HBSBoost: A Hybrid Balancing Technique for Defaulting Enterprise Recognition

Marui DU*, Zuoquan ZHANG

School of Mathematics and Statistics, Beijing Jiaotong University, China
17118446@bjtu.edu.cn (**Corresponding author*), Zuoquanzhang@163.com

Abstract: The identification of defaulting enterprises and the detection of abnormal behavior in the financial field are being faced with the problem of serious imbalance in the proportions of data samples, but numerous machine learning classification models are based on the assumption that the proportions of data samples are relatively close. Therefore, when faced with data imbalances, classification models often have low recognition rates for the minority classes and fail to achieve the desired effect of data classification. With the purpose of solving this problem, this paper proposes a data balancing technique based on a hybrid sampling technique and a boosting algorithm. This model uses a hybrid sampling technique to construct the balanced dataset. The boosting algorithm is then employed in order to improve the discriminative power of the machine learning algorithm with regard to the information on the minority class. The proposed method outperforms the random under-sampling, SMOTE, hybrid sampling, SMOTEBoost, and RUSBoost algorithms for seven real-world datasets.

Keywords: Credit risk, Default recognition, Imbalance classification, Machine learning.

1. Introduction

When measuring the credit risk of an enterprise, enterprises can be typically classified into two categories: good creditors and bad creditors. Those who can pay their debts on time are classified as good creditors, while those who cannot or will not pay their debts are classified as bad creditors. Such enterprises are prone to default (Popescu et al., 2019). In general, the majority of enterprises will be able to make timely repayments, while a minority will default, which will lead to an imbalance due to the fact that the number of non-defaulting enterprises in the database is much higher than the number of defaulting enterprises (Ciampi et al., 2020).

The phenomenon of class imbalance will have an impact on the accurate measurement of enterprise credit risk. First, measuring the credit risk of an enterprise often requires a large amount of information, where errors and spurious information are inevitably present, resulting in a large amount of noise in the data source (Ouadine et al., 2020). In the case of class imbalance, the interference of noise will make it more difficult to find minority samples, and the classifier will have a hard time identifying minority classes. Second, many classifiers suffer from classification bias when classifying imbalanced data (Dastile et al., 2020). For example, class boundary information plays a key role in support vector machine (SVM) model. Whether the boundary of the sample is balanced or not is the essence of whether the data is balanced or not (Yee et al., 2022). In general, the true boundary of a sample cannot be identified. When the boundaries are not balanced, minority class samples are prone to misclassification. Decision Tree (DT) model splits the original problem into several subproblems. On the one hand, the classifier needs to find sample data rules in the incomplete subspace (Lefa et al., 2022). On the other hand, the data rules learned by the classifier across space are not easily mined and may be lost, forming data fragments. Such a data splitting based on the divide-and-conquer idea would lead to a very good performance of the classifier in learning.

To solve the class imbalance problem, two categories of techniques have been put forward: sampling approaches (García et al., 2012) and algorithm-based approaches (Wang et al., 2020). Sampling approaches generally aim at balancing the class distribution by either under-sampling the majority or over-sampling the minority class. The most commonly used under-sampling approach consists in randomly removing samples from the majority class until two classes are approximately equally represented (Kinoshita et al., 2020), which may result in a loss of some useful information. Similarly, over-sampling by randomly duplicating samples of the minority class tends to induce overfitting and decrease the classifier's generalization because of its repeated extraction. Therefore, an intelligent over-sampling method named Synthetic Minority Over-sampling Technique with AdaBoost (SMOTE) (Chawla et al., 2002) is proposed. The SMOTE technique generates new minority samples in the nearest neighbors of every minority class sample rather than simply copy them. Algorithm-

https://doi.org/10.24846/v31i4y202207

based approaches aim at improving a classifier's performance based on its inherent characteristics (Yang et al., 2017). Among them, the ensemble learning approach is strongly recommended.

A series of classification models are applied for imbalanced data. Firstly, García, Marqués & Sánchez (2012) investigated the influence of both the imbalance ratio and the classifier on the performance of sampling approaches and came to the final conclusion that over-sampling of the minority class consistently outperforms under-sampling of the majority class. Subsequently, a hybrid sampling technique (Gao et al., 2020) combining the two sampling techniques above was proposed. It was proved that this method always outperforms the individual sampling techniques as it involves the strengths of the individual techniques while lessening their drawbacks. What's more, an integrated method SMOTEBoost was presented by Chawla et al. (2003), which combines the SMOTE algorithm and an ensemble method named boosting to improve prediction on the minority class. Inspired by this, Seiffert et al. (2010) made a further improvement again, and immediately put forward a hybrid sampling boosting algorithm called RUSBoost by combining boosting with over-sampling. In comparison with SMOTEBoost, RUSBoost has been proved to perform equally well or better than the former.

