IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 19, 2021, accepted October 3, 2021, date of publication October 14, 2021, date of current version October 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120086

A Novel Multi-Stage Ensemble Model With a Hybrid Genetic Algorithm for Credit Scoring on Imbalanced Data

YILUN JIN[1], WENYU ZHANG[1], XIN WU[2], YANAN LIU[1], AND ZEQIAN HU[1]
[1]School of Information Management and Artificial Intelligence, Zhejiang University of Finance and Economics, Hangzhou 310018, China
[2]China Academy of Financial Research, Zhejiang University of Finance and Economics, Hangzhou 310018, China

Corresponding author: Xin Wu (wuxin@zufe.edu.cn)

This work was supported by Zhejiang Natural Science Foundation of China under Grant LZ20E050001.

ABSTRACT Credit scoring models are the cornerstone of the modern financial industry. After years of development, artificial intelligence and machine learning have led to the transformation of traditional credit scoring models based on statistics. In this study, a novel multi-stage ensemble model with a hybrid genetic algorithm is proposed to achieve accurate and stable credit prediction. To alleviate the adverse effects of imbalanced data in credit scoring models, the Instance Hardness Threshold method is extended using a majority voting strategy to deal with data imbalance. To eliminate redundant and irrelevant features in the dataset and select well-performing base classifiers, a new hybrid genetic algorithm is proposed to obtain the optimal feature subset and base classifier subset. To aggregate the predictive power of the base classifiers, a stacking approach is adopted to integrate the optimal base classifiers into the ensemble model. The proposed model is tested on three standard imbalanced credit scoring datasets, compared with similar state-of-the-art approaches, and evaluated using four well-known evaluation indicators. The experimental results prove the effectiveness of the proposed model and demonstrate its superiority.

INDEX TERMS Credit scoring, imbalanced data, genetic algorithm, ensemble model.

NOMENCLATURE

ABBREVIATION

RF	Random forest.
XGBoost	Extreme gradient boosting.
GBDT	Gradient boost tree.
LDA	Linear discriminant analysis.
HGA	Hybrid genetic algorithm.
GA	Genetic algorithm.
IHT	Instance Hardness Threshold.
SMOTE	Synthetic minority oversampling technique.
IHP	Instance Hardness Points.
SMAC	Sequential model-based algorithm configuration.
VIHT	Voting Instance Hardness Threshold.
LR	Logistic regression.
DT	Decision tree.
ET	Extra trees.
LGBM	Light gradient boosting machine.
BACC	Balance accuracy.
TP	True positive.
FP	False positive.
TN	True negative.
FN	False negative.
TPR	True positive rate.
TNR	True negative rate.
CD	Critical distance.

INDICES

i	Classifier index.
g	Generation index.

VARIABLES AND PARAMETERS

Clf_i	The ith classifier.
$IHP\text{-}Clf_i$	The IHP identified by Clf_i.
n	The number of classifiers.
G	The maximum number of generations.
Q	The number of individuals.
Mr_g	The mutation rate in the gth generation.

The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko.

I. INTRODUCTION

The ability to accurately assess the creditworthiness of customers who apply for loans and perform corresponding risk management is the key to the development of the modern financial industry. With economic development, traditional statistics-based credit scoring models have been gradually overwhelmed by the exponentially growing credit big data, and have lost their effectiveness. An intuitive example is that traditional statistics-based credit scoring models require assumptions regarding the statistical distribution of data, which are often not applicable to big data with a complex distribution [54]. Recently, the advancements in artificial intelligence, such as ensemble learning-based methods [32], evolution algorithm-based methods [49], and clustering technique-based methods [28] have been used in credit scoring fields. In our previous study, artificial intelligence and machine learning technologies outperformed statistical approaches in constructing a credit scoring model [60].

Credit scoring data are usually imbalanced data, which means that the numbers of positive and negative samples in the data are inconsistent. In the imbalanced credit scoring data, positive samples refer to the number of defaulting customers, and negative samples refer to the number of non-defaulting customers. Generally, the number of negative samples is much larger than the number of positive samples. The rationale behind this phenomenon is that, in most real-world cases, the number of customers who pay their bills on time is much larger than the number of customers who default. However, both statistics-based and machine learning-based credit scoring models find making accurate predictions challenging when imbalanced data are directly input. Therefore, enhancing the predictive ability of credit scoring models using imbalanced data is the first motivation of this study.

Credit scoring data are also high-dimensional [47]. The feature relations of high-dimensional data are often complex, which makes it difficult to predict the probability of default. Furthermore, redundant or irrelevant features often lead to model overfitting, which affects model performance. Hence, developing an effective feature selection approach is a prerequisite to lower data processing costs, a better understanding of data, and better-performing credit scoring models.

In addition, an appropriate ensemble strategy that integrates multiple base classifiers into an ensemble model, has proven to be an effective approach for solving several data mining tasks [26]. Therefore, various ensemble models based on random forest (RF) [7], extreme gradient boosting (XGBoost) [13], gradient boost tree (GBDT) [19], linear discriminant analysis (LDA) [18], etc., have been utilized for credit scoring. However, it is not taken for granted that an ensemble model composed of random multiple weak classifiers will be a well-performing strong classifier. In particular, multiple poorly-performing or correlated base classifiers in an ensemble model may result in adverse ensemble effects. To overcome these limitations, well-performing and uncorrelated base classifiers must be selected to construct ensemble models. Unfortunately, classifier selection is as complex as feature selection. Therefore, developing an effective classifier selection approach for selecting and composing well-performing base classifiers is another problem worth exploring.

The main contributions of this study can be summarized as follows:

1) A novel multi-stage ensemble model with a hybrid genetic algorithm (HGA) is proposed in this study.

2) The Instance Hardness Threshold (IHT) [53] method is extended via a majority voting strategy to alleviate the adverse effects of imbalanced data in credit scoring.

3) The basic genetic algorithm [25] is extended through a promising initial population and self-adaptive mutation to optimize feature selection and classifier selection.

4) The proposed model is validated on three standard imbalanced credit scoring datasets, indicating its superior performance.

The remainder of the paper is organized as follows: Section 2 reviews related work. Section 3 elaborates on the details of the proposed model. Section 4 presents the experimental design. Section 5 describes the experimental results and provides a comparative analysis. Section 6 presents the conclusions and future work.

II. RELATED WORK

In this study, the proposed model mainly includes three parts: learning from imbalanced data, feature selection and classifier selection, and classifier ensemble. As important subfields of machine learning and credit scoring, these three aspects have drawn much attention from various scholars. In this section, prior studies on the aforementioned subfields are reviewed. The related works all make significant contributions in each subfield, but their limitations are identified to differentiate the proposed study.

A. LEARNING FROM IMBALANCED DATA

A dataset can be considered imbalanced when the number of positive samples is inconsistent with that of the negative samples. In credit scoring data, the number of positive samples is usually much lower than the number of negative samples. In such a situation, the classifiers tend to make false predictions regarding positive samples with fewer numbers [55]. Hence, credit scoring models have difficulty in making accurate predictions when imbalanced data are input directly. Given the importance of this issue, numerous sampling approaches have been proposed to address imbalanced data before training the models.

There are three categories of sampling approaches: oversampling, undersampling, and hybrid-sampling. The oversampling approach can be used to sample imbalanced data by generating new positive samples or replicating some positive samples, such as random oversampling and the synthetic minority oversampling technique (SMOTE) [11]. Almhaithawi *et al.* [3] applied four common classifiers with SMOTE for fraud detection and concluded that SMOTE could improve the performance of most classifiers. The undersampling approach can be used to sample imbalanced data by eliminating some negative samples or generating new negative samples to replace the original negative samples, such as BalanceCascade [39] and cluster centroids [38]. He *et al.* [23] extended the BalanceCascade approach to generate adjustable balanced subsets based on the imbalance ratios of training data and obtained a better predictive performance than using the original BalanceCascade approach. The hybrid-sampling approach can be used to sample imbalanced data by combining oversampling and undersampling approaches, such as SMOTE-Tomek links [59]. Sun *et al.* [54] proposed a hybrid-sampling approach named DTE-SBD, and proved its effectiveness for imbalanced enterprise credit evaluation.

However, the aforementioned approaches only handle the problem of inconsistent sample sizes in imbalanced data. Moreover, traditional oversampling methods tend to inject irrelevant data, which leads to model overfitting. Traditional undersampling methods tend to eliminate too much useful data, leading to information loss. Meanwhile, none of these studies analyzed why imbalanced data affect classifiers. He and Garcia [24] reviewed the advancements of research in imbalanced data and concluded that the class overlap was one of the main reasons that degraded the classifier performance in imbalanced learning. Smith *et al.* [53] identified and analyzed samples that were frequently misclassified by learning algorithms and found that class overlap was a principal contributor to misclassification. To resolve class overlap, Smith *et al.* [53] proposed an undersampling approach named Instance Hardness Threshold (IHT), which identified the sample points that were hard to classify, i.e., Instance Hardness Points (IHP), using classifiers, and eliminated these samples from the training data to alleviate the adverse effect of class overlap. Garcia *et al.* [21] also studied the IHT and proved its effectiveness.

Although the IHT method helps resolve class overlap, it heavily depends on the performance of a single classifier for identifying IHP [34]. Employing a poorly-performing base classifier for identifying IHP tends to eliminate a significant number of negative samples, leading to information loss. Even if a well-performing classifier is employed to identify IHP, some useless negative samples cannot be eliminated, and the sampled data obtained are not widely applicable to all classifiers in the model. Therefore, to further develop the IHT into an effective approach for credit scoring, in this study, it is extended by the majority voting strategy, which not only improves the data quality after sampling but also improves the applicability of the sampled data.

