Multiple Optimized Ensemble Learning for High-Dimensional Imbalanced Credit Scoring Datasets

Sudhansu R. Lenka (sudhansulenka2000@gmail.com)
C.V.Raman Global University
Sukant Kishoro Bisoy
C.V.Raman Global University
Rojalina Priyadarshini
C.V.Raman Global University

Research Article

Keywords: Class imbalanced data, Optimization subset, Feature selection, Ensemble learning, Credit scoring, Resampling

Posted Date: April 3rd, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2757867/v1

Abstract

Banks determine the financial credibility or the credit score of the applicants before allocating loans to them. In recent decades, several machine learning algorithms have been developed to automate the decision-making process by constructing an effective credit scoring models. However, the high-dimensional and imbalanced credit datasets significantly degrade the models' classification ability. In this study to overcome these issues, a novel multiple-optimized ensemble learning (MOEL) is proposed to build a reliable and accurate credit scoring model. MOEL, first generates multiple diverse optimized subsets from various weighted random forests (WRFs), and from each subset more effective and relevant features are selected. A new evaluation measure is then applied to each subset to determine which subsets are more effectively optimized for the ensemble learning process. The subsets are then applied to a novel oversampling strategy to provide balanced subsets for the base classifier, which lessens the detrimental effects of imbalanced datasets. Finally, to further improve the performance of the base classifier, a stacking-based ensemble method is applied to the balanced subsets. Six credit-scoring datasets were used to evaluate the model's efficacy using the F1 score and G-mean metrics. The empirical results on these datasets demonstrate that MOEL achieves the best value of F1_score and G-mean with a mean ranking of 1.5 and 1.333, respectively.

1. Introduction

Financial system risks are delicate subjects that require investigation while taking into account several crucial elements. Banks are subject to a variety of risks, which have a big impact on the country's economy. The capacity to predict a potential applicant's credit score is one of the fundamental components of financial systems. Lenders were able to estimate applicants' capacity to repay their debts within a specified time frame by using the credit scoring approach to determine the likelihood that applicants would default on their loans. In other words, lenders assess an applicant's creditworthiness before approving a loan for them. These credit scoring methods aid in differentiating between creditworthy and untrustworthy borrowers. Therefore, accurate and efficient credit scoring is essential for assisting financial institutions in determining an applicant's default risk.

In recent decades several machine learning-based credit scoring models have been proposed [1], [2]. However, accurately predicting the creditworthiness of an applicant is a potential area of research. The performance of the models is greatly decreased by the high proportion of redundant and invalid features that may be present in the high-dimensional data [2]. The feature selection approaches not only reduces the dimensionality but also selects the highly informative features from the dataset. Every feature carries different information to distinguish the customers. The study of [3] employed four feature selection procedures, namely Genetic Algorithm (GA), Information Gain Ratio (IGR), Relief, and Principal Component Analysis (PCA). In our previous study [4], three set of feature selection techniques, such as Information gain, PCA, and GA are used to select the relevant features form the credit datasets. For improving the classification performance of these models, the researcher of [5] employs three feature selection techniques, i.e. correlation matrices, classification and regression trees, and PCA. The hybrid model of [6] used five feature selection techniques to create a powerful credit scoring model, including classifier feature selection, correlation feature selection, information gain, gain ratio, and relief. A high-dimensional imbalanced credit dataset makes it challenging to choose an optimum

feature subset. Along with high-dimensional data, the class imbalance has grown to be a very difficult problem for the classical classifier [7].

The imbalanced class distribution refers to the uneven distribution of good (majority) and bad (minority) credits, where the number of bad credits is significantly lower than the good credits. Due to potential bias toward the majority class instances in such imbalanced datasets, the classical classifier may misclassify examples belonging to the minority class. Therefore, if an applicant having bad credit is incorrectly predicted as good, the financial industry could experience a significant economic loss [8][9]. Therefore, essential steps should be taken to enhance the models' performance in imbalanced datasets.

Different strategies have been used by researchers to improve the accuracy of predictions by correcting errors brought on by imbalanced datasets. The most common method used to balance the skewed credit scoring datasets is resampling [10][11]. It includes under-sampling, over-sampling, and hybrid sampling. Each approach is simple to use, but each one also provides a set of restrictions that could hinder classification performance. For instance, under-sampling strategies balance class distributions by arbitrarily removing the majority class samples, but they risk losing important data that will have a significant impact on performance. By producing additional samples from minority classes, the oversampling strategy balances the dataset but may overfit or accentuate the errors [12]. The unbalanced data distribution is balanced using hybrid approaches, which mix oversampling and under-sampling techniques.

Ensemble learning approaches have been used by researchers in recent years to deal with imbalanced datasets [13]. In ensemble learning, many base classifiers are trained, and each classifier's responses are merged to obtain results that are more accurate and reliable [14] -[16]. The two key pre-requisite conditions that must be prioritized to produce an optimal ensemble model are diversity and accuracy [17], [18]. Resampling and ensemble learning combined offer an alternative approach to class imbalance issues. Resampling enabled ensemble approaches to provide numerous balanced subsets for classifier training. However, using ensemble methods to resample the original feature subset may cause the classification performance to deteriorate. The performance of the models would deteriorate if there are irrelevant features in high-dimensional imbalanced datasets [7].

In this work, we propose a multiple-optimized ensemble learning (MOEL) approach for high-dimensional imbalanced credit-scoring datasets. First, the training data is run through the weighted random forest, which creates several selective and optimized subsets. The top-ranked features are incrementally added to the feature subset from each WRF to create an optimized feature subset. The generalization prediction abilities of the optimized subsets may vary because they were produced from various WRF's. Therefore, to obtain better-optimized subsets a new evaluation metric is applied to each optimized subset that considers the weight and density of each subset. Based on these two factors the top-ranked subsets are selected as better-optimized subsets. Next, a novel oversampling strategy is applied to these subsets to mitigate the negative effects of imbalanced data distribution on the base classifiers. The proposed oversampling strategy generates new synthetic minority instances by computing the average of two dissimilar minority instances. The newly generated instances are unique and evenly distributed within the minority class region which minimizes the over-generalization problem. Thus, this evenly distributed oversampling strategy enhances the diversity within the balanced subsets. Further, a stacking-based ensemble method is incorporated into these balanced subsets

to train the base classifiers. The parameters of the best-performing classifiers are optimized by the ensemble method which eventually optimizes the subsets. Finally, the optimized ensemble model is employed to predict the responses of the test set, and the final output could be compiled using the soft-voting approach [56]. We conduct experiments on six real-world credit scoring datasets to show MOEL's enhanced performance in contrast to other oversampling and ensemble techniques. The results of the experiments are compared by using F1_score and G-mean metrics. The following list summarises the paper's main contributions:

1. Multiple optimized subsets are generated by applying WRFs on the training set, the subsets not only eliminate the redundant and unimportant features but also produce discriminant and diverse optimized subsets for the ensemble system.
2. To produce more optimal subsets, we employed a unique evaluation metric-weighted density-based to each optimized subset.
3. A novel oversampling technique is applied to each optimized subset to generate balance subsets for the base classifiers.
4. Further, a stacking-based ensemble method is applied to the balanced subsets to train and optimizes the base classifiers.
5. To show the superiority of MOEL, extensive comparison experiments are carried out on a set of real-world high-dimensional imbalanced credit datasets.

The remainder of the paper is structured as follows. Section 2 presents some relevant research on resampling techniques. In Section 3, the proposed technique is discussed in detail. Experimental settings and the results of the experiments are discussed in Section 4. Finally, in Section 5 the conclusions with future research are discussed.

2. Related Work

In this section, different credit scoring evaluation strategies and class imbalance learning methods are discussed. Credit scoring evaluation methods can be broadly classified into three approaches, i.e. expert-based approaches, statistical methods, and machine learning (ML) techniques [1].

In the beginning, the loans are sanctioned based on the subjective judgment of the expert team members. Experts make decisions regarding whether or not to approve loans using their expert knowledge and expertise. In such approaches accurately estimating the creditworthiness of the applicants is a challenging task even for experienced persons [20]. With the advancements in data science, more accurate and effective credit scoring models were proposed. Models constructed using statistical methods are more accurate than those created using methods based on experience [21], [22]. These statistical methods may be used to investigate the linear relationships between the variables and the probability of default, but they were unable to identify the non-linear relationships between various variables [22]. Several ML techniques have been investigated in recent decades to develop credit-scoring models. These models outperformed more established statistical techniques in terms of classification performance [23]. Credit scoring models were created using ML algorithms such as Artificial Neural Networks (ANN) [24], Decision Trees (DT) [25], and Support Vector Machines (SVM) [26]. For example, DT is used in credit scoring models to identify the non-linear mapping between the independent and the dependent variable [25].