This sampling technique suffers from overfitting, and the Boosting algorithm fails to solve the sample class distribution problem, thus this paper proposes a ensemble balancing technique based on a hybrid sampling technique and the Boosting algorithm (HBSBoost). The proposed method can effectively improve the classification ability of the classifier.

The contributions of this paper are as follows:

1. A new balancing technique HBSBoost is proposed, which balances datasets by hybrid sampling techniques and improves the discriminative power of classifiers for minority samples by Boosting algorithm.

2. The proposed technique is evaluated for seven real-world datasets. Experiment results show that in comparison with under-sampling technique, over-sampling technique, hybrid sampling technique, RUSBoost and SMOTEBoost techniques, the classification accuracy of machine learning models has been significantly improved by using HBSBoost technique.

The structure of this paper is as follows. Section 2 introduces the techniques used in the proposed method, including under-sampling technique, over-sampling technique and Boosting algorithm. Section 3 sets forth the hybrid sampling technique and the HBSBoost technique proposed in this paper. Section 4 presents the experiments carried out and the analysis of the obtained results. Section 5 includes the conclusion of this paper.

2. Research Methodology

2.1 Under-sampling

The simplest under-sampling method is random under-sampling (Tahir et al., 2012) by randomly removing samples from the majority class until the remaining number of samples from the majority class is nearly equal to the number of samples from the minority class. It is obvious that there will be a loss of some important information accompanied with deleting samples from the training data. Even so, random under-sampling can still effectively balance the training data, and eventually help these classifiers better recognize the minority class, in spite of the fact that standard classification algorithms usually have a bias towards to the majority class. It has been proved to be one of the most effective sampling approaches. Tomek links and condensed nearest neighbor rule (Gowda & Krishna, 1979) are other under-sampling techniques.

2.2 Over-sampling

Over-sampling technique aims to increase the number of samples from the minority class to balance the training data. There are two kinds of classic over-sampling methods. One is random over-sampling (Johnson & Khoshgoftaar, 2019) which consists in simply replicating the minority class samples until class balance. For this method, some samples in the original minority class may constantly reappear in the balanced training set, which will likely cause overfitting in case no new information is added. Therefore, another method named SMOTE (Chawla et al., 2002) was proposed. It generates new artificial interpolated samples within the k nearest neighbors of samples in the minority class. SMOTE not only increases the number of minority class samples to balance the training data but also to effectively prevent overfitting.

For a training sample S with n minority class samples, suppose the over-sampling rate is N, number of nearest neighbors is k. The following steps are involved to obtain the balanced dataset $S_{balance}$ when using the SMOTE technique:

```
Algorithm 1 SMOTE
Input: Training sample $S$, over sampling rate $N$, number of nearest neighbors $k$, number of minority class samples $n$
Output: SMOTE algorithm
1. Calculate the $k$-nearest neighbors of minority class samples $x_i$.
2. Randomly select $x_a$ samples from the $k$-nearest neighbors of minority class samples $x_i$.
3. Synthetic samples $f_{iq}$ between $x_i$ and $x_a$ use:
     $f_{iq} = x_i + (x_{aq} - x_{iq}) \times rand(0,1)$     (1)
     where $rand(0,1)$ is a random number between 0 and 1.
4. Repeat the above steps for $N$ times.
5. Obtain the new training samples $S_{balance}$.
```

2.3 Boosting

The basic idea of ensemble learning is to combine multiple weak classifiers into one strong classifier (Shahraki et al., 2020). Several scholars have demonstrated the significance of ensemble learning to improving the performance of classifiers from statistical, computational, and theoretical perspectives (Goyal & Vajpayee, 2017). The boosting algorithm is one of the commonly used ensemble learning algorithms. The basic idea is to enforce learning on samples that are prone to misclassification. The steps of Boosting are as follows: (1) assign the same weight to each training sample; (2) learn a training set T by using basic classifier, comparing the real label with the label predicted by the classifier, and improving the weight of the misclassified sample; (3) retraining by using the training set with the adjusted weight; (4) the process is repeated until the iteration is finished; (5) take the linear combination of all basic classifiers as the final strong classifier.

The Boosting algorithm aims to combine multiple weak learners into one strong learner with arbitrarily high accuracy. The basic idea is that when building a classifier, it is hoped that the classifier will pay more attention to the samples with classification errors in the previous round, since some classifiers may have better classification performance for minority samples and some classifiers may have better performance for majority samples, therefore, a Boosting-based complementary classifier system combining multiple classifiers was proposed..