B. FEATURE SELECTION AND CLASSIFIER SELECTION

In credit scoring, redundant or irrelevant features in training data often lead to model overfitting, which affects model performance. Furthermore, poorly-performing or correlated base classifiers in an ensemble model may affect ensemble performance. These problems highlight the importance of effectively selecting optimal features and base classifiers. Feature selection and classifier selection in credit scoring can be considered as a NP-hard problem [47]. Thus, several optimization approaches have been proposed to resolve these problems, such as grid search and random search [5]. However, with the increase of data and model complexity in credit scoring, the search space of feature selection and classifier selection increases sharply, considerably increasing the cost of the aforementioned approaches.

To address the data and model complexity, Hutter *et al.* [30] proposed a sequential model-based algorithm configuration (SMAC) procedure for solving general optimization problems by training an RF in the search space. This allows the optimization of both numerical and categorical parameters on a set of instances with less overhead. However, despite its success, the SMAC method is restricted to problems with moderate dimensions [57]. Hence, finding the optimal feature subset and classifier subset in a high-dimensional search space becomes challenging.

To overcome the aforementioned limitations, many scholars have adopted the genetic algorithm (GA) [25] that is motivated by the Darwinian biological evolution theory and has been widely employed in the optimization field to reduce the overhead of the optimization process and find the optimal solution. Similar to the biological evolution process, GA mainly solves the problem through four strategies, population, selection, crossover, and mutation. It has been proven that, with the same time cost, GA generally outperforms grid search and random search [37]. Ali *et al.* [2] proposed an LDA-GA-SVM, which used GA to optimize the parameters of the support vector machine (SVM). Chen *et al.* [12] used the GA to improve the complexity and weights of a learning vector quantization model for optimal or near-optimal cost-sensitive bankruptcy prediction. Oreski and Oreski [44] proposed a hybrid GA with neural networks and used it to select the optimal feature subset in credit risk assessment. Zhang *et al.* [61] proposed a multi-population GA that enhanced the selection, crossover, and mutation steps through multi-population interaction, and applied this method to feature selection and classifier selection for credit scoring. The aforementioned studies have demonstrated the application and superiority of the GA in machine learning models. However, the population is randomly initialized in the above applications of GA, which requires more iterations to determine the direction of evolution for populations, and hence, increases the time cost. Therefore, in this study, a new HGA is proposed that generates a promising initial population for GA and enhances the mutation step through self-adaptation, to achieve better performance in terms of feature selection and classifier selection.

C. CLASSIFIER ENSEMBLE

The ensemble model has been proven to be an effective approach for improving the performance of the credit scoring model [56]. It is designed to enhance model performance by training multiple single base classifiers and integrating their decisions using an ensemble strategy. Currently, voting, bagging [6], boosting [51], and stacking [58] are the mainstream ensemble strategies for credit scoring. In particular, owing to its high robustness and excellent performance, the stacking approach has been extensively employed in credit scoring models [61]. Fedorova *et al.* [17] applied different stacking-based combinations of machine learning algorithms to construct a well-performing ensemble model for the bankruptcy prediction of Russian manufacturing companies. Wang *et al.* [56] compared three ensemble strategies in credit scoring and concluded that stacking could significantly improve model performance. Ali *et al.* [1] proposed a stacked support vector machine and demonstrated its superior performance over the other state-of-the-art machine learning ensemble models. The aforementioned works have demonstrated the superiority of the stacking strategy in constructing ensemble models. However, the effect of the stacking strategy also depends on the performance of the base classifiers [61]. Hence, this study uses a stacking strategy to construct a multi-stage ensemble model by integrating the decisions of the selected base classifiers from the candidate classifiers, which are trained by sampled data with selected features.

III. PROPOSED MODEL

In this study, a novel multi-stage ensemble model with an HGA is proposed. As shown in Figure 1, the proposed model consists of four stages: data sampling, feature selection, classifier selection, and classifier ensemble. In the data sampling stage, the traditional IHT method is extended through a majority voting strategy so that it can handle imbalanced training data. In the feature selection stage, the proposed HGA is used to select optimal feature subsets for each base classifier, and then, majority voting is employed to integrate all subsets into the aggregated optimal feature subset. In the classifier selection stage, the proposed HGA is used to select the optimal base classifier subset and further develop it into an ensemble model with a stacking strategy in the classifier ensemble stage. The details of the above stages and approaches are presented in the following subsections.

A. VOTING-BASED IHT (VIHT) METHOD

To handle imbalanced training data, the IHT method [53] is employed in this study to identify the sample points that are hard to classify, i.e., IHPs, using a classifier. For example, the four scatter plots at the bottom right corner of Figure 2 illustrate the IHPs that are identified by different base classifiers. The axes of the scatter plots demonstrate the values of the two dimension-reduced features through principal component analysis [41]. The pink, blue, and red points represent the IHP, negative, and positive samples, respectively. It can be seen that the IHPs identified using different base classifiers differ significantly in terms of quantity and distribution. Thus, the credit scoring data sampled through IHT using a certain classifier are not applicable to all base classifiers. Therefore, the IHT method is extended by combining multiple classifiers with a majority voting strategy to alleviate the aforementioned limitations in this study. The different IHPs, obtained through IHT using multiple base classifiers are integrated using the majority voting strategy into the aggregated IHP, which will be eliminated eventually, thus enhancing the applicability of sampled data to diverse base classifiers.

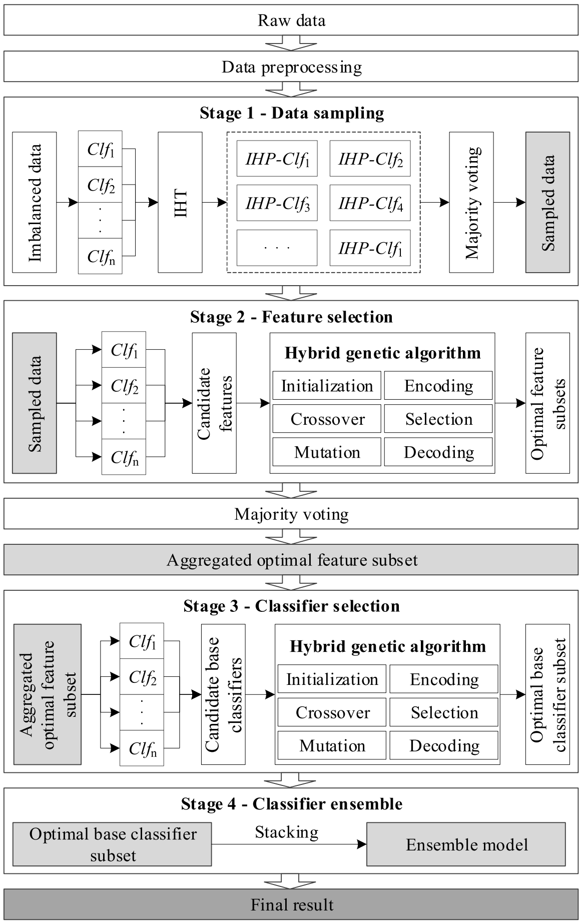


FIGURE 1. **Framework of the proposed model.**

The framework of the proposed voting-based IHT (VIHT) method is illustrated in Figure 2. The scatter plot at the top-left corner illustrates the raw training data that will be identified using the VIHT method. In the grids, −1, *1*, and *0* represent the negative sample points, positive sample points, and aggregated IHP, respectively. First, multiple base classifiers Clf_i $(i = 1{,}2\ldots n)$ are used to identify the different IHPs, i.e., $IHP\text{-}Clf_i$ $(i = 1{,}2\ldots n)$ through IHT. Then, record the number of times for each sample point to be identified as an IHP. Next, these different IHPs are integrated into aggregated IHPs using the majority voting strategy. For example, if a sample point is identified as an IHP by more than half of the base classifiers, it will be considered as an aggregated IHP. Finally, the sampled data are obtained by eliminating aggregated IHPs. Hence the output of the VIHT is a sampled data that are applicable to most classifiers.

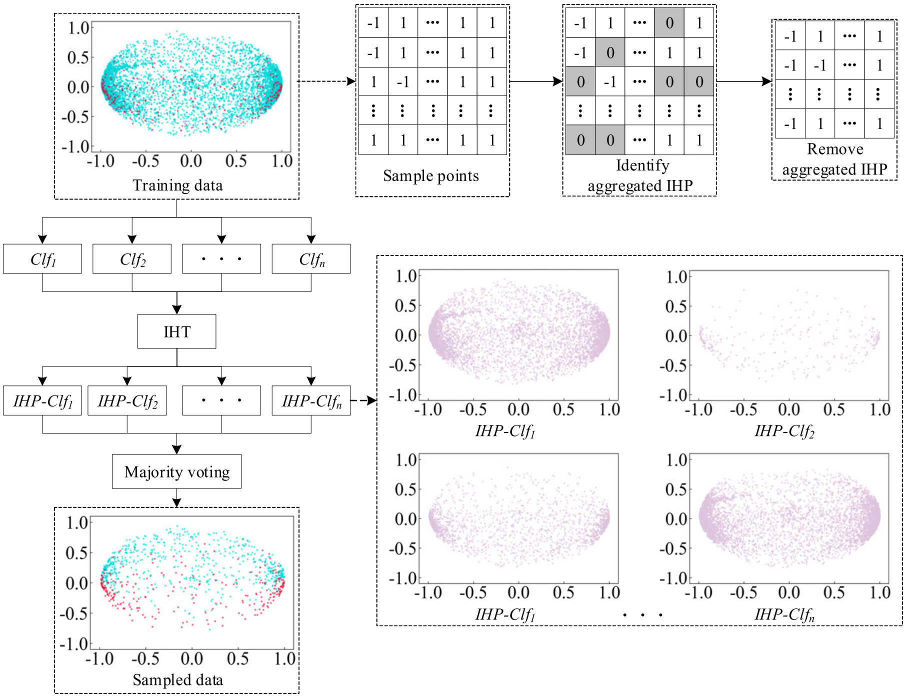


FIGURE 2. Framework of VIHT method.