Class imbalance regularly happens in many real-world situations when there are significantly more members of one class than there are members of other classes [27], [28] and in such situations, ML algorithms very often misclassify the minority class samples. Correctly predicting the defaulter clients is more crucial in credit-scoring datasets than the non-defaulters [29].

Sampling is the most common approach used to balance skewed datasets for improving classification performance [30]. Random undersampling (RUS) balances the class distributions by arbitrarily eliminating the samples from the majority class. However, it might exclude some valuable information from the dataset, which would worsen the performance [31]. In our previous study [2], a clustering-based undersampling strategy is proposed which balances the uneven class distributions by selecting the more informative majority samples from the sub-clusters. The empirical analysis shows that our method outperforms other classical resampling methods by using different performance measures, such as precision, recall, F1_score, and AUC (Area Under the Curve). In the study [33], two clustering-based approaches are proposed to address the imbalance issue, in the first approach the cluster centers of the majority class subset are used to balance the dataset, and in the second approach, the instances closer to the cluster centers are used in the resampling. Experimental results illustrate that the second strategy produces better results than the first one. The oversampling strategy balances the imbalanced dataset by generating new artificial minority samples. Random oversampling (ROS) balances the imbalance dataset by randomly duplicating the minority samples. Although ROS is simple and easy to implement, it may cause overfitting [3]. In addition to ROS, several effective oversampling methods have been proposed, including SMOTE [4], Borderline-SMOTE [5], and ADASYN [6] which implement similar approaches for balancing the imbalanced class distributions. These techniques choose samples from the minority class subset to create synthetic instances using the K-nearest neighbor (KNN) method. These methods locate the closest neighbors of the data instance under consideration using the Euclidean distance metric. SMOTE generates synthetic instances by linearly interpolating two nearest minority instances. Borderline-SMOTE chooses the minority class instances that are located in the boundary region or lie near the boundary. ADASYN builds synthetic samples around the minority class instances that are difficult to learn. Since these oversampling methods generate new instances based on the KNN algorithm, they have less chance of generating diverse data points. Therefore, a series of efficient oversampling strategies, like the majority-weighted minority oversampling technique (MWMOTE) [38], have been developed to get around these limitations. Douzas *et al.* [7] propose an oversampling by combining K-means and SMOTE, which balances the class distribution by generating synthetic instances within the minority class sub-clusters. In SMOTE-ENN [40], a SMOTE-based oversampling technique, in which undesirable synthetic samples produced by oversampling are removed by applying an edited nearest neighbour (ENN).

Boosting is another technique that addresses class imbalance problems. The most well-known boosting method, AdaBoost [41], sequentially uses weak classifiers to raise classification performance. AdaBoost has been modified in several ways to handle class imbalance issues, including SMOTEBoost [44], an ensemble-based oversampling method that combines AdaBoost and SMOTE to provide a powerful hybrid oversampling method for unbalanced datasets. Similar to SMOTEBoost, RUSBoost [45] combines RUS and AdaBoost and is a faster, simpler oversampling technique that performs better on imbalanced datasets. Since the performance of the classification may be impacted by the presence of noisy data points in the minority class subset. Hence, Kang *et al.*[8] filter the noisy samples from the multi-class imbalanced ensemble model, which improves the

classification performance of the model. Galar *et al.*[9] in their paper, implement different ensemble techniques to tackle the imbalance datasets and concluded that integrating RUS with various ensemble approaches, such as bagging and boosting, could result in better classification.

Although the class unbalanced issues have been somewhat mitigated by all the studies listed above, these models still have some limitations when handling high-dimensional imbalanced datasets. The performance of the classifier was significantly hampered by the irrelevant features in the imbalanced datasets because most of these methods directly applied the resampling techniques on the original feature space without implementing any feature selection or extraction techniques. Additionally, in the KNN-based methods, the new samples may overlap with the majority class region if the neighborhood instances are not closely associated and thus provide no additional informative information to the classifier.

3. Proposed Model

Let $T = \{(x_i, y_i) | i = 1,2, \cdots n\}$ a training dataset, where x_i denotes an instance and $y_i \in [0,1]$is its corresponding output class. A sample is a member of the majority or negative class if $y_i = 0$ and it belongs to the minority or positive class if $y_i = 1$. To build an effective credit scoring model for high-dimensional datasets, a novel multiple-optimized ensemble learning (MOEL) method is proposed. The overall process of MOEL is illustrated in Fig. 1, which consists of four main steps. In the first step, the training set is partitioned into multiple subsets using weighted random forests to construct diverse and optimized subsets. In the second step, a new evaluation metric is applied on the subsets by considering the data distribution in each subset, which helps to select better-optimized subsets. In the third step, to balance the class distribution in each optimized subset a novel oversampling method is implemented, Finally, a stack-based ensemble method is implemented to build an optimized credit scoring model. The detailed mechanisms of the above steps are discussed in the following sub-sections.

3.1. Generation of multiple diverse optimized subsets (MDOS)

The redundant and irrelevant features in the high-dimensional imbalanced datasets have negative impacts on the credit scoring models, which causes the banks to suffer significant financial losses. The adoption of resampling on the original feature space of the high-dimensional imbalanced dataset may not only amplifies the error rate but also increases the training time. As a result, in addition to resampling, choosing the relevant features is crucial for classification to increase the model's performance. To overcome the above two issues, the training set is segmented into a multiple divese optimized subsets (MDOS) to choose the relevant features from the high-dimensional imbalanced dataset. To make the subsets more diverse and discriminate, WRFs are applied to each subset which consists of multiple random trees with different class weights. The WRFs are adopted because of the following reasons: 1) in each subset, it generates enough diversities by randomly selecting both the features and the instances, 2) based on the data from split nodes, the relevance of the features in the random tree is calculated, 3) it also handles the uneven class distributions by adjusting the weights of the samples, and 4) combining the weighted random trees helps to build a more accurate and effective predictive model.

To lessen the negative impact of the imbalanced data, WRF assigns more weight to the minority class instances to increase their importance. The weight of the samples in the ithweighted random tree is defined as:

$$w_i^C = \frac{N_i}{C * N_i^C}$$

1

where W_i^c represents the weight of the instances of cth class. N_i represents the total number of instances and N_i^c represents the number of instances belonging to cth class.

In this work, the relevance of the features for each random tree is calculated using the Gini index [47]. In this study, the non-creditworthy applicants are considered as positive and the creditworthy applicants as negative samples. The Gini index explains how the jth node of the ith weighted random split the total number of samples into positive and negative samples. Mathematically, it is expressed as:

$$G_{i,j} = 1 - p_{i,0}^2 - p_{i,1}^2$$

2

where $p_{i,0}$ and $p_{i,1}$ is the number of negative and positive samples in the jth node of the ith weighted tree, respectively.

The importance of feature f is computed by considering both the weight of samples and the Gini value at each jth node of the ith weighted tree before and after branching, which is expressed as:

$$IM_{i,j}\left(f\right) = \frac{\sum_{c=0}^{1} W_{i,c} * N_{i,j,c}}{N_i} * G_{i,j} - \frac{\sum_{c=0}^{1} W_{i,c} * N_{i,j\_l,c}}{N_i} * G_{i,j\_l} - \frac{\sum_{c=0}^{1} W_{i,c} * N_{i,j\_r,c}}{N_i} * G_{i,j\_r}$$

3

where the weighted number of samples and the Gini value of the jth node before splitting are represented by the first part of Eq. (3) and the weighted number of samples and the Gini value after splitting are represented by the other two parts. $N_{i,j,c}$ represents the number of samples belonging to cth class. The numbers $N_{i,j\_l,c}$ and$N_{i,j_r,c}$, respectively, represent the number of samples of the cth class that are located in the left and right children of the jth node in the ith tree. $G_{i,j_l}$and G_{i,j_r} represents the Gini value of the left and right child for the jth node, respectively.

The importance of each feature $f \in F_i$ in the ith weighted random tree is defined as:

$$IM_i\left(f\right) = \sum_{j \in F_i} IM_{i,j}$$

4

Finally, the importance of the features can be computed by combining their importance in all random trees of the WRF, which is expressed as:

$$FIM\left(f\right)=\sum_{i=1}^{m}IM_{i}\left(f\right)$$

5

where m = number of trees in the WRF.