AdaBoost is the most representative algorithm in the Boosting family. The AdaBoost is introduced in Algorithm 2:

```
Algorithm 2 AdaBoost
Require: Training set
  $T = (x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$, where
  $x_i \in \mathcal{X} = \mathbf{R}^n$, $y_i \in \mathcal{Y} = \{-1, 1\}, i = 1, 2, ..., N$
Ensure: The final classifier $G(x)$
1: Initialize the weight distribution of training samples:
     $D_1 = (w_{11}, ..., w_{1i}, w_{1N})$, $w_{1i} = \frac{1}{N}, i = 1, 2, ..., N$
2: for $m = 1, 2, ..., M$ do
3:     (a) The training dataset with weight distribution $D_m$ is used for learning to obtain a classifier:
         $G_n(x) \in \{-1, +1\}$
4:     (b) Calculate the classification error of $G_m(x)$ on the training dataset:
         $e_m = P(G_m(x_i) \neq y_i)$
         $= \sum_{i=1}^{N} w_{mi} I(G_m(x_i) \neq y_i)$
5:     (c) Compute the weight of $G_m(x)$:
         $\alpha_m = \frac{1}{2} log \frac{1 - e_m}{e_m}$
6:     (d) Update weight distribution $D_m$ of the training dataset:
         $D_{m+1} = (w_{m+1,1}, ..., w_{m+1,i}, ..., w_{m+1,N})$
         $w_{m+1,i} = \frac{w_{mi}}{Z_m} exp(-\alpha_m y_i G_m(x_i)),$
         $i = 1, 2, ..., N$
         where $Z$ is the normalization factor:
         $Z_m = \sum_{i=1}^{N} w_{mi} exp(-\alpha_m y_i G_m(x_i))$
7:     Obtain the final classifier:
         $f_x = \sum_{m=1}^{M} \alpha_m G_m(x)$
         $G_x = sign(f(x))$
         $= sign(\sum_{m=1}^{M} \alpha_m G_m(x))$
8: end for
```

3. Proposed Models

3.1 Hybrid Sampling

There are two types of methods to solve the data imbalance problem at the data level: under-sampling and over-sampling. Using only under-sampling results in a loss of information about the sample data, while using only over-sampling results in an over-fitting of the underlying classifier training. Hybrid sampling can effectively combine the strengths and weaknesses of the two sampling methods to achieve data class balance. In hybrid sampling, SMOTE over-sampling and random under-sampling techniques are commonly used to adjust the sample distribution. The process of hybrid sampling is introduced in Algorithm 3:

```
Algorithm 3 Hybrid Sampling

Require: Majority class sample $S_{majority}$, minority class
sample $S_{minority}$, number of majority class samples
$N_{majority}$, number of minority class samples $N_{minority}$,
number of synthesized majority class samples $\alpha$,
number of synthesized minority class samples $\beta$.
Ensure: Balanced dataset $S_{balance}$
1: The SMOTE algorithm is used to increase the number
     of minority samples, obtain the new minority
     samples:
     $S^*_{minority} = S_{minority} \cup SMOTE(\beta, N_{minority})$   (2)
2: The random under-sampling is used to reduce the
     number of majority samples, obtain the new majority
     samples:
     $S^*_{majority} = S_{majority} \setminus RUS(\alpha, N_{majority})$   (3)
3: Obtain the balanced dataset:
     $S_{balance} = S^*_{minority} \cup S^*_{majority}$   (4)
```

3.2 The HBSBoost technique

In the existing research, there is the method which combines over-sampling with ensemble algorithm Boosting (RUSBoost) and the method which combines under-sampling with ensemble algorithm Boosting (SMOTEBoost). In this section, the HBSBoost algorithm combining hybrid sampling and Boosting is presented. The proposed algorithm is based on the following aspects. First, hybrid-sampling technique has been shown to significantly outperform single over-sampling and under-sampling technique in solving class imbalance problems. Secondly, over-sampling and under-sampling technique have been successfully embedded in ensemble algorithm Boosting. Finally, the ensemble algorithm based on data sampling is more effective than the use of sampling technique alone. Therefore, guided by previous works, this paper aims to present a combination of the hybrid-sampling and Boosting algorithms to achieve higher classification performance and improve the recognition rate for minority samples. The procedure of the HBSBoost algorithm is divided into four parts. In the first part the hybrid-sampling technique (algorithm 3) is employed to obtain a balanced sample set; in the second part a classifier is trained under the balanced dataset. In the third part, the sample weights and classifier weights are changed based on prediction results, and iteration is performed to obtain a strong classifier. Finally, the predicted labels for the test set are outputted. The flow chart and the pseudo-code of the proposed algorithm are shown in Figure 1 and Algorithm 4, respectively.