B. HYBRID GENETIC ALGORITHM (HGA) METHOD

To optimize feature selection and classifier selection for ensemble modeling, a new HGA with a promising initial population and self-adaptive mutation is proposed in this study (Figure 3).

Step 1: Parameter initialization: The number of individuals in the initial population, crossover rate, initial mutation rate, maximum number of generations, and gene number of each individual are initialized.

Step 2: Encoding: A binary encoding scheme is employed to encode a candidate feature subset or candidate base classifier subset, where 0 indicates that the feature or classifier corresponding to the current gene is not selected and 1 represents that it is selected.

Step 3: Population initialization: The initial population plays an important role in the evolution of the GA toward a promising direction. Inspired by the SMAC procedure [30] for solving general optimization problems by training an RF in the search space, this study incorporates the procedure into the GA to generate a promising initial population in the HGA, as shown in Figure 4.

a) Generate Q individuals randomly, representing the candidate feature subset or candidate base classifier subset, and employ balance accuracy [8] to evaluate the practical classification performance (i.e., fitness) of each individual through 5-fold cross validation.

b) Train an RF using the individuals of the candidate feature or candidate base classifier subset as predictors and the practical classification performance of individuals as the target label.

c) Generate $Q1$ ($Q1 \gg Q$) new individuals randomly, and use the trained RF to predict the classification performance of $Q1$ generated individuals.

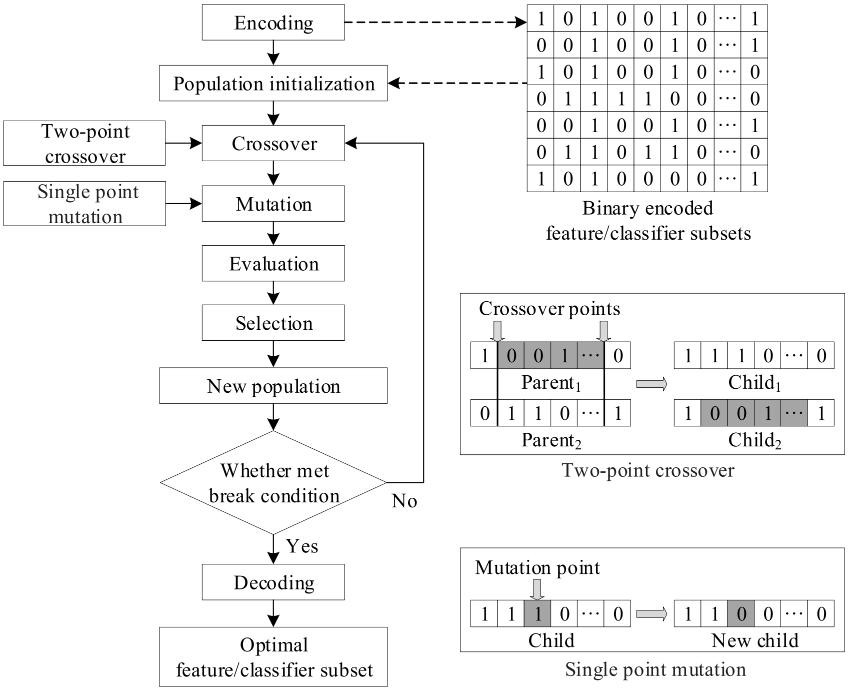
d) Select the Q individuals with top-ranked predicted classification performance through RF as the initial population of GA. Use the predicted classification performance, instead of practical classification performance, to select the initial population of GA will reduce the algorithmic overhead of GA greatly.

Step 4: Crossover: A two-point crossover approach is used to evolve the individuals in the parent population according to a certain crossover rate to obtain the offspring population.

Step 5: Mutation: A fixed mutation rate tends to lead GA to fall into the local optimum or makes it difficult for the GA to converge. Hence, a self-adaptive mutation is proposed to overcome the above problems in HGA. The single point mutation approach is used to randomly mutate a gene of individuals in the offspring population according to a certain mutation rate and then update the mutation rate using Equation (1):

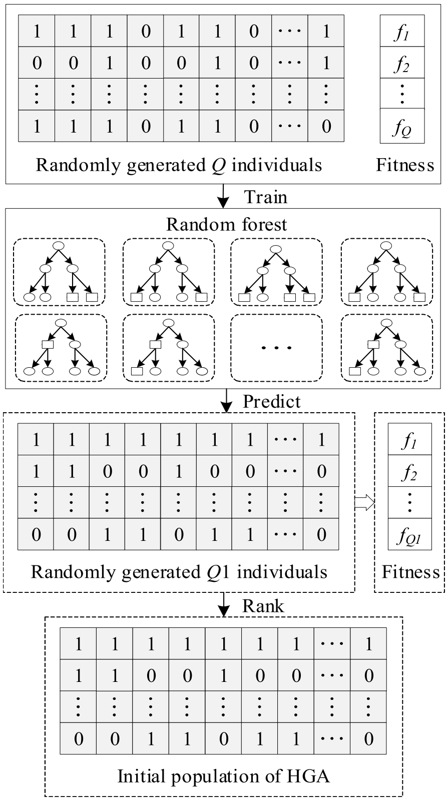
$$Mr_{g+1} = \begin{cases} Mr_g + \dfrac{(1-Mr_0)}{G}\left(1-e^{-g}\right) & g \le \dfrac{G}{2} \\ Mr_g - \dfrac{(1-Mr_0)}{G}\left(1-e^{-g}\right) & g > \dfrac{G}{2}, \end{cases} \quad (1)$$

where Mr_0 and Mr_g represent the initial mutation rate and the mutation rate in the gth generation, respectively, $g \in \{1, 2\ldots, G\}$, and G indicates the maximum number of generations. It can be seen from the equation, in the early stage of evolution, the mutation rate is increasing with the iteration to



1	0	1	0	0	1	0	⋯	1
0	0	1	0	0	1	0	⋯	1
1	0	1	0	0	1	0	⋯	0
0	1	1	1	1	0	0	⋯	0
0	0	1	0	0	1	0	⋯	1
0	1	1	0	1	1	0	⋯	0
1	0	1	0	0	0	0	⋯	1

FIGURE 3. Framework of the HGA.


Randomly generated Q individuals
Fitness
f_1
f_2
f_Q
Train
Random forest
Predict
Randomly generated $Q1$ individuals
Fitness
f_1
f_2
f_{Q1}
Rank
Initial population of HGA

1	1	1	0	1	1	0	⋯	1
0	0	1	0	0	1	0	⋯	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	0	1	1	0	⋯	0

1	1	1	1	1	1	1	⋯	1
1	1	0	0	1	0	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	1	1	0	1	1	⋯	0

1	1	1	1	1	1	1	⋯	1
1	1	0	0	1	0	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	1	1	0	1	1	⋯	0

FIGURE 4. Population initialization.

enhance the diversity of the population. In the late stage of evolution, the mutation rate is decreasing with the iterations to speed up the convergence.

Step 6: Evaluation: Calculate the practical classification performance (i.e., fitness) of each individual in both the parent and offspring populations through 5-fold cross validation.

Step 7: Selection: Select Q individuals with top-ranked practical classification performance in both the parent and offspring populations, and used as the new population for further evolution.

Step 8: Let $g = g+1$. Repeat steps 4-7 until the termination condition is satisfied, and the optimal individual is decoded to obtain the optimal feature/classifier subset.

C. FEATURE SELECTION

The proposed HGA is used to select the optimal feature subset for each base classifier to eliminate redundant or irrelevant features and increase the applicability of the selected features to various base classifiers. Because the different classifiers have different optimal feature subsets, to enhance the applicability of the selected feature subset, the majority voting method in VIHT is also employed for feature selection to output the aggregated optimal feature subset. For example, if a feature is selected into the optimal feature subsets by more than half of the classifiers, this feature will be added to the aggregated optimal feature subset, otherwise it will not be added. In the feature selection procedure, an individual in the HGA represents a candidate feature subset, a population in the generation consists of multiple individuals, and the optimal individual represents the optimal feature subset that is obtained through genetic evolution. Furthermore, the practical classification performance (i.e., fitness) of each candidate feature subset for each base classifier is evaluated using the balance accuracy through 5-fold cross validation.

D. CLASSIFIER SELECTION

Considering that the poorly-performing or correlated base classifiers in an ensemble model may affect the ensemble performance, the proposed HGA is used to select the optimal base classifier subset, which is further integrated into an effective ensemble model through a stacking strategy. In the classifier selection procedure, an individual in the HGA represents a candidate base classifier subset, a population in the generation consists of multiple individuals, and the optimal individual represents the optimal base classifier subset that is obtained through genetic evolution. The practical classification performance (i.e., fitness) of each candidate base classifier subset corresponding to a candidate ensemble model is evaluated using balance accuracy through 5-fold cross validation.

E. CLASSIFIER ENSEMBLE

Although the optimal base classifier subset is obtained, the stacking strategy is employed to integrate the trained selected base classifier subset. The stacking strategy consists of two stages. First, the base classifiers are trained using training data through cross validation, and the predictions of the base classifiers are combined into a new feature matrix. Second, the obtained new feature matrix is used to train a meta-classifier to output the final decision. Due to the superiority of kernel ridge classifier [46], it is employed as a meta-classifier to integrate the decisions of multiple base classifiers in this study.

IV. EXPERIMENTAL DESIGN

In this study, 10 types of popular classifiers were used as candidate base classifiers for the proposed model, namely logistic regression (LR) [16], XGBoost, GBDT, RF, decision tree (DT) [50], LDA, bagging [6], extra trees (ET) [22], light gradient boosting machine (LightGBM) [33], and support vector machine (SVM) [48].