To obtain the optimized feature subset (FS), the features with the highest FIM values are incrementally added to the subset. The feature selection procedure starts with an empty subset, and the performance of the classifier M is evaluated after each feature is added to the subset. F1 score is utilized in the experiment as a performance metric because it is considered to be the more appropriate for assessing the model's predictive ability even in the imbalanced dataset [47]. The feature with the highest FIM value is added to the FS. The performance obtained by the classifier is considered as a baseline and then sequentially the subset is expanded by adding more informative features. After each addition, the performance of the current FS is compared against the baseline. If the current FS's F1_score is better than the baseline, then it is considered as the current best FS. In each iteration, the next best feature is added to the current best FS, and the performance of the newly obtained FS is compared against the current best FS. If the newly generated subset achieves better results than the current best FS, then the newly generated FS will be considered as the current best FS. This process is repeated until the classifier's performance is evaluated against all the features. Finally, the optimized FS is used in the subsequent steps. By repeating this process on different WRFs, a set of optimized feature subset $FS_{i}\ ,i=\{1\cdots M\}$ is generated. This rank-based feature selection process helps to minimize the invalid and redundant features from the high-dimensional imbalanced credit dataset. Algorithm 1 presents the pseudocode of MDOS.

Algorithm 1: MDOS

Input: Training dataset T = $\{(x_i, y_i)\}_{i=1}^{n}$ with F number of features and m number of WRFs

Output: Optimized feature subset FS

Procedure:

1. $for i = 1,2, \cdots m$ do

2. Build a WRF_i using T

3. Compute the FIM_i of each feature of the WRF_i using (1)-(5)

4. $FS_i = \{\}$

5. Select the feature f_1 with highest FIM_i value as baseline feature subset i.e., $FS_i = \{f_1\}$

6. The performance of the classifier is checked by using f_1 i.e. $baseline = F1\_score\,(f_1)$.

7. $for j = 2 to F$ do

8. Select the next best feature f_j and the performance is checked by using the feature subset $\left\{ f_1 \cup f_j \right\}$

9. If $F1\_score\left(f_1 \cup f_j\right) > baseline$, then $FS_i = \left\{ f_1 \cup f_j \right\}$

10. **end for**

11. **end for**

3.2. Selection of better-optimized subsets

In the previous step, a set of multiple diverse optimized subsets are generated by implementing a rank-based feature selection process on the WRFs. Since the subsets are obtained from different WRFs, they may have different generalization capabilities. The major goal of this stage is to pick the optimized subsets with the best performance by evaluating each optimized subset using a newly developed evaluation metric called weight density. By using this evaluation metric, the weight, and density of each sample of the subset are assessed to produce more optimized subsets by taking into account the feature space and the imbalanced class distribution.

Due to imbalanced class distributions, the minority class instances are given more importance by assigning more weights than the majority class instances. The weights of the majority class W_{maj} and the minority class samples W_{min} are defined as:

$$W_{maj} = 1$$

6

$$W_{min} = |T_{maj}| \,/\, |T_{min}|$$

7

where $|T_{maj}|$ and $|T_{min}|$ represents the number of majority and minority class instances in the optimized subset, respectively.

Additionally, the density of the instances in each optimized subset is calculated by counting the number of instances of neighboring classes. Through this density factor, we can eliminate the noisy samples from the dataset. If a particular instance x_i's nearest neighbors $KNN\left(x_i\right)$ exclusively contain examples of a different class, that instance will be regarded as noise and could harm the model's performance.

In each subset, the density of the instance x_i is computed by finding its K-nearest neighbors,$KNN\left(x_i\right)$ .Let $KNN\left(x_i\right) = N_{i,maj} \cup N_{i,min}$, where $N_{i,maj}$and $N_{i,min}$ be the majority and minority set located in the neighborhood of x_i. For each $x_i$belonging to jth optimized subset, its density is defined as:

$$D\left(x_i, j, min\right) = \frac{|N_{i,maj}|}{K}$$

8

$$D\left(x_i, j, maj\right) = \frac{|N_{i,min}|}{K}$$

9

where $D\left(x_i, j, min\right)$ and $D\left(x_i, j, maj\right)$ represents the density of minority and majority class instances, respectively.

The weight-density measure of the jth optimized subset is computed by combining the weights and density, which is expressed as:

$$WD_j = W_{maj} \times D\left(x_i, j, maj\right) + W_{min} \times D\left(x_i, j, min\right)$$

10

When an instance neighborhood contains more instances from a different class, the value of WD_j increases. As a result, the decision boundary becomes fuzzier and more complicated, increasing the likelihood that this instance may be wrongly classified, which lowers the ability to generalize. Therefore, to achieve better-

optimized ensemble learning, we select the top-α optimized subsets $(_1, _2, \cdots)$ having a minimum value of WD_j. The process of generating optimized subsets for ensemble learning is presented in Algorithm 2.

Algorithm 2: Selection of better-optimized subsets
Input: Training dataset T = $\{(x_i, y_i)\}_{i=1}^{n}$ with F number of features and m number of WRFs
Output: Better optimized subsets
Procedure:
1. $for i = 1,2, \cdots m$ do 2. Implement **Algorithm-1** to obtain feature subsets $FS_i \ (i = 1, 2, \cdots, m)$ 3. Set the weights of majority and minority instances using (6) and (7), respectively 4. Compute the density of the minority and majority class instances using (8) and (9), respectively. 5. Compute the weighted density measure WD_i of each feature subset using (10) end for 6. Select top-α optimized subsets with minimum value of WD i.e., $= (_1, _2, \cdots)$

3.3. Resampling of optimized subsets

Each of the optimized subsets $\varnothing_i (i = 1,2, \cdots n)$ are used to build the base classifiers of the stack-based ensemble model. To increase the effectiveness of the ensemble model, a novel oversampling method is applied to each optimized subset to balance the subsets. To oversample the subsets, the Mahalanobis distance metric is adopted to compute the distance between the data samples. It is adopted due to the following reasons: 1) it could able to recognize the highly correlated data and the outliers in the dataset, and 2) the elimination of these data samples could enhance the performance of the model. The Mahalanobis distance $\left(Dist^2\right)$ among the minority sample $x_i^{min} \in \varnothing_i$ is computed as::

$$Dist_i^2 = (x_i - \mu)^T \sum^{-1} (x_i - \mu)$$

11

where μ is the mean of the T_{min} and $\sum$ is the covariance matrix. The minority samples are initially organized according to the $Dist^2$ values in descending order. Next, the distance vector ($Dist^2$) is segregated into two equal partitions, and then sequentially a unique label is assigned to each sample of both partitions. The instances of $partition1$ have $Dist^2$ values greater than or equal to the $Dist^2$ value of the middle instance, while the rest instances belong to the $partition2$.

$$partition1 = \left(x_1^{min}, x_2^{min}, \cdots x_{\frac{n}{2}}^{min}\right) and partition2 = \left(x_{\frac{n}{2}+1}^{min}, x_{\frac{n}{2}+2}^{min}, \cdots x_n^{min}\right)$$

12

where n is the number of minority examples in the optimized subset. Then, sequentially a unique label l_i is assigned to each instance of both partitions. The first instance of both partitions is assigned the same label, and similarly, the other instances of both partitions are labeled. The instances of both partitions are labeled as:

$$parition1\left(label\left(x_i^{min}\right)\right) = l_i, i = \{1, \cdots, n/2\}\, and$$

$$partition2\left(label\left(x_i^{min}\right)\right) = l_{i-n/2}, i = \left\{\frac{n}{2} + 1, \cdots, n\right\}$$

13

The final stage involves creating synthetic instances by averaging two instances from each partition with the same label, which is defined as:

$$x_i^{syn\_min} = \left(x_i + x_j\right)/2$$

14

where x_i and x_j are the instances of partition1 and partition2 of the same label. The synthetic instances are added to the minority subset of the optimized subset$\varnothing_i$. The newly generated instances are uniquely distinct but are related to the instances of both partitions. The same procedure is sequentially repeated for the remaining instances, and the same label is given to each new instance as that of their parents. This procedure is done until the optimized subset is balanced. The advantages of the proposed oversampling methods are 1) Contrary to SMOTE or KNN-based techniques, which generate the synthetic instances within a specific cluster of the minority class subset and give the classifier no additional information, the synthetic instances produced by the proposed method are evenly distributed throughout the minority class decision boundary, 2) the newly generated instances lies strictly within the region of the minority class instances, hence no overlapping of synthetic instances with the majority class takes place. The pseudocode of the proposed Mahalanobis-based oversampling method is presented in Algorithm 3.