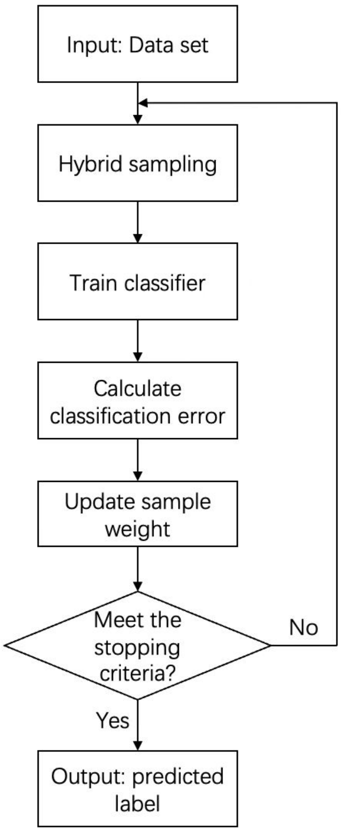

Figure 1. HBSBoost flow chart

```
**Algorithm 4** HBSBoost
**Require**: The majority training set $S_{majority} = (x_1, y_1), ..., (x_n, y_n)$, the minority training set $S_{minority} = (x_{n+1}, y_{n+1}), ..., (x_{n+m}, y_{n+m})$, $y_i \in \{1, -1\}$; the basic classifier $L$; number of iteration T.
**Ensure**: $T(x_i) = sign(\sum_{t=1}^{T} (Z_{it} \times a(t)))$
Initialize the weight distribution
    $H = ones(n+m, T+1) / n+m$, $D = zeros(n+m, T)$.
    $H$ stores the weight of the original training set, $D$ stores the weights of the balanced training set.
**for** $t = 1, 2, ..., T$ **do**
    (a) Balance the class distribution and get the new training set use algorithm 3.
    (b) Train a weak classifier $L_t$ with the balanced training set $S_{balance}$.
    (c) Get the predicted class label including the new training data and the test data
        $Z_t = L_t(S_{balance})$, $T_t = L_t(T)$
    (e) Calculate the error of $L_t$:
        $\varepsilon(t) = \sum_{Z_{it \neq y_i}} D_{i,t}$
    (f) Weight each selected sample in the new training set:
    **for** $i = 1, ..., \left\lfloor \frac{1}{2} \times (n+m) \right\rfloor$ **do**
        $D(i,t) = H(i,t)$ % i is the sample index of a bootstrap sample through under-sampling
    **end for**
    **for** $i = \left\lfloor \frac{1}{2} \times (n+m) \right\rfloor + 1, ..., n+m$ **do**
        $D(i,t) = H(i + \left\lceil \frac{1}{2} \times (n-m) \right\rceil, t)$
    **end for**
    (g) Calculate the weight of each weak classifier $L_i$: $a_t = \frac{1}{2} \times \ln \frac{1-\varepsilon(t)}{\varepsilon(t)}$
    (h) Update the weight of these misclassified samples in the new training set:
        $H(i, t+1) = H(i,t) \times exp(a(t) \times Z_{it} \times y_i)$
    **for** $i = 1, ..., \left\lfloor \frac{1}{2} \times (n+m) \right\rfloor$ **do**
        $H(i, t+1) = D(i,t)$
    **end for**
    **for** $i = \left\lfloor \frac{1}{2} \times (n+m) \right\rfloor + 1, ..., n+m$ **do**
        $H(i + \left\lceil \frac{1}{2} \times (n-m) \right\rceil, t+1) = D(i,t)$
    **end for**
**end for**
```

4. Experiment and Results

4.1 Dataset Information

In this paper, seven real datasets are used. Table 1 shows the information for all the datasets.

Table 1. Dataset information

Dataset	Sample Size	Imbalance Ratio	Features
GEM	4960	4.82	50
STAR	1792	10.13	50
SMB	11103	3.12	50
German	1000	2.33	20
Australia	690	0.8	14
Polish	240	1.14	30
Japanese	653	0.82	15

The GEM dataset belongs to the Shenzhen Stock Exchange, and it includes 528 small and medium-sized enterprises (SMEs), the STAR dataset belongs to the Shanghai Stock Exchange, and it includes 297 SMEs and the SMB dataset belongs to the Shenzhen Stock Exchange, and it contains 722 SMEs. These three datasets were downloaded from the CSMAR database (Du et al., 2022). The features used in GEM, STAR, and SMB datasets are the 50 most effective CountSim MP features proposed in (Du et al., 2022). The other four datasets are from the University of California Irvine (UCI) Machine Learning Repository (Tanveer et al., 2019). These datasets contain different sample sizes, different features, and different imbalance ratios, which can represent different default problems.

4.2 Data Pre-processing

Variables with more than 50% missing data were removed, and then the data was filled using mean values. In order to eliminate the adverse effects caused by sample data, the data was normalized and processed as a decimal between (0, 1). The normalization formula is shown in equation (5):

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5}$$

4.3 Evaluation indices of model performance

Accuracy is the most commonly used measure for verifying the performance of machine learning classification. However, for imbalanced data classification, it is necessary to pay attention to the classification of minority classes. Therefore, Area Under Receiver Operating Characteristic Curve (AUC), F-measure and geometric-mean (G-mean) were chosen as the evaluation metrics in this paper. The confusion matrix (in Table 2) consists of True Positive (TP), False Negative (FN), False

Positive (FP), and True Negative (TN) values and is used for calculating the evaluation indices.

