature review of methodological issues. International Journal of Law and Management, 57(5), 461-485.

Asuncion, A., & Newman, D. (2007). *UCI Machine Learning Repository*. Irvine, CA: School of Information and Computer Science, University of California. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140. https://doi.org/10.1007/BF00058655

Breiman, L. (1999). Pasting small votes for classification in large databases and on-line. *Machine Learning, 36*, 85–103.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, pp. 93-104, May 15-18, 2000.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review, 78*(1), 1–3.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.

Chen, T. Q., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA, pp. 785-794, August 13-17, 2016.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Fama, E. F. (1976). Efficient Capital Markets: Reply. *The Journal of Finance, 31*(1), 143. https://doi.org/10.2307/2326404

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research, 9*, 1871–1874.

Fedorova, E., Gilenko, E., & Dovzhenko, S. (2013). Bankruptcy prediction for Russian companies: Application of combined classifiers. *Expert Systems with Applications, 40*(18), 7285–7293. https://doi.org/10.1016/j.eswa.2013.07.032

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, pp. 148-156, July 3-6, 1996.

Friedman, J. H. (2001). machine. *The Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Friedman, M. (1940). A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *The Annals of Mathematical Statistics, 11*(1), 86–92. https://doi.org/10.1214/aoms/1177731944

García, V., Marqués, A. I., & Sánchez, J. S. (2019). Exploring the synergetic effects of sample types on the performance of ensembles for credit risk and corporate bankruptcy prediction. *Information Fusion, 47*, 88–101. https://doi.org/10.1016/j.inffus.2018.07.004

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning, 63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*(1), 29–36. https://doi.org/10.1148/radiology.143.1.7063747

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. https://doi.org/10.1016/j.eswa.2018.01.012

Hodges, J. L. (1958). The significance probability of the smirnov two-sample test. *Arkiv fiur Matematik, 3*(5), 469–486. https://doi.org/10.1007/BF02589501

Kim, H.-J., Jo, N.-O., & Shin, K.-S. (2016). Optimization of cluster-based evolutionary undersampling for the artificial neural networks in corporate bankruptcy prediction. *Expert Systems with Applications, 59*, 226–234. https://doi.org/10.1016/j.eswa.2016.04.027

Kirkos, E. (2015). Assessing methodologies for intelligent bankruptcy prediction. *Artificial Intelligence Review, 43*(1), 83–123. https://doi.org/10.1007/s10462-012-9367-6

Komer, B., Bergstra, J., & Eliasmith, C. (2014). Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In Proceedings of the 13th Python in Science Conference, Austin, Texas, USA, pp. 32-37, July 6-12, 2014.

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. https://doi.org/10.1016/j.ejor.2015.05.030

Littlestone, N., & Warmuth, M. K. (1994). The Weighted Majority Algorithm. *Information and Computation, 108*(2), 212–261. https://doi.org/10.1006/inco.1994.1009

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008a). Isolation forest. In Proceedings of the 8th IEEE International Conference on Data Mining, Pisa, Italy, pp. 413-422, December 15-19, 2008.

Liu, X. Y., Wu, J., & Zhou, Z. H. (2008). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39*(2), 539–550.

Manevitz, L. M., & Yousef, M. (2001). One-class SVMs for document classification. *Journal of Machine Learning Research, 2*, 139–154.

Nemenyi, P. B. (1963). *Distribution-free multiple comparisons*. PhD thesis. Princeton University.

Ohlson, J. A. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research, 18*(1), 109. https://doi.org/10.2307/2490395

Peihani, M. (2016). Basel committee on banking supervision. *Brill Research Perspectives in International Banking & Securities Law, 89*(1), 335–347.

Rousseeuw, P. J., & Driessen, K. V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics, 41*(3), 212–223. https://doi.org/10.1080/00401706.1999.10485670

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5*(2), 197–227. https://doi.org/10.1007/BF00116037

Schapire, R. E. (1999). A brief introduction to boosting. In Proceedings of the 16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, pp. 1401-1406, July 31-August 6, 1999.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 40*(1), 185–197. https://doi.org/10.1109/TSMCA.2009.2029559

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

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment, 62*(1), 77–89. https://doi.org/10.1016/S0034-4257(97)00083-7

Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence, 23(4), 687-719.

Thomas, P. S., da Silva, B. C., Barto, A. G., Giguere, S., Brun, Y., & Brunskill, E. (2019). Preventing undesirable behavior of intelligent machines. *Science, 366*(6468), 999–1004.

Vukovic, S., Delibasic, B., Uzelac, A., & Suknovic, M. (2012). A case-based reasoning model that uses preference theory functions for credit scoring. *Expert Systems with Applications, 39*(9), 8389–8395. https://doi.org/10.1016/j.eswa.2012.01.181

Wei, S., Yang, D., Zhang, W., & Zhang, S. (2019). A Novel Noise-Adapted Two-Layer Ensemble Model for Credit Scoring Based on Backflow Learning. *IEEE Access, 7*, 99217–99230. https://doi.org/10.1109/ACCESS.2019.2930332

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

Xia, Y., Liu, C., Li, Y. Y., & Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications, 78*, 225–241. https://doi.org/10.1016/j.eswa.2017.02.017

Xiao, J., Zhou, X.u., Zhong, Y.u., Xie, L., Gu, X., & Liu, D. (2020). Cost-sensitive semi-supervised selective ensemble model for customer credit scoring. *Knowledge-Based Systems, 189*, 105118. https://doi.org/10.1016/j.knosys.2019.105118

Yeh, I.-C., & Lien, C.-H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications, 36*(2), 2473–2480. https://doi.org/10.1016/j.eswa.2007.12.020

Yen, S. J., & Lee, Y. S. (2006). Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In Proceedings of the 2nd International Conference on Intelligent Computing, Kunming, China, pp. 731-740, August 16-19, 2006.

Yu, C., Wang, Q.-G., Zhang, D., Wang, L., & Huang, J. (2016). System identification in presence of outliers. *IEEE Transactions on Cybernetics, 46*(5), 1202–1216. https://doi.org/10.1109/TCYB.2015.2430356

Zhang, H., He, H., & Zhang, W. (2018). Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring. *Neurocomputing, 316*, 210–221. https://doi.org/10.1016/j.neucom.2018.07.070

Zhang, W., He, H., & Zhang, S. (2019). A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring. *Expert Systems with Applications, 121*, 221–232. https://doi.org/10.1016/j.eswa.2018.12.020

Zhang, W. Y., Yang, D. Q., Zhang, S., Ablanedo-Rosas, J. H., Wu, X., & Yu, L. (2020). A novel multi-stage ensemble model with enhanced outlier adaptation for credit scoring. *Expert Systems with Applications, 165*, Article 113872.