A. DATASET PREPROCESSING AND PARAMETER SETTING

In this study, three standard credit scoring datasets from the UC Irvine (UCI) machine learning repository, i.e., the German [4], Polish 1, and Polish 2 [62] datasets, were used to test the effectiveness of the proposed model. The details of these datasets are presented in Table 1.

As shown in Table 1, the German dataset contains 1000 samples, 300 of which are positive and 700 are negative. The dimension of the features, including the target label, is 21, with seven numerical features and 14 nominal features. The target label is a binary class, consisting of 0 and 1. The Polish1 dataset contains 7027 samples, 271 of which are positive and 6756 are negative. The dimension of the features, including the class label, is 65, with 64 numerical features and one nominal feature. The target label is a binary class, consisting of 0 and 1. The Polish2 dataset contains 10173 samples, 400 of which are positive and 9733 are negative. The dimension of the features, including the class label, is 65, with 64 numerical features and one nominal feature. The target label is a binary class, consisting of 0 and 1.

TABLE 1. Description of datasets.

Dataset	Total samples	Positive samples	Negative samples	Dimension of features (numerical/nominal)
German	1000	300	700	21(7/14)
Polish1	7027	271	6756	65(64/1)
Polish2	10173	400	9733	65(64/1)

Three basic data preprocessing approaches, namely standardization, normalization, and one-hot encoding [43], were used to pre-process the datasets before training the models. The numerical features were standardized and normalized by removing the mean and scaling to unit variance, to handle different orders of magnitude on different numerical features. The nominal features were handled via one-hot encoding to leverage the meaningful distance relationships between different nominal feature values. After data preprocessing, the dimension of features including the target label in the German dataset, is 58, with five numerical features and 53 nominal features, the range of numerical feature is $[-1,1]$, and the value of the nominal feature belongs to $\{0,1\}$. After data preprocessing, the dimension of features including the target label in Polish1 and Polish2 data set, are both 65, with 64 numerical features and one nominal feature, and the range of numerical feature is $[-1,1]$, and the value of the nominal feature belong to $\{0,1\}$.

To ensure effectiveness and comparability of the experiment, the optimal parameters were preset through a trial-run for HGA. In the HGA, the maximum number of generations G was set to 100, the number of individual Q in each population was set to 50, and the randomly generated individual $Q1$ was set to 200. A greater number of generations G will lead to more computation time but will also result in better performance. The crossover rate and initial mutation rate were set to 0.8 and 0.2, respectively. A higher crossover and mutation rate will lead to algorithmic convergence difficulty, but also produce a larger search space.

B. EVALUATION INDICATORS

In this study, four comprehensive evaluation indicators were employed to evaluate the performance of the proposed model, namely, balance accuracy (BACC) [8], F-score [36], G-mean [35], and Recall [9]. These comprehensive evaluation indicators are all determined by true positive (TP), false positive (FP), true negative (TN), and false negative (FN), and the higher comprehensive indicator value represents the better performance of evaluated models. These comprehensive evaluation indicators are widely employed in imbalanced learning [24], and their superiorities are detailed as follows.

BACC can better reflect the performance of classification models than accuracy during imbalanced learning [55]. Hence, in this study, BACC was adopted to evaluate model performance. Its calculation rule is shown in Equation (2), and the true positive rate (TPR) and true negative rate (TNR) are defined in Equations (3) and (4), respectively.

$$\text{BACC} = \frac{\text{TPR} + \text{TNR}}{2} \tag{2}$$

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3}$$

$$\text{Sensitivity} = \text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

Recall evaluates the ability of models to identify positive samples, which is critical for the credit scoring model. Its calculation rule is shown in Equation (4).

G-mean is another comprehensive evaluation indicator for the imbalanced learning model. The G-Mean shows whether the balance between classes is reasonable. The calculation rule of G-mean is shown in Equation (5), where Sensitivity and Specificity are defined in Equations (4) and (6), respectively.

$$\text{G-mean} = \sqrt{\text{Sensitivity} * \text{Specificity}} \tag{5}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{6}$$

F-score is the harmonic average of Precision and Recall, and reflects the tradeoffs between precision and recall. The calculation rule of F-score is shown in Equation (7), where Precision is defined in Equation (8).

$$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

C. EXPERIMENTAL SETTING

The raw dataset was divided as follows: Twenty percent of the total data were used as the test data, and the remaining 80% were further divided into 80% for training and 20% for validation. The data preprocessing approaches (e.g., standardization, normalization, and one-hot encoding) were imported from the Python module "sklearn". In the proposed VIHT method, the basic IHT method was imported from the Python module "imblearn". In the ensemble stage, the stacking approach was imported from the Python module "mlxtend". The classifiers LDA, RF, GBDT, SVC, DT, LR, ET, Bagging, and ridge regression were imported from Python module "sklearn". The classifiers Xgboost and LightGBM were imported from the Python modules "xgboost" and "ligtgbm", respectively. For a fair comparison, default parameters were adopted in all imported modules.

V. EXPERIMENTAL ANALYSIS

To enhance the diversity of the base classifiers, three base classifiers were reproduced from each type of base classifier with different appropriate parameters through trial and run. Four comprehensive evaluation indicators were used to evaluate the model performance, namely BACC, F-score, G-mean, and Recall. To enhance the robustness of the experiments and avoid single-results bias, each experiment was performed 10 times, and the average values were used as evaluation results. All the experiments were performed using Python version 3.7.5 on a PC with a 3.8-GHz Intel Core I7-10700 K, 32 GB RAM, and a Windows 10 operating system.

A. BASELINE RESULTS

To verify the performance of the proposed model, all base classifiers were tested on three datasets, and the results were indicated as the baseline results. As shown in Table 2, 10 types of base classifiers were tested on the German, Polish 1, and Polish 2 datasets.

B. PERFORMANCE EVALUATION OF THE VIHT METHOD

To prove the effectiveness of the proposed VIHT method on real datasets, its performance was evaluated on three datasets as shown in Table 3. The bolded values of evaluation indicators represent better performance of base classifiers after VIHT was applied than the baseline results. It can be seen from the Table 3, by comparing with the baseline results, all the base classifiers are significantly enhanced by the VIHT method in all or most evaluation indicators on all datasets.

To further verify the effectiveness of the VIHT method, two traditional sampling approaches, namely, IHT and SMOTE, were tested under the same conditions for comparison, with the results outlined in Table 4. The bolded values indicate better performance after IHT or SMOTE were applied than after VIHT was applied. It can be seen from the Table 4, most base classifiers perform worse after IHT or SMOTE are applied than after VIHT is applied in most evaluation indicators on all datasets. The outperformance of VIHT is owing to the following reasons:

1) VIHT eliminates the IHPs effectively to alleviate the class overlap problem.

2) VIHT extends the traditional sampling approaches by integrating the decisions of various classifiers to improve the applicability of sampled data to diverse base classifiers.

3) VIHT only eliminates samples considered hard to learn by multiple classifiers, instead of adding additional data in sampled data or eliminating too much information of data. Hence it can alleviate the model overfitting.

C. PERFORMANCE EVALUATION OF HGA-BASED FEATURE SELECTION METHOD

To prove the effectiveness of the proposed HGA-based feature selection method, its performance was evaluated on three sampled datasets.

The feature correlations before and after the HGA-based feature selection method were performed on three sampled datasets were shown through heatmaps in Figure 5 respectively. The axis represents the index of the features; the bluer region in the heatmap represents higher correlations between the corresponding feature pairs, and the redder region represents the contrary.

It can be seen from the figures, the feature correlations between different features on any sampled datasets are significantly reduced after HGA-based feature selection is

TABLE 2. Baseline results.

Dataset	Base classifier	BACC	F-score	G-mean	Recall
German	SVM	0.6357	0.4587	0.5714	0.3600
	LDA	0.6706	0.5341	0.6462	0.4933
	LightGBM	0.6526	0.5020	0.6188	0.4517
	DT	0.6113	0.4578	0.5930	0.4683
	RF	0.6526	0.5032	0.6207	0.4567
	LR	0.6631	0.5217	0.6361	0.4783
	XGBoost	0.6443	0.4917	0.6131	0.4500
	ET	0.6412	0.4762	0.5914	0.3967
	Bagging	0.6165	0.4349	0.5607	0.3617
	GBDT	0.6446	0.4888	0.6067	0.4300
Polish 1	SVM	0.5000	0.0000	0.0000	0.0000
	LDA	0.5086	0.0391	0.1374	0.0222
	LightGBM	0.5077	0.0349	0.1333	0.0192
	DT	0.5305	0.1041	0.3229	0.1131
	RF	0.5083	0.0379	0.1416	0.0212
	LR	0.5000	0.0000	0.0000	0.0000
	XGBoost	0.5086	0.0384	0.1407	0.0212
	ET	0.5010	0.0110	0.0602	0.0061
	Bagging	0.5039	0.0217	0.0959	0.0121
	GBDT	0.5092	0.0401	0.1416	0.0222
Polish 2	SVM	0.5000	0.0000	0.0000	0.0000
	LDA	0.5319	0.1179	0.2622	0.0709
	LightGBM	0.5546	0.1872	0.3371	0.1155
	DT	0.5753	0.1896	0.4366	0.2029
	RF	0.5584	0.1974	0.3466	0.1233
	LR	0.5028	0.0115	0.0533	0.0058
	XGBoost	0.5500	0.1728	0.3241	0.1068
	ET	0.5023	0.0174	0.0752	0.0097
	Bagging	0.5398	0.1430	0.2898	0.0864
	GBDT	0.5419	0.1510	0.2956	0.0883