3.4. Stacking-based ensemble method

In the final step, a stacking-based ensemble method with self-adaptive parameter optimization is adopted on the optimized balanced subsets to train the base classifiers. The performance of the base classifiers can be enhanced by optimizing their parameters. In this method, the parameters of the base classifiers are auto-optimized, and hence it improves the performance of the ensemble model.

As shown in Fig. 2, the base classifiers ($clf_1, clf_2, \cdots clf_n$) like logistic regression (LR), random forest (RF), XGBoost, gradient boosting decision tree (GBDT), and light gradient boosting machine (LGBM) are employed. The algorithms are trained on the optimized balanced subsets and their performances are validated on the validation set. Next, the top m classifiers ($sclf_1, sclf_2, \cdots sclf_m$) are selected based on their superior performances in terms of F1_score. To further improve the performance of the selected classifiers, their

parameters are optimized using a hyperparameter optimization framework [48]. The predicted response of the optimized classifiers $(ocl f_1, ocl f_2, \cdots ocl f_m)$ is stacked and integrated by a meta-classifier to obtain the final predicted output. As a meta-classifier LR is employed because of its high predictive performance in the stacked-based ensemble method [8].

Algorithm-3: Mahalanobis-based Oversampling Technique

Input: Optimized subsets $\varnothing_i$, minority class subset T_{min}, and majority class subset T_{maj}

Output: Balanced subsets

Procedure:

1. $for i = 1,2,\cdots m$ do

2. Find the number of new synthetic samples to be generated, i.e. $r = \left|\left|T_{maj}\right|\right| - |T_{min}|$

3. For each $x_i \in T_{min}$, its Mahalanobis distance $Dist^2$ is computed using (11).

4. Sort the minority instances according to their $Dist^2$ values in descending order.

5. Divide the minority class subset into two equal partitions, i.e. $mid = n/2$.

6. For each $x_i \in partition1$ and $y_i \in partition2$, a sequentially unique label is assigned using (13).

7. Initialize $c = 0$

8. Initialize synthetic minority subset, $T_{min}^{syn} = \{\}$

9. for j = 1,2, $\cdots n/2$ do

10. select x_i and $x_{i+\mathbb{2}}$ from partition1 and partition2, respectively, such that $label\left(x_i\right) = lable\left(x_{i+\mathbb{2}}\right)$

11. Generate synthetic instance x by averaging the instances x_i and $x_{i+\mathbb{2}}$

12. $T_{min}^{syn} = T_{min}^{syn} \cup \{x\}$.

13. $c = c + 1$.

end for

Algorithm-3: Mahalanobis-based Oversampling Technique

14. If $c < r$, pair the synthetic minority instances T_{min}^{syn} with the samples of both the partitions and repeat steps (9)-(13). If still$c < r$, then pair the instances of the current T_{min}^{syn} with the immediate previous level and later with the predecessor levels and for each subsequent level repeat steps (9)-(13).

15. If$c \geq r$, the generated synthetic minority samples are combined with the original minority instances, $T_{gmin} = T_{min} \cup \left\{T_{min}^{syn}\right\}$.

16. Finally, the optimized subset gets balanced, i.e. $_{i\_bal} = T_{gmin} \cup T_{maj}$.

end for

4. Experimental Setup And Results

This section outlines the experimental design and results. To verify the efficacy of the suggested strategy, the experimental setup consists of credit scoring datasets and evaluation metrics. The effects of the parameters used in the proposed method are discussed in the results section. The effectiveness of the MOEL is then confirmed by comparing it to other traditional and ensemble-based oversampling methods, and finally, nonparametric statistical tests are used to demonstrate its superiority to other resampling methods.

4.1. Credit scoring datasets and pre-processing

Six credit-scoring datasets from the UCI machine learning repository were used in this study. It contains credit datasets from Australia, Germany, Japan, Taiwan, and Polish datasets [49], which are utilized in the trials to assess how well the proposed model works. The Polish credit dataset includes two separate datasets i.e., the Polish 1, and the Polish 2. A detailed description of these datasets is provided in Table 1, which contains the number of instances, number of features, number of minority and majority instances, and imbalance ratio (IR). The Australian Credit Scoring dataset includes 690 customer credit records, of which 307 samples are positive and 383 samples are negative. There are 15 attributes total, including the class label, of which 6 are numerical and 9 are category. To be compatible with machine learning techniques, all categorical variables in the datasets are transformed into numerical values. Similarly, the details of other datasets are shown in Table 1.

In the pre-processing step, the mean value of each column is used to fill up the gaps left by the missing values. Then mean and unit variance is taken out to standardize the quantitative features. The raw data used in the studies are split into two groups: training (80% of the data) and testing (20% of the data). The training set is then split into training and validation sets using nested 5-fold cross-validation. It has been widely accepted that cross-validation outperforms traditional train-test split in terms of performance. Ultimately, 64%, 16%, and 20% of the data were allocated to training, validation, and testing, respectively. The five-fold cross-validation is repeated five times to lessen the impact of randomness on the predictions and each subset is trained using a

decision tree algorithm. In the stacking-based ensemble method, the base classifiers with default parameters were implemented. The base classifiers LR, RF, and GBDT were implemented by importing 'sklearn' python module; the base classifiers XGBoost and LGBM were implemented by importing 'xgboost' and 'lightbgm' python module, respectively.

Table 1
Description of the credit scoring datasets

Dataset	Instance size	Number of features (Categorical/ Numerical)	Number of Majority Instances	Number of Minority Instances	IR
Australian	690	15 (9/6)	383	307	1025
German	1000	21 (14/7)	700	300	2.33
Japanese	690	16 (11/5)	383	307	1.25
Taiwan	30,000	24 (9/15)	23,364	6636	3.31
Polish 1	7027	65 (1/64)	6756	271	24.93
Polish 2	10,173	65 (1/64)	9773	400	24.43

4.2. Evaluation metrics

In this study, we have applied two evaluation metrics such as F1_score and G-mean to evaluate the efficacy of the proposed method. In credit scoring models, the good and bad applicants are considered as negative and positive samples, respectively. Positive and negative class samples that are correctly classified are denoted by the terms TP and TN, while the wrongly classified positive and negative class samples are denoted by the terms FP and FN. All of these metrics are defined using the components of the confusion matrix (given in Table 2).

Table 2
Confusion Matrix

	Predicted Positive Class	Predicted Negative Class
Actual Positive (Bad)	True Positive (TP)	False Negative (FN)
Actual Negative (Good)	False Positive (FP)	True Negative (TN)

The precision and the recall are defined in Eq. (15) and Eq. (16), respectively. The F1_score evaluates the accuracy of the positive (or bad) samples, which is defined in Eq. (17) by computing the average of precision and recall.

The G-mean or geometric mean defines the overall classification performance by computing the accuracy of creditworthy (good) and default (bad) applicants, which is defined in (18).

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

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

16

$$F1\_score = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

17

$$G - mean = \sqrt{Sensitivity \times Specificity}$$

18

4.3. Setting the number of weighted random forests$T$

In the proposed model, the most important parameter is determining the number of weighted random forests (WRFs)T. In MOEL, a different set of random trees are used and its effect on the performances is analyzed. Figure 3 displays the outcomes of its F1 score performance after employing various numbers of random trees. It has been observed that in most of the data sets the F1_score value is maximum when the size of T is set to 15. When the number of random forests is less than 15, the F1_score value gradually increases and when it is greater than 15, the value gradually decreases. This happens because by increasing the number of random forests more diverse optimized subsets will be obtained. However, there may be the chance of generating redundant optimized subsets or some subsets may not achieve the required generalization ability. This indicates that the model's performance may degrade if the number of random forests is increased arbitrarily. Therefore in this study, we set the default value of T to 15.