Table 2. Confusion matrix

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

For defaulting enterprise recognition, TP is the number of enterprises which are correctly classified as defaulting, FN is the number of defaulting enterprises which are incorrectly classified as non-defaulting, FP is the number of non-defaulting enterprises which are incorrectly classified as defaulting, and TN is the number of enterprises which are correctly classified as non-defaults. From the confusion matrix, the formula of the AUC (Tomczak & Zięba, 2015), F-measure (F) and G-mean (G) can be derived as they are expressed in equations (6), (9) and (10), respectively.

$$AUC = \frac{1}{2}(1 + \frac{TP}{TP+FN} - \frac{FP}{FP+TN}) \tag{6}$$

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

$$F = \frac{1 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

$$G = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} \tag{10}$$

4.4 Result Analysis

To verify the classification advantages of the HBSBoost technique, a detailed comparative analysis was carried out. Five balancing techniques were selected for comparison purposes, namely:

1. Random under-sampling (RUS): by randomly reducing majority class samples to achieve balanced data.
2. Synthetic Minority Over-sampling Technique (SMOTE): through some strategies for synthesizing minority samples to achieve balanced data.
3. Hybrid sampling (HBS): use both random under-sampling and SMOTE to achieve balanced data.
4. Synthetic Minority Over-sampling Technique with AdaBoost (SMOTEBoost): an ensemble learning algorithm that combines SMOTE and Boosting algorithm.
5. Random under-sampling with AdaBoost (RUSBoost): an ensemble learning algorithm that combines random under-sampling and Boosting algorithm.

The data balance ratio is set to 1:1. The performance of all the comparison techniques is tested on support vector machine (SVM), artificial neural network (ANN) and decision tree (DT) models. To more accurately compare the effect of the different machine learning models, during the experiments, a 10-fold-cross-validation method was used where the dataset was divided into ten parts, nine of which were used for training and one for testing. Each experiment was performed 20 times and the average value was taken as the final result. All the experiments were implemented in Python 2.7.17 on Win 8.1+ with CPU i5-9300 processor and 8G+RAM. The imbalanced-learn package of Python was used for implementing the imbalance methods. The sklearn package of Python was used for implementing the SVM, ANN and DT models.

Tables 3 to 11 show the values of AUC, F-measure and G-mean for each balancing technique for the three classifiers mentioned above for the seven datasets employed. The title of the second

Table 3. AUC score comparison for SVM

Dataset	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
GEM	0.708	0.722	0.741	0.748	0.746	0.743	0.752
STAR	0.715	0.721	0.721	0.726	0.732	0.764	0.787
SMB	0.713	0.72	0.725	0.736	0.758	0.779	0.795
German	0.705	0.727	0.729	0.733	0.732	0.729	0.745
Australia	0.655	0.675	0.736	0.738	0.762	0.773	0.824
Polish	0.801	0.815	0.816	0.816	0.821	0.822	0.836
Japanese	0.5	0.548	0.58	0.63	0.667	0.669	0.705

Table 4. AUC score comparison for ANN

Dataset	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
GEM	0.694	0.708	0.696	0.703	0.738	0.712	0.727
STAR	0.703	0.712	0.713	0.72	0.72	0.728	0.741
SMB	0.718	0.724	0.728	0.731	0.731	0.736	0.742
German	0.687	0.701	0.693	0.698	0.728	0.723	0.731
Australia	0.627	0.629	0.637	0.649	0.663	0.667	0.681
Polish	0.532	0.567	0.641	0.653	0.644	0.73	0.759
Japanese	0.687	0.712	0.744	0.747	0.712	0.827	0.86

Table 5. AUC score comparison for DT

Dataset	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
GEM	0.637	0.646	0.662	0.665	0.67	0.674	0.707
STAR	0.65	0.633	0.675	0.681	0.676	0.679	0.696
SMB	0.7	0.711	0.719	0.72	0.725	0.728	0.731
German	0.646	0.653	0.667	0.671	0.709	0.694	0.721
Australia	0.667	0.687	0.691	0.701	0.708	0.711	0.715
Polish	0.536	0.564	0.57	0.562	637	0.654	0.692
Japanese	0.56	0.597	0.597	0.599	0.637	0.642	0.656

Table 6. F-measure comparison for SVM

Dataset	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
GEM	0.619	0.637	0.638	0.651	0.649	0.658	0.665
STAR	0.583	0.602	0.613	0.657	0.688	0.781	0.737
SMB	0.703	0.707	0.721	0.746	0.817	0.821	0.853
German	0.37	0.486	0.527	0.547	0.546	0.628	0.642
Australia	0.427	0.438	0.476	0.478	0.537	0.525	0.648
Polish	0.62	0.763	0.789	0.805	0.806	0.837	0.841
Japanese	0.422	0.452	0.467	0.479	0.548	0.558	0.575