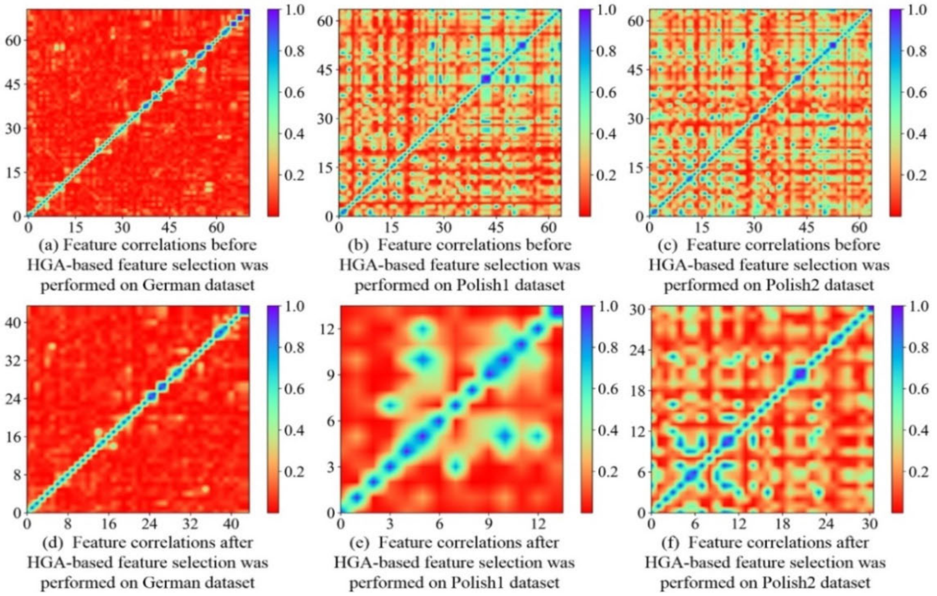
TABLE 3. Performance evaluation of the VIHT method.

Dataset	Base classifier	BACC	F-score	G-mean	Recall
German	SVM	**0.6761**	**0.5600**	**0.6697**	**0.7550**
	LDA	**0.6808**	**0.5665**	**0.6729**	**0.7817**
	LightGBM	**0.6614**	**0.5467**	**0.6519**	**0.7650**
	DT	**0.6244**	**0.5093**	**0.6178**	**0.7067**
	RF	**0.6705**	**0.5564**	**0.6592**	**0.7867**
	LR	**0.6883**	**0.5745**	**0.6774**	**0.8067**
	XGBoost	**0.6606**	**0.5462**	**0.6501**	**0.7683**
	ET	**0.6758**	**0.5610**	**0.6666**	**0.7767**
	Bagging	**0.6626**	**0.5448**	**0.6583**	**0.7217**
	GBDT	**0.6714**	**0.5570**	**0.6615**	**0.7800**
Polish 1	SVM	**0.6488**	**0.1284**	**0.6015**	**0.8909**
	LDA	**0.6581**	**0.1330**	**0.6213**	**0.8737**
	LightGBM	**0.6453**	**0.1263**	**0.5897**	**0.9071**
	DT	**0.6388**	**0.1259**	**0.5965**	**0.8667**
	RF	**0.6461**	**0.1263**	**0.5892**	**0.9111**
	LR	**0.6644**	**0.1355**	**0.6299**	**0.8747**
	XGBoost	**0.6461**	**0.1266**	**0.5917**	**0.9051**
	ET	**0.6456**	**0.1285**	**0.6067**	**0.8657**
	Bagging	**0.6545**	**0.1312**	**0.6147**	**0.8788**
	GBDT	**0.6481**	**0.1280**	**0.6000**	**0.8929**
Polish 2	SVM	**0.6805**	**0.1541**	**0.6397**	**0.9117**
	LDA	**0.6855**	**0.1609**	**0.6620**	**0.8631**
	LightGBM	**0.6754**	0.1506	**0.6261**	**0.9282**
	DT	**0.6542**	0.1453	**0.6174**	**0.8699**
	RF	**0.6709**	0.1491	**0.6220**	**0.9223**
	LR	**0.6781**	**0.1580**	**0.6550**	**0.8534**
	XGBoost	**0.6740**	0.1507	**0.6276**	**0.9194**
	ET	**0.6691**	**0.1508**	**0.6326**	**0.8864**
	Bagging	**0.6704**	**0.1527**	**0.6396**	**0.8709**
	GBDT	**0.6719**	0.1505	**0.6286**	**0.9087**

performed. It indicates that the HGA-based feature selection method effectively reduces the relevant or redundant features. Its evaluation performance on three sampled datasets is shown in Table 5, where the bolded values of evaluation indicators represent that the base classifiers performed better after HGA-based feature selection was applied. It can be seen from

TABLE 4. Performance evaluation of IHT and SMOTE methods.

Dataset	Base classifier	IHT				SMOTE			
		BACC	F-score	G-mean	Recall	BACC	F-score	G-mean	Recall
German	SVM	0.6664	0.5329	0.6518	0.5300	**0.6876**	**0.5649**	**0.6810**	0.5967
	LDA	**0.6907**	**0.5718**	**0.6901**	0.6700	**0.6911**	**0.5710**	**0.6878**	0.6300
	LightGBM	**0.6619**	0.5202	0.6346	0.4767	0.6565	0.5258	0.6507	0.5817
	DT	0.6051	0.4509	0.5877	0.4717	0.5875	0.4449	0.5809	0.5150
	RF	0.6585	0.5173	0.6352	0.4883	0.6673	0.5399	**0.6625**	0.5967
	LR	**0.6951**	**0.5768**	**0.6941**	0.6617	0.6876	0.5659	**0.6839**	0.6217
	XGBoost	0.6562	0.5133	0.6314	0.4817	0.6486	0.5164	0.6426	0.5700
	ET	0.6551	0.5040	0.6166	0.4367	0.6723	0.5456	0.6648	0.5767
	Bagging	0.6270	0.4676	0.5951	0.4333	0.6424	0.5004	0.6274	0.5133
	GBDT	0.6711	0.5393	0.6563	0.5350	0.6774	0.5529	**0.6732**	0.6083
Polish 1	SVM	0.5903	0.1094	0.4972	**0.9071**	0.5570	**0.1648**	0.3719	0.1434
	LDA	0.6141	0.1167	0.5475	**0.8909**	0.5671	**0.1613**	0.4258	0.1939
	LightGBM	0.6161	0.1171	0.5488	0.8949	0.5619	**0.1733**	0.3871	0.1556
	DT	0.5926	0.1124	0.5506	0.8081	0.5493	0.1232	0.4217	0.2000
	RF	0.6200	0.1186	0.5576	0.8889	0.5666	**0.1792**	0.4016	0.1687
	LR	0.6343	0.1235	0.5816	**0.8848**	0.5622	**0.1578**	0.4080	0.1768
	XGBoost	0.6219	0.1192	0.5611	0.8889	0.5600	**0.1730**	0.3786	0.1485
	ET	0.6079	0.1155	0.5516	0.8616	0.5142	0.0614	0.1836	0.0394
	Bagging	0.6111	0.1179	0.5724	0.8232	0.5291	0.1034	0.2907	0.0899
	GBDT	0.6182	0.1178	0.5533	0.8929	0.5635	**0.1670**	0.4018	0.1697
Polish 2	SVM	0.5955	0.1224	0.5023	**0.9136**	0.6154	**0.2689**	0.5101	0.2728
	LDA	0.5983	0.1239	0.5195	**0.8932**	0.6545	**0.2896**	0.5922	0.3777
	LightGBM	0.6013	0.1247	0.5242	0.8951	0.6283	**0.3121**	0.5272	0.2874
	DT	0.5830	0.1204	0.5231	0.8388	0.6006	**0.1949**	0.5173	0.2990
	RF	0.6080	0.1268	0.5317	0.9019	0.6310	**0.3100**	0.5344	0.2961
	LR	0.6215	0.1313	0.5536	**0.9019**	0.6322	**0.2761**	0.5491	0.3204
	XGBoost	0.6082	0.1269	0.5352	0.8961	0.6172	**0.2935**	0.5055	0.2641
	ET	0.5985	0.1247	0.5355	0.8641	0.5333	0.1204	0.2841	0.0835
	Bagging	0.5921	0.1230	0.5316	0.8515	0.5914	**0.2397**	0.4536	0.2146
	GBDT	0.5926	0.1223	0.5153	0.8845	0.6338	**0.3037**	0.5429	0.3078



(a) Feature correlations before HGA-based feature selection was performed on German dataset

(b) Feature correlations before HGA-based feature selection was performed on Polish1 dataset

(c) Feature correlations before HGA-based feature selection was performed on Polish2 dataset

(d) Feature correlations after HGA-based feature selection was performed on German dataset

(e) Feature correlations after HGA-based feature selection was performed on Polish1 dataset

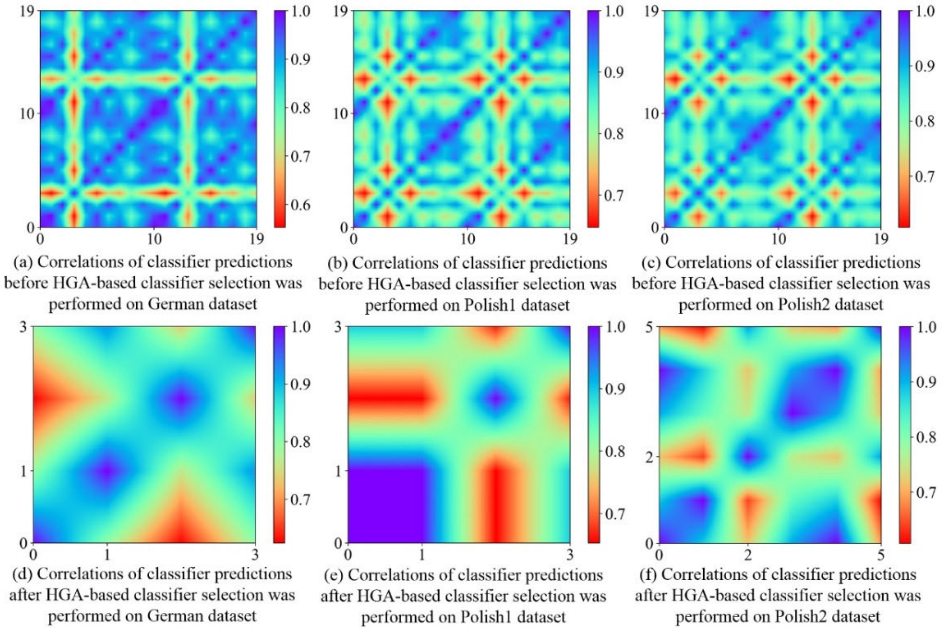
(f) Feature correlations after HGA-based feature selection was performed on Polish2 dataset

FIGURE 5. Heatmaps of the feature correlations before and after the HGA-based feature selection method was performed on three sampled datasets.

the Table 5, all base classifiers are significantly enhanced by the HGA-based feature selection method in all or most evaluation indicators on all sampled datasets, because the HGA-based feature selection method effectively eliminates the irrelevant and redundant features.