4.4. Setting the number of optimized subsets,α

The optimized subsets that are generated from different WRFs may have different generalization performance behavior. To select the better-optimized subsets a novel evaluation metric has been used in the proposed model. This metric helps to select the better-optimized subsets that could enhance the ensemble learning performance. In this experiment, we have adopted a different number of optimized subsets to train the base classifiers. F1 score analysis is used to assess the effectiveness of the classifiers. Figure 4 presents the average F1 score values. It has been noted that when the number of optimized subsets is set to 4, the base classifiers have attained the highest F1 score. In the experiment, decision trees are used as a base classifier. Therefore, in the proposed method to achieve better classification results, we set the default value of α to 4.

4.5. Comparison of the proposed oversampling with traditional oversampling methods

Our proposed approach (MOEL) is compared with different imbalanced learning methods, i.e. SMOTE [4], Borderline_SMOTE (BSMOTE) [5], ADASYN[6] and random oversampling (ROS). In each of the K-nearest-based oversampling techniques i.e. SMOTE, BSMOTE, and ADASYN the K-value was set to 5. Tables 3–4 compare the outcomes of each approach using the F1 score and G-mean, respectively. In the experiment, all of the oversampling strategies are classified using k-nearest neighbors (k = 5) and the decision tree, while in MOEL,

the decision tree serves as the base classifier. The decision tree with default parameters is used in all the oversampling techniques. From Table 3, it can be shown that our proposed strategy outperforms other imbalanced learning approaches and gets the best F1 score value on 5 out of the 6 datasets. In the case of G-mean, our approach outperforms other approaches on all the datasets (illustrated in Table 4). For example, MOEL achieves the F1_score value of 75.61, 85.59, and 85.23 on German, Australian, and Japanese datasets, which are 7.35%, 2.84%, and 5.45% higher than the second-best performer, respectively. Similarly, it achieves the G-mean of 83.26, 86.95, and 85.27 on German, Australian, and Japanese datasets which are 4.36%, 3.38%, and 3.27% higher than the second-best performer, respectively. This is because the K-nearest neighbor approach, which is used in KNN-based oversampling techniques, tends to cluster in a specific area of the minority class region and thus gives the classifier no additional information, as opposed to our approach, which creates synthetic samples that are evenly distributed within the convex hull of the minority class samples and thus gives the classifier more useful information. Additionally, most of the resampling techniques are executed directly on the original datasets, which may affect the performance of each classifier due to the presence of redundant and noisy features, whereas MOEL implements a feature selection technique to generate multiple diverse and optimized subsets. Thus, the above two reasons help MOEL to obtain the best performance.

Table 3
Comparison of MOEL with Imbalanced learning algorithms in terms of F1_score (Best results are in bold fonts)

Dataset	**SMOTE**		**BSMOTE**		**ADASYN**		**ROS**		**MOEL**
	KNN	DT	KNN	DT	KNN	DT	KNN	DT	
German	66.41	61.72	68.26	65.38	65.15	60.01	66.23	53.33	**75.61**
Australian	79.99	73.07	82.75	71.69	75.86	71.69	78.57	75.00	**85.59**
Japanese	75.71	75.13	77.83	73.44	79.78	74.72	77.96	78.45	**85.23**
Taiwan	45.28	41.47	45.62	42.32	44.66	40.55	44.71	42.17	**55.21**
Polish1	45.78	45.26	48.29	44.03	**49.21**	42.39	49.41	44.38	44.21
Polish2	35.17	37.28	36.97	37.44	35.23	37.53	34.17	35.28	**48.29**
Number of wins	0/6	0/6	0/6	0/6	1/6	0/6	0/6	0/6	**5/6**

Table 4
Comparison of MOEL with Imbalanced learning algorithms in terms of G-mean (Best results are in bold fonts)

Dataset	SMOTE		BSMOTE		ADASYN		ROS		MOEL
	KNN	DT	KNN	DT	KNN	DT	KNN	DT	
German	76.69	72.68	78.90	76.21	75.90	71.83	76.69	66.27	**83.26**
Australian	83.57	77.66	86.06	76.57	80.01	76.57	82.40	78.67	**86.95**
Japanese	78.56	77.91	80.17	76.57	82.00	77.49	80.54	80.87	**85.27**
Taiwan	64.20	59.79	64.55	60.56	63.65	58.97	63.70	59.11	**67.28**
Polish1	58.27	60.15	59.14	61.27	59.72	62.33	60.14	61.43	**65.78**
Polish2	63.97	64.54	63.23	65.28	63.33	64.41	62.12	63.45	**68.29**
Number of wins	0/6	0/6	0/6	0/6	0/6	0/6	0/6	0/6	**6/6**

4.6. Comparison of proposed stacked-based ensemble method with benchmark ensemble methods

In the proposed stacked-based ensemble method five base classifiers were applied to each dataset and for each dataset, the classifiers that outperformed the others on balanced subsets were selected, which are illustrated in Table 5. The best-performed ensemble models of the datasets D_i are represented as $Eclf_i$. Then, the parameters of the selected base classifiers were optimized using a hyperparameter optimization framework. The efficacy of the proposed ensemble model (MOEL) is compared with a set of benchmark ensemble-based class imbalance learning methods, which include SMOTEBoost [44], RUSBoost [45], BalanceRF [50], BalanceCascade [51], and EasyEnsemble [52]. SMOTEBoost and RUSBoost integrate resampling with the AdaBoost algorithm. SMOTEBoost balances the class distributions by generating synthetic instances by interpolating between the existing minority instances, while RUSBoost balances the uneven distributions of instances of both classes by randomly eliminating samples from the majority class. BalancedRF implements random forest (RF) to balance the uneven class distributions. BalanceCascade and EasyEnsmble implement bagging as an ensemble-learning method and each bag gets trained using the AdaBoost classifier. Tables 6 and 7 present the comparison results of the proposed ensemble model with five benchmark ensemble-based imbalanced learning methods in terms of F1_score and G-mean, respectively. We observe that in most datasets (5 out of 6) our proposed method yields significantly better F1_score values, and in the case of G-mean, our method outperforms the other ensemble methods in all datasets. For example, MOEL obtains the F1_score of 75.61, 85.59, 85.23, 55.21, and 48.29 on German, Australian, Japanese, Taiwan, and Polish2 datasets, which are 5.27%, 1.95%, 2.9%, 2.52%, and 2.82% better than the second best performer, respectively. Table 7 shows that MOEL achieves the G-mean of 83.26, 86.95, 85.27, 65.78, and 61.29 on German, Australian, Japanese, Polish1, and Polish2 datasets, which are 6.78%, 3.04%, 1.8%, 2.57%, and 2.17% better than the second-best results, respectively. Comparative results demonstrate that the implementation of two-phase optimized techniques improves significantly the performance of the oversampling method.

Table 5
The best-performed ensembles on each dataset

Dataset	Ensemble model	LR	XGBoost	GBDT	RF	LGBM
German	Eclf1		✓	✓	✓	✓
Australian	Eclf2	✓	✓	✓		✓
Japanese	Eclf3	✓	✓	✓		
Taiwan	Eclf4		✓	✓	✓	
Polish1	Eclf5			✓	✓	✓
Polish2	Eclf6			✓	✓	✓

Table 6
Comparison of MOEL with Ensemble Imbalanced learning algorithms in terms of F1_score (Best results are in bold fonts)

Dataset	SMOTEBoost	RUSBoost	BalancedRF	BalanceCascade	EasyEnsemble	MOEL
German	62.63	62.33	68.93	68.35	70.37	**75.61**
Australian	74.07	83.64	82.09	76.49	75.05	**85.59**
Japanese	78.60	82.21	82.33	79.61	78.47	**85.23**
Taiwan	46.28	47.29	51.33	50.42	52.69	**55.21**
Polish1	41.28	42.33	**45.37**	45.27	43.28	44.21
Polish2	43.25	44.27	45.47	45.71	44.11	**48.29**
Number of wins	0/6	0/6	1/6	0/6	0/6	**5/6**

Table 7
Comparison of MOEL with Ensemble imbalanced learning algorithms in terms of G-mean (Best results are in bold fonts)

Dataset	SMOTEBoost	RUSBoost	BalancedRF	BalanceCascade	EasyEnsemble	MOEL
German	72.85	71.99	76.45	75.28	74.29	**83.26**
Australian	78.56	83.91	83.25	82.47	81.91	**86.95**
Japanese	80.59	82.61	83.47	82.61	82.28	**85.27**
Taiwan	64.47	65.63	**68.29**	67.33	67.21	67.28
Polish1	61.47	62.33	63.21	62.12	61.89	**65.78**
Polish2	63.29	64.24	66.12	65.47	65.12	**68.29**
Number of wins	0/6	0/6	1/6	0/6	0/6	**5/6**

4.7 Nonparametric Tests

To further highlight the efficacy of the proposed approach, we conduct pairwise comparison tests between MOEL and ensemble imbalanced learning approaches using a set of nonparametric statistical tests. Friedman test [12], [13]is a rank-based nonparametric test applied to detect the statistical performances of comparison methods. In the Friedman test, first, the ranks of each imbalance learning method are computed by using the average F1_score and G-mean on all the datasets. Next, the Friedman statistic value is computed as:

$$\Omega = \frac{12D}{K\left(K+1\right)}\left[\sum_{j=1}^{K} AR_j^2 - \frac{K(K+1)^2}{4}\right]$$

19

where $AR_j = \frac{1}{D}\sum_{i=1}^{D} r_{ij}$ (20)

In Eq. (19), D is the number of datasets, K is the number of imbalance learning methods, and r_{ij} is the rank of ith imbalance learning method on jth dataset. The distribution of Ω can be approximated by a Chi-squared $\left(\chi^2\right)$ distribution with K-1 degree of freedom. In all the Friedman tests, we set the significant level of α = 0.05. If the p-value of tests is less than α, then there exists a significant difference in performances between the competitors.