Table 7. F-measure comparison for ANN

Dataset	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
GEM	0.426	0.439	0.441	0.453	0.453	0.456	0.473
STAR	0.431	0.465	0.472	0.482	0.493	0.552	0.562
SMB	0.607	0.678	0.682	0.693	0.717	0.725	0.752
German	0.369	0.427	0.474	0.477	0.519	0.526	0.537
Australia	0.422	0.436	0.449	0.457	0.456	0.465	0.477
Polish	0.068	0.073	0.097	0.099	0.101	0.122	0.203
Japanese	0.099	0.148	0.152	0.163	0.167	0.182	0.188

Table 8. F-measure comparison for DT

Dataset	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
GEM	0.323	0.338	0.352	0.363	0.361	0.367	0.382
STAR	0.333	0.34	0.346	0.36	0.372	0.4	0.42
SMB	0.43	0.447	0.476	0.501	0.522	0.536	0.569
German	0.375	0.413	0.434	0.454	0.508	0.523	0.524
Australia	0.51	0.553	0.562	0.56	0.583	0.591	0.64
Polish	0.083	0.12	0.13	0.175	0.228	0.256	0.287
Japanese	0.27	0.374	0.451	0.467	0.482	0.476	0.493

Table 9. G-mean comparison for SVM

Dataset	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
GEM	0.686	0.698	0.713	0.719	0.73	0.734	0.739
STAR	0.843	0.995	0.957	0.961	0.963	0.962	0.976
SMB	0.92	0.948	0.951	0.975	0.976	0.983	0.991
German	0.701	0.73	0.732	0.747	0.762	0.787	0.826
Australia	0.68	0.684	0.72	0.796	0.795	0.836	0.857
Polish	0.603	0.621	0.63	0.632	0.654	0.703	0.701
Japanese	0.472	0.546	0.568	0.626	0.627	0.654	0.661

Table 10. G-mean comparison for ANN

Dataset	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
GEM	0.605	0.62	0.627	0.636	0.646	0.647	0.661
STAR	0.787	0.799	0.812	0.82	0.832	0.847	0.864
SMB	0.897	0.914	0.917	0.921	0.924	0.927	0.932
German	0.522	0.542	0.56	0.742	0.737	0.767	0.778
Australia	0.58	0.599	0.603	0.597	0.617	0.618	0.627
Polish	0.446	0.607	0.623	0.655	0.658	0.674	0.702
Japanese	0.544	0.652	0.737	0.744	0.754	0.765	0.777

Table 11. G-mean comparison for DT

Dataset	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
GEM	0.758	0.771	0.909	0.912	0.925	0.931	0.947
STAR	0.618	0.628	0.634	0.653	0.66	0.663	0.675
SMB	0.813	0.839	0.849	0.85	0.868	0.881	0.884
German	0.48	0.527	0.534	0.55	0.596	0.613	0.623
Australia	0.63	0.639	0.648	0.648	0.722	0.721	0.754
Polish	0.57	0.608	0.696	0.725	0.764	0.82	0.823
Japanese	0.48	0.538	0.565	0.584	0.621	0.663	0.7

column, that is “None”, indicates that no sampling technique is applied.

To compare the results for the three evaluation indices, first Friedman’s rank is calculated (as it is shown in Tables 12 to 14) and then Friedman test (Adler et al., 2002) is applied. For each balancing technique, its rank in each dataset in calculated first, and then their average is obtained. Each technique is ranked from best to worst for each dataset. If there are multiple techniques with the same performance for the same dataset, the

Table 12. Friedman rank values for SVM

Evaluation Indices	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
AUC	7	6.07	5.07	3.5	3	2.64	1
F-measure	7	6	5	3.71	3.29	2	1.14
G-mean	7	6	5	3.36	3	2	1.14
Average	7	6.02	5.02	2.86	3.1	2.21	1.09

Table 13. Friedman rank values for ANN

Evaluation Indices	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
AUC	7	5.36	5.14	3.86	3.21	2.28	1.14
F-measure	7	6	5	3.79	3.21	2	1
G-mean	7	5.86	4.86	4.14	3.14	2	1
Average	7	5.74	5	3.93	3.19	2.09	1.05

Table 14. Friedman rank values for DT

Evaluation Indices	None	RUS	SMOTE	HBS	SMOTEBoost	RUSBoost	HBSBoost
AUC	6.85	6.21	5.21	3.71	3	2.29	1
F-measure	7	6	4.86	4	3	2.14	1
G-mean	7	6	4.93	4.07	2.86	2.14	1
Average	6.95	6.07	5	3.93	2.95	1.76	1

ranking is divided equally. The Friedman test statistics is based on the average rank, which is expressed in equation (11):

$$\mathcal{X}_{\mathcal{F}}^2 = \frac{12N}{k(k+1)}[\sum_{j=1}^{k} r_j^{\ 2} - \frac{k(k+1)^2}{4}] \qquad (11)$$

where N and k are the number of datasets and the number of balancing techniques respectively, and r_j is the average ranking of the j -th algorithm.