D. PERFORMANCE EVALUATION OF THE CLASSIFIER SELECTION AND ENSEMBLE

To prove the effectiveness of the proposed HGA-based classifier selection method, its performance was evaluated on three sampled datasets after feature selection.



(a) Correlations of classifier predictions before HGA-based classifier selection was performed on German dataset

(b) Correlations of classifier predictions before HGA-based classifier selection was performed on Polish1 dataset

(c) Correlations of classifier predictions before HGA-based classifier selection was performed on Polish2 dataset

(d) Correlations of classifier predictions after HGA-based classifier selection was performed on German dataset

(e) Correlations of classifier predictions after HGA-based classifier selection was performed on Polish1 dataset

(f) Correlations of classifier predictions after HGA-based classifier selection was performed on Polish2 dataset

FIGURE 6. Heatmaps of correlations of classifier predictions before and after the HGA-based classifier selection method was performed on three sampled datasets after feature selection.

TABLE 5. Performance evaluation of the feature selection by HGA approach.

Dataset	Base classifier	BACC	F-score	G-mean	Recall
German	SVM	**0.6761**	**0.5600**	**0.6697**	**0.7550**
	LDA	**0.6808**	**0.5665**	**0.6729**	**0.7817**
	LightGBM	**0.6614**	**0.5467**	**0.6519**	**0.7650**
	DT	**0.6244**	**0.5093**	**0.6178**	**0.7067**
	RF	**0.6705**	**0.5564**	**0.6592**	**0.7867**
	LR	**0.6883**	**0.5745**	**0.6774**	**0.8067**
	XGBoost	**0.6606**	**0.5462**	**0.6501**	**0.7683**
	ET	**0.6758**	**0.5610**	**0.6666**	**0.7767**
	Bagging	**0.6626**	**0.5448**	**0.6583**	**0.7217**
	GBDT	**0.6714**	**0.5570**	**0.6615**	**0.7800**
Polish 1	SVM	**0.6488**	**0.1284**	**0.6015**	**0.8909**
	LDA	**0.6581**	**0.1330**	**0.6213**	**0.8737**
	LightGBM	**0.6453**	**0.1263**	**0.5897**	**0.9071**
	DT	**0.6388**	**0.1259**	**0.5965**	**0.8667**
	RF	**0.6461**	**0.1263**	**0.5892**	**0.9111**
	LR	**0.6644**	**0.1355**	**0.6299**	**0.8747**
	XGBoost	**0.6461**	**0.1266**	**0.5917**	**0.9051**
	ET	**0.6456**	**0.1285**	**0.6067**	**0.8657**
	Bagging	**0.6545**	**0.1312**	**0.6147**	**0.8788**
	GBDT	**0.6481**	**0.1280**	**0.6000**	**0.8929**
Polish 2	SVM	**0.6805**	**0.1541**	**0.6397**	**0.9117**
	LDA	**0.6855**	**0.1609**	**0.6620**	**0.8631**
	LightGBM	**0.6754**	0.1506	**0.6261**	**0.9282**
	DT	**0.6542**	0.1453	**0.6174**	**0.8699**
	RF	**0.6709**	0.1491	**0.6220**	**0.9223**
	LR	**0.6781**	**0.1580**	**0.6550**	**0.8534**
	XGBoost	**0.6740**	0.1507	**0.6276**	**0.9194**
	ET	**0.6691**	**0.1508**	**0.6326**	**0.8864**
	Bagging	**0.6704**	**0.1527**	**0.6396**	**0.8709**
	GBDT	**0.6719**	0.1505	**0.6286**	**0.9087**

The correlations between the classifier predictions before and after the HGA-based classifier selection method was performed on three sampled datasets after feature selection were shown through heatmaps in Figure 6 respectively. The axis represents the classifier index; and the bluer region in the heatmap represents higher prediction correlations between the corresponding classifier pairs, and the redder region represents the contrary. It can be seen form the figures, the prediction correlations between different classifiers on any sampled datasets are significantly reduced after HGA-based classifier selection is performed. It indicates that the HGA-based classifier selection method effectively reduces the correlated classifiers with high prediction correlations. Further, the selected optimal base classifiers were incorporated into the proposed ensemble model through stacking strategy. The proposed ensemble model was then compared to the general ensemble model that was combined from all base classifiers without classifier selection (abbreviated as ‘‘The general ensemble model’’). Its evaluation performance on three sampled datasets after feature selection is shown in Table 6, where the bolded values of evaluation indicators represent that the proposed ensemble model performed better than the general ensemble model without classifier selection. It can be seen from the Table 6, all ensemble models are significantly enhanced by the HGA-based classifier selection method in all evaluation indicators on all sampled datasets, because the HGA-based classifier selection method effectively eliminates the poor-performed and correlated classifiers.

E. STATISTIC RESULTS

To demonstrate the reliability of the experimental results, a statistical test should be performed. The well-known analysis of variance (ANOVA) and its non-parametric counterpart, the Friedman test can be used to test the effectiveness of all models under different methods. Friedman [20] experimentally compared ANOVA with his proposed Friedman test on 56 independent problems and showed that the two methods

TABLE 6. Performance evaluation of the classifier selection and ensemble.

Dataset	Base classifier	BACC	F-score	G-mean	Recall
German	The general ensemble model	0.6770	0.5620	0.6691	0.7733
	The proposed ensemble model	**0.7137**	**0.5999**	**0.7031**	**0.8267**
Polish 1	The general ensemble model	0.6479	0.1294	0.6077	0.8677
	The proposed ensemble model	**0.6671**	**0.1366**	**0.6311**	**0.8798**
Polish 2	The general ensemble model	0.6770	0.1514	0.6289	0.9272
	The proposed ensemble model	**0.6924**	**0.1576**	**0.6471**	**0.9379**

TABLE 7. Performance evaluation of the classifier selection and ensemble.

Dataset	Method	BACC	F-score	G-mean	Recall
German	Baseline	7	7	7	7
	IHT	5	5	4	6
	SMOTE	6	6	6	5
	VIHT	4	4	5	3
	Feature selection	3	3	3	4
	The general ensemble model	2	2	2	2
	The proposed ensemble model	1	1	1	1
Polish 1	Baseline	7	7	7	7
	IHT	6	2	6	3
	SMOTE	5	6	5	6
	VIHT	3	4	3	1
	Feature selection	3	4	3	1
	The general ensemble model	3	3	2	5
	The proposed ensemble model	1	1	1	4
Polish 2	Baseline	7	6	7	7
	IHT	5	1	6	5
	SMOTE	6	7	5	6
	VIHT	3	4	2	3
	Feature selection	3	4	2	3
	The general ensemble model	2	3	4	2
	The proposed ensemble model	1	2	1	1

mostly agree. In recent years, Demšar [15] presented that due to the possible violations of the tests' assumptions by a typical machine learning data, non-parametric tests (Friedman test) should be preferred over parametric tests (ANOVA). Hence, the Friedman test was adopted in this study. When the null-hypothesis was rejected, the Nemenyi test [42] was applied. In the Friedman test, 52 classification models were ranked based on different evaluation indicators. These models included 10 types of base classifiers before the VIHT method was applied, 10 types of base classifiers after the IHT method was applied, 10 types of base classifiers after the SMOTE method was applied, 10 types of base classifiers after the VIHT method was applied, 10 types of base classifiers after the HGA-based feature selection method was applied, the general ensemble model without classifier selection, and the proposed ensemble model. Then the score of each method was calculated by averaging the ranking of the models that employ this method. Finally, the statistical significance of the Friedman test can be obtained from the scores of all the methods. Table 7 shows the scores of each method on different datasets and evaluation indicators, and the results of the Friedman test. It can be seen from the table, all of proposed methods obtain higher scores than the comparison methods in all or most evaluation indicators on all datasets. It can be seen from Table 7 that the statistics value of the Friedman test on most evaluation indicators is larger than the critical value (i.e., 2.996). The null-hypothesis, i.e., all of the methods having the same performance, is rejected, according to Demšar [15]. Subsequently, the Nemenyi test was used to perform the post hoc test, the results of which are shown in Figure 7, where the critical distance (CD) indicates the mean ranking score difference. The higher the position of the classifier on the coordinate axis to the left, the better is the performance of the classifier, and vice versa. It can be seen from the figure that the proposed methods perform better than the corresponding comparison methods on most evaluation indicators, and the proposed ensemble model always has the best performance.