If the statistical difference among the methods is achieved, then three post-hoc tests will be conducted to compare MOEL (control algorithm) with ensemble imbalance learning methods. Results of the F1 score and G-mean are shown in Table 8–10 for a set of nonparametric statistical tests [55], including the Bonferroni-Dunn, Holm, and Hochberg tests. In terms of F1_score and G-mean, the tests show that the proposed method outperforms the ensemble-based oversampling methods. MOEL performs significantly better than other methods in terms of F1 score and G-mean with an average ranking of 1.5 and 1.333, respectively. Furthermore, the adjusted p-values show that MOEL performs better than other ensemble-based oversampling strategies. We get the conclusion that the MOEL is more suitable for high-dimensional imbalanced credit-scoring datasets from the nonparametric tests.

Table 8
Average Ranking of Ensemble imbalance learning algorithms

	F1_score	G-Mean
SMOTEBoost	5.667	5.833
RUSBoost	4.167	4.083
BalancedRF	2.500	2.000
BalanceCascade	3.333	3.250
EasyEnsemble	3.833	4.500
MOEL	**1.500**	**1.333**

Table 9
Adjusted p-values in nonparametric (Bonferroni-Dunn, Holm, and Hochberg) tests on F1_score

Algorithm	SMOTEBoost	RUSBoost	BalancedRF	BalanceCascade	EasyEnsemble
p_{Bonf}	1.43E-07	2.23E-08	1.79E-09	1.64E-06	2.47E-05
p_{Holm}	1.34E-07	8.36E-09	9.28E-08	3.17E-06	1.31E-05
p_{Hoch}	2.89E-07	9.25E-09	9.27E-09	3.48E-05	2.47E-05

Table 10
Adjusted p-values in nonparametric (Bonferroni-Dunn, Holm, and Hochberg) tests on G-mean

Algorithm	SMOTEBoost	RUSBoost	BalancedRF	BalanceCascade	EasyEnsemble
P_{Bonf}	1.53E-05	1.67E-10	1.21E-04	2.16E-16	3.47E-06
P_{Holm}	1.41E-05	2.75E-10	3.65E-04	2.35E-15	3.73E-06
P_{Hoch}	1.34E-05	2.52E-11	1.57E-03	2.23E-15	2.49E-06

5. Conclusions And Future Work

Credit scoring tool is used by banks and the financial sector to discover and differentiate borrowers with few credit history issues and to reduce credit risk. Due to the importance of credit scoring, financial decision-makers are currently looking for novel, cutting-edge machine learning algorithms to enhance performance and generate large revenues. In this paper, a novel multiple-optimized ensemble learning method called MOEL is proposed to manage high-dimensional imbalanced credit-scoring datasets. To build a reliable and optimized ensemble model, MOEL involves three main factors: generation of multiple diverse optimized subsets, selection of better-optimized subsets, and resampling of optimized subsets. In the first stage, WRFs are used to create more varied and optimal subsets, and the important features with more importance are sequentially added to the subset to improve the ensemble model's classification capabilities. Since the optimized subsets are derived from several WRFs, their generalization abilities may differ. In the next step, to obtain better-optimized subsets, each subset is evaluated using the weighted-density evaluation metric, a newly developed evaluation metric applied to generate more optimized subsets. Finally, to lessen the effects of imbalanced data on the classification, we employ an oversampling strategy on the optimized subsets to make them balanced, Experimental findings demonstrate that our method significantly outperforms the other widely used conventional imbalance learning techniques and ensemble learning techniques on various high-dimensional imbalanced credit scoring datasets.

In the future, a deep learning-based hybrid sampling technique can be used to build an effective credit scoring model. A conditional generative adversarial network can be applied which will provide more realistic minority

samples to address the imbalance issue. To further enhance the model, an ensemble learning framework can be proposed to train multiple base classifiers.

Declarations

Conflict of Interests The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. J. P. Barddal, L. Loezer, F. Enembreck, and R. Lanzuolo, “Lessons learned from data stream classification applied to credit scoring,” *Expert Syst Appl*, vol. 162, Dec. 2020, doi: 10.1016/j.eswa.2020.113899.
2. S. K. Trivedi, “A study on credit scoring modeling with different feature selection and machine learning approaches,” *Technol Soc*, vol. 63, Nov. 2020, doi: 10.1016/j.techsoc.2020.101413.
3. F. N. Koutanaei, H. Sajedi, and M. Khanbabaei, “A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring,” *Journal of Retailing and Consumer Services*, vol. 27, pp. 11–23, 2015, doi: 10.1016/j.jretconser.2015.07.003.
4. S. R. Lenka, S. K. Bisoy, R. Priyadarshini, and M. Sain, “Empirical Analysis of Ensemble Learning for Imbalanced Credit Scoring Datasets: A Systematic Review,” *Wireless Communications and Mobile Computing*, vol. 2022. Hindawi Limited, 2022. doi: 10.1155/2022/6584352.
5. S. Sadatrasoul, M. Gholamian, and K. Shahanaghi, “Combination of Feature Selection and Optimized Fuzzy Apriori Rules: The Case of Credit Scoring,” 2015.
6. J. Nalić, G. Martinović, and D. Žagar, “New hybrid data mining model for credit scoring based on feature selection algorithm and ensemble classifiers,” *Advanced Engineering Informatics*, vol. 45, Aug. 2020, doi: 10.1016/j.aei.2020.101130.
7. Y. Xu, Z. Yu, and C. L. P. Chen, “Classifier Ensemble Based on Multiview Optimization for High-Dimensional Imbalanced Data Classification,” *IEEE Trans Neural Netw Learn Syst*, 2022, doi: 10.1109/TNNLS.2022.3177695.
8. Y. Xia, C. Liu, B. Da, and F. Xie, *A novel heterogeneous ensemble credit scoring model based on bstacking approach*, vol. 93. Elsevier Ltd, 2018. doi: 10.1016/j.eswa.2017.10.022.
9. S. F. Crone and S. Finlay, “Instance sampling in credit scoring: An empirical study of sample size and balancing,” *Int J Forecast*, vol. 28, no. 1, pp. 224–238, 2012, doi: 10.1016/j.ijforecast.2011.07.006.
10. Q. Kang, X. S. Chen, S. S. Li, and M. C. Zhou, “A Noise-Filtered Under-Sampling Scheme for Imbalanced Classification,” *IEEE Trans Cybern*, vol. 47, no. 12, pp. 4263–4274, Dec. 2017, doi: 10.1109/TCYB.2016.2606104.
11. Z. Chen, J. Duan, L. Kang, and G. Qiu, “A hybrid data-level ensemble to enable learning from highly imbalanced dataset,” *Inf Sci (N Y)*, vol. 554, pp. 157–176, Apr. 2021, doi: 10.1016/j.ins.2020.12.023.
12. P. Soltanzadeh and M. Hashemzadeh, “RCSMOTE: Range-Controlled synthetic minority over-sampling technique for handling the class imbalance problem,” *Inf Sci (N Y)*, vol. 542, pp. 92–111, Jan. 2021, doi: 10.1016/j.ins.2020.07.014.