At a 5% significance level, the null hypothesis h_0 (There is no significant difference in the performance of the balancing techniques) is rejected and the performance of each algorithm can be considered to be significantly different.

Furthermore, the Nemenyi post-hoc test (Nemenyi, 1963) is applied to report any significant difference for all the sampling techniques. The Nemenyi test states that the performances of two or more balancing techniques are significantly different if their average ranks differ by at least the critical difference (CD), expressed in equation (12):

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \qquad (12)$$

where q_α is based on the studentized range statistic, N is the number of datasets and k is the number of balancing techniques. The test was implemented for all three evaluation indices and the test was carried out with a 5% significance level ($\alpha = 0.05$). Tables 15 to 17 show the P-values for the Nemenyi test for different pairs of balancing techniques for each classifier.

Table 15. P-values of the Nemenyi test for SVM

Balancing Technique	P-value
HBSBoost vs. RUS	0.00000018
HBSBoost vs. SMOTE	0.0000062
HBSBoost vs. HBS	0.00084
HBSBoost vs. SMOTEBoost	0.000041
HBSBoost vs. RUSBoost	0.000029

Table 16. P-values of the Nemenyi test for ANN

Balancing Technique	P-value
HBSBoost vs. RUS	0.000073
HBSBoost vs. SMOTE	0.00000036
HBSBoost vs. HBS	0.000034
HBSBoost vs. SMOTEBoost	0.00009
HBSBoost vs. RUSBoost	0.0025

Table 17. P-values of the Nemenyi test for DT

Balancing Technique	P-value
HBSBoost vs. RUS	0.00000035
HBSBoost vs. SMOTE	0.00054
HBSBoost vs. HBS	0.0000017
HBSBoost vs. SMOTEBoost	0.000042
HBSBoost vs. RUSBoost	0.0038

From these results, several aspects can be noticed.

All three metrics under the HBSBoost algorithm perform better than in the case when only the classiers are used, which indicates the predictive power of the HBSBoost technique. Among them, the combination of SVM and HBSBoost achieves the best training results, and the training results are significantly improved in comparison with those obtained under SVM alone. For the ANN and DT classifiers, the HBSBoost technique also shows a clear improvement.

For all the methods employed, no matter which imbalanced data learning method, the training performance of its classifier on imbalanced data sets is improved. Among them, all sampling techniques obtain better results than those obtained in the case of training based solely on classifiers, and the boosting ensemble algorithm (SMOTEBoost, RUSBoost, HBSBoost) outperforms the sampling technique alone (RUS, HBS, SMOTE). In comparison with all the sampling methods, except for ANN classifiers, hybrid sampling is better than SMOTE algorithm, which is better than random under-sampling because SVM and DT algorithms generate classification bias when classifying imbalanced data, which is mitigated after data balancing. As ensemble learning approaches are concerned, RUSBoost significantly outperforms SMOTEBoost in most of the datasets. For the SVM classifier, the proposed HBSBoost algorithm outperforms RUSBoost. According to all the data, both random under-sampling method and boosting method can improve the classification performance. In particular, the combined performance of both random under-sampling method and boosting algorithm for DT is much better than that of random under-sampling method alone. For all classifiers, SMOTE algorithm combined with ensemble algorithm Boosting achieves better results than SMOTE alone. In case of the hybrid sampling and boosting algorithms, it was found that the improvement obtained by hybrid sampling alone is higher than that obtained by boosting.

5. Conclusion

This paper presents the application of the proposed data balancing technique, HBSBoost, to the problem of identifying defaulting enterprises. The conclusions are as follows: (1) Hybrid sampling technique can effectively balance the data; (2) in comparison with the existing techniques such as under-sampling, SMOTE, hybrid sampling, SMOTEBoost, and RUSBoost, HBSBoost is significantly effective in solving the classification problem for imbalanced datasets; (3) Support vector machine is superior to artificial neural network model with regard to classification performance for imbalanced datasets, and artificial neural network is superior to the decision tree model.

As imbalanced classification techniques are being developed and widely applied, their development in the area of credit risk measurement for enterprises will become more and more extensive. This paper extends the application of data pre-processing in the identification of defaulting enterprises and lays the foundation for improving the identification of defaulting enterprises. In addition to classical machine learning classification, there are many ensemble machine learning methods that can be validated with a set of data balancing algorithms. Moreover, the effectiveness of more sophisticated ensemble data balancing techniques would be the next research direction. With regard to the type of data balancing method, the hybrid sampling method which can automatically adjust the balance ratio would be the research direction in the next stage.