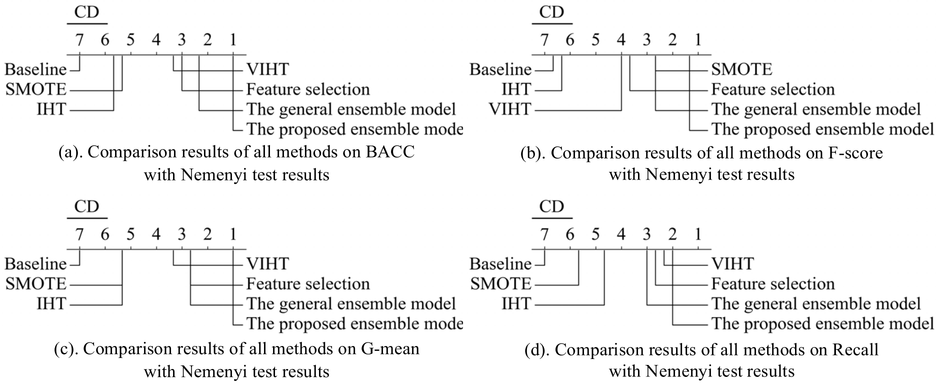
F. PERFORMANCE COMPARISON BETWEEN THE PROPOSED ENSEMBLE MODEL AND THE BENCHMARK ENSEMBLE MODELS

The performance comparison between the proposed ensemble model and benchmark ensemble models proposed by Seiffert *et al.* [52], Sun *et al.* [54], and Liu *et al.* [40] is presented in Table 8, where the bolded values represent the best performance in the comparison. The source codes for the ensemble models by Seiffert *et al.* [52] and Liu *et al.* [40] are public; thus, for fair comparison, they were tested with the same experimental settings as the proposed ensemble model, including the datasets, running times, and preprocessing approaches. The source codes for the ensemble model proposed by Sun *et al.* [54] are not public; hence, this model was rigorously reproduced using the provided modeling scheme and methodology, and then, tested under the same experimental settings as the proposed ensemble model in this study.

To observe the performance of each ensemble model more intuitively, all the results are presented in the form of histograms in Figure 8, where the indices represent the dataset and evaluation indicators, respectively. For example, the histogram corresponding to the indices German and BACC represents the performance of the four comparison models using the BACC indicator on the German dataset. It can be seen from both Table 8 and Figure 8, the proposed ensemble model achieves the best performance in all or most evaluation indicators on all datasets.

G. COMPLEXITY AND APPLICATION ANALYSIS

To verify the practicality of the proposed model, a German dataset was used to assess the computational complexity of the various methods employed in the proposed model. Furthermore, the time of model prediction was tested. All methods were implemented in Python version 3.7.5 on a PC with a 3.8-GHz Intel Core I7-10700 K, 32 GB RAM, and


CD
7 6 5 4 3 2 1
Baseline
SMOTE
IHT
VIHT
Feature selection
The general ensemble model
The proposed ensemble mode
CD
7 6 5 4 3 2 1
Baseline
IHT
VIHT
SMOTE
Feature selection
The general ensemble model
The proposed ensemble model
CD
7 6 5 4 3 2 1
Baseline
SMOTE
IHT
VIHT
Feature selection
The general ensemble model
The proposed ensemble mode
CD
7 6 5 4 3 2 1
Baseline
SMOTE
IHT
VIHT
Feature selection
The general ensemble model
The proposed ensemble model

(a). Comparison results of all methods on BACC with Nemenyi test results

(b). Comparison results of all methods on F-score with Nemenyi test results

(c). Comparison results of all methods on G-mean with Nemenyi test results

(d). Comparison results of all methods on Recall with Nemenyi test results

FIGURE 7. Nemenyi test results.

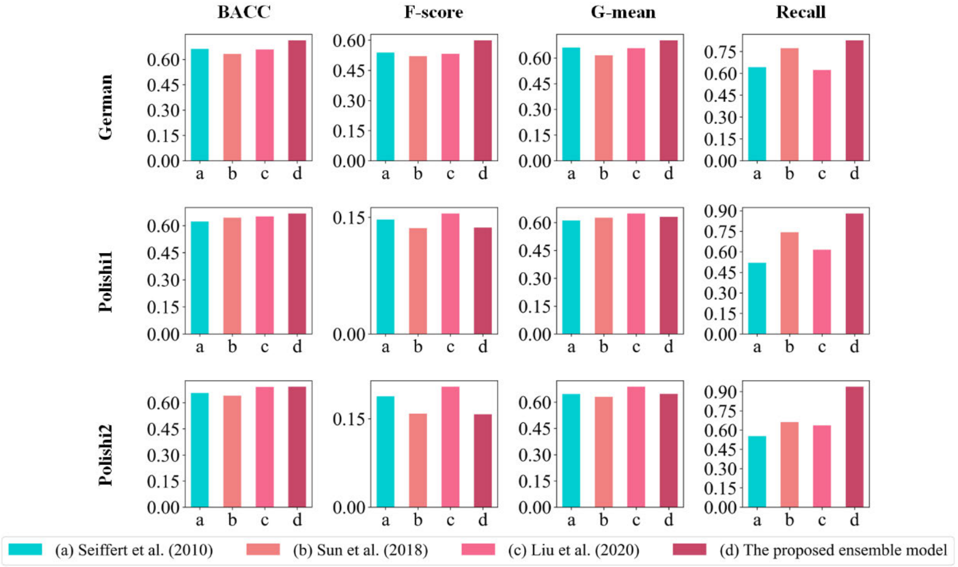


FIGURE 8. Histograms of performance comparison between the proposed ensemble model and benchmark ensemble models.

TABLE 8. Results of performance comparison between the proposed ensemble model and the benchmark ensemble models.

Dataset	Method	BACC	F-score	G-mean	Recall
German	Seiffert et al. (2010)	0.6619	0.5387	0.6607	0.6417
	Sun et al. (2018)	0.6323	0.5213	0.6150	0.7717
	Liu et al. (2020)	0.6587	0.5331	0.6568	0.6217
	The proposed model	**0.7137**	**0.5999**	**0.7031**	**0.8267**
Polish 1	Seiffert et al. (2010)	0.6224	0.1470	0.6112	0.5192
	Sun et al. (2018)	0.6434	0.1360	0.6259	0.7434
	Liu et al. (2020)	0.6507	**0.1548**	**0.6491**	0.6141
	The proposed model	**0.6671**	0.1366	0.6311	**0.8798**
Polish 2	Seiffert et al. (2010)	0.6562	0.1879	0.6469	0.5534
	Sun et al. (2018)	0.6405	0.1583	0.6298	0.6612
	Liu et al. (2020)	0.6909	**0.2043**	**0.6885**	0.6368
	The proposed model	**0.6924**	0.1576	0.6471	**0.9379**

TABLE 9. Running times of different methods and model prediction on German datasets.

Method		Time (s)
Imbalanced data processing	VIHT	1.639
Feature selection	HGA-based feature selection	161.046
Classifier selection	HGA-based classifier selection	846.887
Model prediction	The proposed model	0.324

Windows 10 operating system. The running times of each model and method are listed in Table 9.

As observed in Table 9, during imbalanced data processing, VIHT only requires 1.639 s, whereas the HGA-based feature selection and classifier selection requires 161.046 s to select optimal feature subsets and 846.887 s to select optimal classifier subsets respectively with 1000 training samples. Furthermore, the running time of the proposed model in predicting 1000 test samples was only 0.324 s. Notably, the training process of the proposed model can be performed offline. In addition, the trained model occupies less than 1M of memory, which means that the trained models can be easily installed. Hence, the proposed model is feasible for practical applications.

VI. CONCLUSION AND FUTURE WORK

Credit scoring is currently a promising research field in data mining. Herein, to address imbalanced data on credit scoring, a novel multi-stage ensemble model with a hybrid genetic algorithm was proposed. First, VIHT approach was proposed to address imbalanced data. Next, a novel HGA approach was proposed and subsequently applied to select optimal feature and classifier subset. Finally, a stacking method was applied to reach the final prediction. Four performance indicators, i.e., BACC, F-score, G-mean, and Recall were used to evaluate the performance of the proposed model. The results demonstrated that the superior performance of the proposed model compared to other benchmark credit scoring models.

In future studies, the demands from the financial industry to reduce the complexity of classification models that are used to obtain prediction results rapidly, will be given serious consideration. The proposed ensemble model requires fair-sized memory and resources in the classifier selection and ensemble stages of the base classifiers. Therefore, more effective strategies for classifier selection and ensemble will be explored to further reduce the complexity and enhance the scalability of the model. In addition, the proposed ensemble model only involves the classic machine learning classifiers instead of deep learning classifiers, which affects the diversity of the base classifiers. In recent years, some scholars have demonstrated the effectiveness of clustering techniques for feature learning in the ensemble model [10], [31]. The clustering techniques proposed by Hu *et al.* [27] and Hu *et al.* [29] have superior clustering performance, which can be integrated into the ensemble model to enhance the model performance in our future work.

REFERENCES

[1] L. Ali, A. Niamat, J. A. Khan, N. A. Golilarz, X. Xingzhong, A. Noor, R. Nour, and S. A. C. Bukhari, "An optimized stacked support vector machines based expert system for the effective prediction of heart failure," *IEEE Access*, vol. 7, pp. 54007–54014, 2019.

[2] L. Ali, I. Wajahat, N. Amiri Golilarz, F. Keshtkar, and S. A. C. Bukhari, "LDA–GA–SVM: Improved hepatocellular carcinoma prediction through dimensionality reduction and genetically optimized support vector machine," *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2783–2792, Apr. 2021.

[3] D. Almhaithawi, A. Jafar, and M. Aljnidi, "Example-dependent cost-sensitive credit cards fraud detection using SMOTE and Bayes minimum risk," *Social Netw. Appl. Sci.*, vol. 2, no. 9, pp. 1–12, Sep. 2020.