13. S. Wang and X. Yao, “Relationships between diversity of classification ensembles and single-class performance measures,” *IEEE Trans Knowl Data Eng*, vol. 25, no. 1, pp. 206–219, 2013, doi: 10.1109/TKDE.2011.207.
14. Y. Xu, Z. Yu, C. L. P. Chen, and W. Cao, “A Novel Classifier Ensemble Method Based on Subspace Enhancement for High-Dimensional Data Classification,” *IEEE Trans Knowl Data Eng*, 2021, doi: 10.1109/TKDE.2021.3087517.
15. Y. Xu, Z. Yu, W. Cao, C. L. P. Chen, and J. You, “Adaptive Classifier Ensemble Method Based on Spatial Perception for High-Dimensional Data Classification,” *IEEE Trans Knowl Data Eng*, vol. 33, no. 7, pp. 2847–2862, Jul. 2021, doi: 10.1109/TKDE.2019.2961076.
16. Z. Yu *et al.*, “Adaptive Semi-Supervised Classifier Ensemble for High Dimensional Data Classification,” *IEEE Trans Cybern*, vol. 49, no. 2, pp. 366–379, Oct. 2017, doi: 10.1109/tcyb.2017.2761908.
17. J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso, “Rotation Forest: A New Classifier Ensemble Method.”
18. S. Asadi and S. E. Roshan, “A bi-objective optimization method to produce a near-optimal number of classifiers and increase diversity in Bagging,” *Knowl Based Syst*, vol. 213, Feb. 2021, doi: 10.1016/j.knosys.2020.106656.
19. Y. Song, Y. Wang, X. Ye, D. Wang, Y. Yin, and Y. Wang, “Multi-view ensemble learning based on distance-to-model and adaptive clustering for imbalanced credit risk assessment in P2P lending,” *Inf Sci (N Y)*, vol. 525, pp. 182–204, 2020, doi: 10.1016/j.ins.2020.03.027.
20. D. Liang, C. F. Tsai, and H. T. Wu, “The effect of feature selection on financial distress prediction,” *Knowl Based Syst*, vol. 73, no. 1, pp. 289–297, 2015, doi: 10.1016/j.knosys.2014.10.010.
21. D. J. Hand and W. E. Henley, “Statistical classification methods in consumer credit scoring: A review,” *J R Stat Soc Ser A Stat Soc*, vol. 160, no. 3, pp. 523–541, 1997, doi: 10.1111/j.1467-985X.1997.00078.x.
22. R. Emekter, Y. Tu, B. Jirasakuldech, and M. Lu, “Evaluating credit risk and loan performance in online Peer-to-Peer (P2P) lending,” *Appl Econ*, vol. 47, no. 1, pp. 54–70, 2015, doi: 10.1080/00036846.2014.962222.
23. L. Yu, R. Zhou, L. Tang, and R. Chen, “A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data,” *Applied Soft Computing Journal*, vol. 69, pp. 192–202, 2018, doi: 10.1016/j.asoc.2018.04.049.
24. E. Angelini, G. di Tollo, and A. Roli, “A Neural Network Approach for Credit Risk Evaluation.”
25. C. Serrano-Cinca and B. Gutiérrez-Nieto, “The use of profit scoring as an alternative to credit scoring systems in peer-to-peer (P2P) lending,” *Decis Support Syst*, vol. 89, pp. 113–122, Sep. 2016, doi: 10.1016/j.dss.2016.06.014.
26. X. Yao, J. Crook, and G. Andreeva, “Support vector regression for loss given default modelling,” *Eur J Oper Res*, vol. 240, no. 2, pp. 528–538, 2015, doi: 10.1016/j.ejor.2014.06.043.
27. T. M. Khoshgoftaar, J. van Hulse, and A. Napolitano, “Comparing boosting and bagging techniques with noisy and imbalanced data,” *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 41, no. 3, pp. 552–568, May 2011, doi: 10.1109/TSMCA.2010.2084081.
28. J. F. Díez-Pastor, J. J. Rodríguez, C. I. García-Osorio, and L. I. Kuncheva, “Diversity techniques improve the performance of the best imbalance learning ensembles,” *Inf Sci (N Y)*, vol. 325, pp. 98–117, Dec. 2015, doi: 10.1016/j.ins.2015.07.025.

29. H. Faris *et al.*, “Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: a case from the Spanish market,” *Progress in Artificial Intelligence*, vol. 9, no. 1, pp. 31–53, Mar. 2020, doi: 10.1007/s13748-019-00197-9.
30. G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, “A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data.”
31. K. Mccarthy, B. Zabar, and G. Weiss, “Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes?,” 2005.
32. S. R. Lenka, S. K. Bisoy, R. Priyadarshini, and B. Nayak, “Representative-Based Cluster Undersampling Technique for Imbalanced Credit Scoring Datasets,” 2022, pp. 119–129. doi: 10.1007/978-981-19-0475-2_11.
33. W. C. Lin, C. F. Tsai, Y. H. Hu, and J. S. Jhang, “Clustering-based undersampling in class-imbalanced data,” *Inf Sci (N Y)*, vol. 409–410, pp. 17–26, 2017, doi: 10.1016/j.ins.2017.05.008.
34. R. C. Prati, G. E. A. P. A. Batista, and M. C. Monard, “Class imbalances versus class overlapping: An analysis of a learning system behavior,” in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2004, vol. 2972, pp. 312–321. doi: 10.1007/978-3-540-24694-7_32.
35. N. v. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
36. H. Han, W. Y. Wang, and B. H. Mao, “Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning,” *Lecture Notes in Computer Science*, vol. 3644, no. PART I, pp. 878–887, 2005, doi: 10.1007/11538059_91.
37. H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” *Proceedings of the International Joint Conference on Neural Networks*, no. 3, pp. 1322–1328, 2008, doi: 10.1109/IJCNN.2008.4633969.
38. S. Barua, M. M. Islam, X. Yao, and K. Murase, “MWMOTE - Majority weighted minority oversampling technique for imbalanced data set learning,” *IEEE Trans Knowl Data Eng*, vol. 26, no. 2, pp. 405–425, 2014, doi: 10.1109/TKDE.2012.232.
39. G. Douzas, F. Bacao, and F. Last, “Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE,” *Inf Sci (N Y)*, vol. 465, pp. 1–20, 2018, doi: 10.1016/j.ins.2018.06.056.
40. L. Peng, H. Zhang, B. Yang, and Y. Chen, “A new approach for imbalanced data classification based on data gravitation,” *Inf Sci (N Y)*, vol. 288, no. 1, pp. 347–373, 2014, doi: 10.1016/j.ins.2014.04.046.
41. S. Y. Kim and A. Upneja, “Predicting restaurant financial distress using decision tree and AdaBoosted decision tree models,” *Econ Model*, vol. 36, pp. 354–362, 2014, doi: 10.1016/j.econmod.2013.10.005.
42. D. Mease, A. J. Wyner, and A. Buja, “Boosted Classification Trees and Class Probability/Quantile Estimation,” 2007.
43. M. v Joshi, V. Kumar, and R. C. Agarwal, “Evaluating Boosting Algorithms to Classify Rare Classes: Comparison and Improvements.”
44. N. v Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, “LNAI 2838 - SMOTEBoost: Improving Prediction of the Minority Class in Boosting,” 2003.

45. C. Seiffert, T. M. Khoshgoftaar, J. van Hulse, and A. Napolitano, “RUSBoost: A hybrid approach to alleviating class imbalance,” *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010, doi: 10.1109/TSMCA.2009.2029559.
46. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 4, pp. 463–484, 2012, doi: 10.1109/TSMCC.2011.2161285.
47. B. H. Menze *et al.*, “A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data,” *BMC Bioinformatics*, vol. 10, Jul. 2009, doi: 10.1186/1471-2105-10-213.
48. B. Komer, J. Bergstra, and C. Eliasmith, “Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn,” 2014.
49. M. Zięba, S. K. Tomczak, and J. M. Tomczak, “Ensemble Boosted Trees with Synthetic Features Generation in Application to Bankruptcy Prediction.”
50. C. Chen and A. Liaw, “Using Random Forest to Learn Imbalanced Data.”
51. X. Y. Liu, J. Wu, and Z. H. Zhou, “Exploratory undersampling for class-imbalance learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 539–550, 2009, doi: 10.1109/TSMCB.2008.2007853.
52. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 4. pp. 463–484, Jul. 2012. doi: 10.1109/TSMCC.2011.2161285.
53. M. Friedman, “The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance,” *J Am Stat Assoc*, vol. 32, no. 200, pp. 675–701, 1937, doi: 10.1080/01621459.1937.10503522.
54. M. Friedman, “A Comparison of Alternative Tests of Significance for the Problem of m Rankings,” 1940. [Online]. Available: http://www.jstor.orgURL:http://www.jstor.org/stable/2235971
55. R. Eisinga, T. Heskes, B. Pelzer, and M. te Grotenhuis, “Exact p-values for pairwise comparison of Friedman rank sums, with application to comparing classifiers,” *BMC Bioinformatics*, vol. 18, no. 1, Jan. 2017, doi: 10.1186/s12859-017-1486-2.
56. 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.

Figures

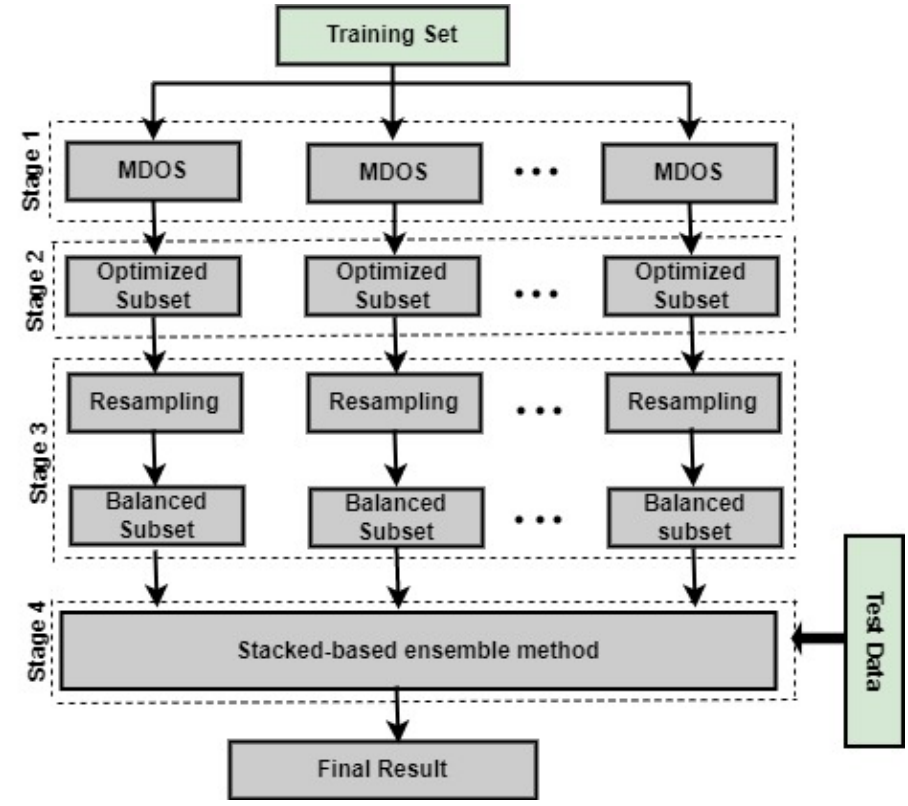

Figure 1

Framework of Multiple Optimized Ensemble Model

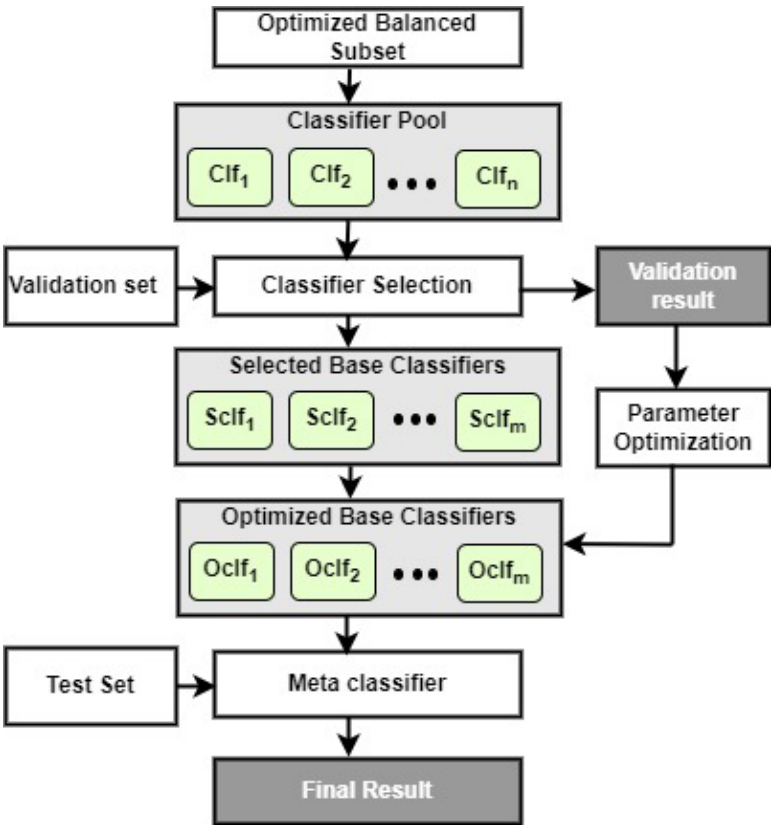

Figure 2

Flow diagram of stacking-based ensemble method

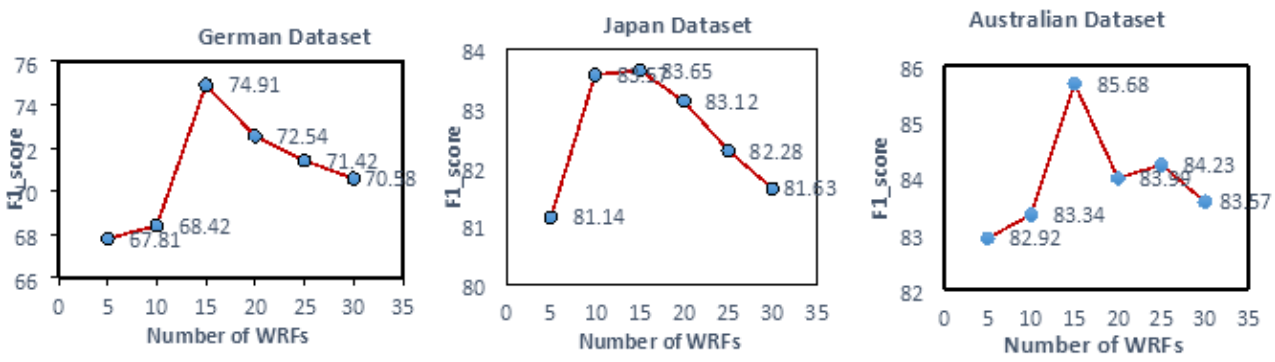

Figure 3

The performance of MOEL on different numbers of WRFs

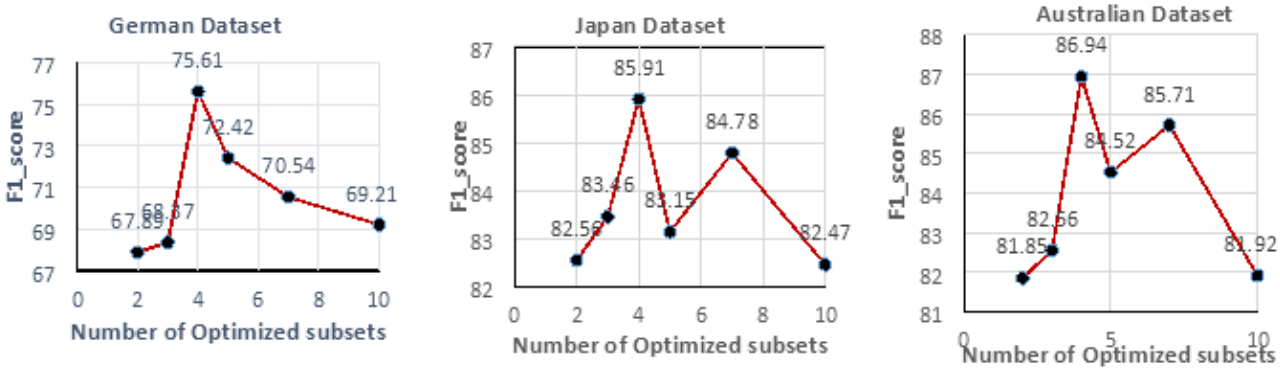

Figure 4

The performance of MOEL on different numbers of optimized subsets