[4] A. Asuncion and D. Newman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Tech. Rep., 2007. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

[5] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[6] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[7] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[8] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proc. 20th Int. Conf. Pattern Recognit.*, Istanbul, Turkey, Aug. 2010, pp. 3121–3124.

[9] M. Buckland and F. Gey, "The relationship between recall and precision," *J. Amer. Soc. Inf. Sci.*, vol. 45, no. 1, pp. 12–19, 2014.

[10] S. Chatterjee, S. Bandopadhyay, and D. Machuca, "Ore grade prediction using a genetic algorithm and clustering based ensemble neural network model," *Math. Geosci.*, vol. 42, no. 3, pp. 309–326, Apr. 2010.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Dec. 2012.

[12] N. Chen, B. Ribeiro, A. S. Vieira, J. Duarte, and J. C. Neves, "A genetic algorithm-based approach to cost-sensitive bankruptcy prediction," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12939–12945, Sep. 2011.

[13] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.

[14] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Inf. Fusion*, vol. 41, pp. 195–216, Sep. 2018.

[15] J. Demar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.

[16] B. Efron, "Logistic regression, survival analysis, and the kaplan-meier curve," *J. Amer. Stat. Assoc.*, vol. 83, no. 402, pp. 414–425, Jun. 1988.

[17] E. Fedorova, E. Gilenko, and S. Dovzhenko, "Bankruptcy prediction for Russian companies: Application of combined classifiers," *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7285–7293, 2013.

[18] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[19] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[20] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, 1940.

[21] L. P. F. Garcia, A. C. P. L. F. de Carvalho, and A. C. Lorena, "Effect of label noise in the complexity of classification problems," *Neurocomputing*, vol. 160, pp. 108–119, Jul. 2015.

[22] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

[23] H. He, W. Zhang, and S. Zhang, "A novel ensemble method for credit scoring: Adaption of different imbalance ratios," *Expert Syst. Appl.*, vol. 98, pp. 105–117, May 2018.

[24] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[25] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and Artificial intelligence*. Ann Arbor, MI, USA: Univ. Michigan Press. 1975.

[26] Y. Hong and S. Kwong, "To combine steady-state genetic algorithm and ensemble learning for data clustering," *Pattern Recognit. Lett.*, vol. 29, no. 9, pp. 1416–1423, Jul. 2008.

[27] L. Hu, C. Keith, X. Yuan, and S. Xiong, "A variational Bayesian framework for cluster analysis in a complex network," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 11, pp. 2115–2128, Apr. 2019.

[28] L. Hu, X. Pan, H. Yan, P. Hu, and T. He, "Exploiting higher-order patterns for community detection in attributed graphs," *Integr. Comput.-Aided Eng.*, vol. 28, no. 2, pp. 207–218, 2021.

[29] L. Hu, S. C. Yang, X. Luo, and M. C. Zhou, "An algorithm of inductively identifying clusters from attributed graphs," *IEEE Trans. Big Data*, early access, Jan. 7, 2020, doi: 10.1109/TBDATA.2020.2964544.

[30] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Proc. Int. Conf. Learn. Intell. Optim.* Berlin, Germany: Springer, Jan. 2011, pp. 507–523.

[31] F. Jiang, J. He, and T. Tian, "A clustering-based ensemble approach with improved pigeon-inspired optimization and extreme learning machine for air quality prediction," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105827.

[32] Y. Jin, Y. Liu, W. Zhang, S. Zhang, and Y. Lou, "A novel multi-stage ensemble model with multiple K-means-based selective undersampling: An application in credit scoring," *J. Intell. Fuzzy Syst.*, vol. 40, no. 5, pp. 9471–9484, Apr. 2021.

[33] G. Ke, Q. Meng, T. Finley, and T. Wang, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Conf. Neural Inf. Process. Syst.*, Sacramento, CA, USA, Dec. 2017, pp. 3146–3154.

[34] M. Khushi, K. Shaukat, T. M. Alam, I. A. Hameed, S. Uddin, S. Luo, X. Yang, and M. C. Reyes, "A comparative performance analysis of data resampling methods on imbalance medical data," *IEEE Access*, vol. 9, pp. 109960–109975, 2021, doi: 10.1109/ACCESS.2021.3102399.

[35] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training data sets: One-sided selection," in *Proc. Int. Conf. Mach. Learn.*, Nashville, TN, USA, Jul. 1997, pp. 170–186.

[36] N. Lazarevic-McManus, J. R. Renno, D. Makris, and G. A. Jones, "An object-based comparative methodology for motion detection based on the F-Measure," *Comput. Vis. Image Understand.*, vol. 111, no. 1, pp. 74–85, Jul. 2008.

[37] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for NAS," 2019, *arXiv:1912.06059*. [Online]. Available: http://arxiv.org/abs/1912.06059

[38] W. C. Lin, C. F. Tsai, Y. H. Hu, and J. S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Inf. Sci.*, vol. 409, pp. 17–26, Apr. 2017.

[39] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.

[40] Z. Liu, W. Cao, Z. Gao, J. Bian, H. Chen, Y. Chang, and T.-Y. Liu, "Self-paced ensemble for highly imbalanced massive data classification," in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Jakarta, IN, USA, Apr. 2020, pp. 841–852.

[41] W. Ratajczak, "Principal components analysis PCA," *Comput. Geosci.*, vol. 19, no. 3, pp. 303–342, 1993.

[42] P. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Dept. Math. Statist., Princeton Univ., Princeton, NJ, USA, 1963.

[43] S. Okada, M. Ohzeki, and S. Taguchi, "Efficient partition of integer optimization problems with one-hot encoding," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, Dec. 2019.

[44] S. Oreski and G. Oreski, "Genetic algorithm-based heuristic for feature selection in credit risk assessment," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2052–2064, Mar. 2014.

[45] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 683–697, Sep. 1992.

[46] B. Pan, G. Zhang, J. J. Xia, P. Yuan, H. H. S. Ip, Q. He, P. K. M. Lee, B. Chow, and X. Zhou, "Prediction of soft tissue deformations after CMF surgery with incremental kernel ridge regression," *Comput. Biol. Med.*, vol. 75, pp. 1–9, Aug. 2016.

[47] M. Papouskova and P. Hajek, "Two-stage consumer credit risk modelling using heterogeneous ensemble learning," *Decis. Support Syst.*, vol. 118, pp. 33–45, Mar. 2019.

[48] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[49] H. T. Rauf, W. H. K. Bangyal, and M. I. Lali, "An adaptive hybrid differential evolution algorithm for continuous optimization and classification problems," *Neural Comput. Appl.*, vol. 33, pp. 10841–10867, Jun. 2021.

[50] D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man Cybern.*, vol. 21, no. 3, pp. 660–674, May 1991.

[51] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, Jun. 1990.

[52] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010.

[53] M. R. Smith, T. Martinez, and C. Giraud-Carrier, "An instance level analysis of data complexity," *Mach. Learn.*, vol. 95, no. 2, pp. 225–256, 2014.

[54] J. Sun, J. Lang, H. Fujita, and H. Li, "Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates," *Inf. Sci.*, vol. 425, pp. 76–91, Dec. 2018.

[55] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, 2009.

[56] G. Wang, J. Hao, J. Ma, and H. Jiang, "A comparative assessment of ensemble learning for credit scoring," *Expert Syst. Appl.*, vol. 38, pp. 223–230, Jan. 2011.

[57] Z. Y. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. De Freitas, "Bayesian optimization in high dimensions via random embeddings," in *Proc. International Joint Conf. Artif. Intell.*, Beijing, China, Aug. 2013, pp. 1778–1784.

[58] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992.

[59] M. Zeng, B. Zou, F. Wei, X. Liu, and L. Wang, "Effective prediction of three common diseases by combining SMOTE with tomek links technique for imbalanced medical data," in *Proc. IEEE Int. Conf. Online Anal. Comput. Sci. (ICOACS)*, Chongqing, China, May 2016, pp. 225–228.

[60] H. Zhang, H. He, and W. Zhang, "Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring," *Neurocomputing*, vol. 316, pp. 210–221, Nov. 2018.

[61] W. Zhang, H. He, and S. Zhang, "A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring," *Expert Syst. Appl.*, vol. 121, pp. 221–232, May 2019.

[62] M. Zięba, S. K. Tomczak, and J. M. Tomczak, "Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction," *Expert Syst. Appl.*, vol. 58, pp. 93–101, Oct. 2016.

YILUN JIN is currently pursuing the M.S. degree with the School of Information Management and Artificial Intelligence, Zhejiang University of Finance and Economics, China. His current research interest includes credit scoring and fair classification.

WENYU ZHANG received the Ph.D. degree from Nanyang Technological University, Singapore, in 2002. He is currently a full-time Professor with the School of Information, Zhejiang University of Finance and Economics, China. He has published more than 60 articles in international journals and more than 20 papers in international conference proceedings in the recent ten years, covering supply chain management, digital library, bibliometrics, concurrent engineering, distributed manufacturing, credit scoring, business analytics, data mining, multi-agent technology, and semantic web.

XIN WU is currently a full-time Associate Professor with China Academy of Financial Research, Zhejiang University of Finance and Economics, China. He has published more than 20 articles in international journals, covering data mining, financial technology, and sustainable development.

YANAN LIU received the Ph.D. degree in computer science and technology from Zhejiang University, China, in 2010. She is currently a full-time Associate Professor with the School of Information Management and Artificial Intelligence, Zhejiang University of Finance and Economics, China. She has published more than 20 articles in international journals, covering data mining, artificial intelligence, and credit scoring.

ZEQIAN HU is currently pursuing the B.S. degree with the School of Information Management and Artificial Intelligence, Zhejiang University of Finance and Economics, China. His current research interests include machine learning and data science.