REFERENCES

Adler, N., Friedman, L. & Sinuany-Stern, Z. (2002). Review of ranking methods in the data envelopment analysis context, *European Journal of Operational Research*, *140*(2), 249-265. DOI: 10.1016/s0377-2217(02)00068-1

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

Chawla, N. V., Lazarevic, A., Hall, L. O. & Bowyer, K. W. (2003). SMOTEBoost: Improving prediction of the minority class in boosting, In *European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 107-119). Springer, Berlin, Heidelberg.

Ciampi, F., Cillo, V. & Fiano, F. (2020). Combining Kohonen maps and prior payment behavior for small enterprise default prediction, *Small Business Economics*, *54*(4), 1007-1039. DOI: 10.1007/s11187-018-0117-2

Dastile, X., Celik, T. & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey, *Applied Soft Computing*, *91*, 106263. DOI: 10.1016/j.asoc.2020.106263

Du, M., Ma, Y. & Zhang, Z. (2022). A Meta-Path-Based Evaluation Method for Enterprise Credit Risk, *Computational Intelligence and Neuroscience*, 2022(6), DOI: 10.1155/2022/1783445

Gao, X., Ren, B., Zhang, H., Sun, B., Li, J., Xu, J., He, Y. & Li, K. (2020). An ensemble imbalanced classification method based on model dynamic selection driven by data partition hybrid sampling, *Expert Systems with Applications*, *160*, 113660. DOI: 10.1016/j.eswa.2020.113660

García, V., Marqués, A. I. & Sánchez, J. S. (2012). On the use of data filtering techniques for credit risk prediction with instance-based models, *Expert Systems with Applications*, *39*(18), 13267-13276. DOI: 10.1016/j.eswa.2012.05.075

García, V., Sánchez, J. S. & Mollineda, R. A. (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance, *Knowledge-Based Systems*, *25*(1), 13-21. DOI: 10.1016/j.knosys.2011.06.013

Gowda, K. & Krishna, G. (1979). The condensed nearest neighbor rule using the concept of mutual nearest neighborhood (corresp.), *IEEE Transactions on Information Theory*, *25*(4), 488-490. DOI: 10.1109/TIT.1979.1056066

Goyal, J. & Vajpayee, E. A. (2017). Improving classification performance using ensemble learning approach, *International Journal of Research in Computer Application & Management, 7*(11), 81-87.

Johnson, J. M. & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance, *Journal of Big Data*, *6*(1), 1-54. DOI:10.1186/s40537-019-0192-5

Kinoshita, T., Fujiwara, K., Kano, M., Ogawa, K., Sumi, Y., Matsuo, M. & Kadotani, H. (2020). Sleep spindle detection using RUSBoost and synchrosqueezed wavelet transform, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *28*(2), 390-398. DOI:10.1109/TNSRE.2020.2964597

Lefa, M., Abd-Elkader, H. & Salem, R. (2022). Enhancement of Very Fast Decision Tree for Data Stream Mining, *Studies in Informatics and Control*, *31*(2), 49-60. DOI:10.24846/v31i2y202205

Nemenyi, P. B. (1963). *Distribution-free multiple comparisons*. Princeton University ProQuest Dissertation Publishing.

Ouadine, A. Y., Mjahed, M., Ayad, H. & El Kari, A. (2020). UAV quadrotor fault detection and isolation using artificial neural network and Hammerstein-Wiener model, *Studies in Informatics and Control*, *29*(3), 317-328. DOI: 10.24846/v29i3y202005

Popescu, D. I., Ceptureanu, S.-I., Alexandru, A. & Ceptureanu, E.-G. (2019). Relationships between knowledge absorptive capacity, innovation performance and information technology. Case study: The Romanian creative industries SMEs, *Studies in Informatics and Control*, *28*(4),463-476. DOI: 10.24846/v28i4y201910

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

Shahraki, A., Abbasi, M. & Haugen, Ø. (2020). Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost, *Engineering Applications of Artificial Intelligence*, *94*, 103770. DOI: 10.1016/j.engappai.2020.103770

Tahir, M. A., Kittler, J. & Yan, F. (2012). Inverse random under sampling for class imbalance problem and its application to multi-label classification, *Pattern Recognition, 45*(10), 3738-3750. DOI: 10.1016/j.patcog.2012.03.014

Tanveer, M., Gautam, C. & Suganthan, P. N. (2019). Comprehensive evaluation of twin SVM based classifiers on UCI datasets, *Applied Soft Computing*, *83*, 105617. DOI: 10.1016/j.asoc.2019.105617

Tomczak, J. M. & Zięba, M. (2015). Classification restricted Boltzmann machine for comprehensible credit scoring model, *Expert Systems with Applications*, *42*(4), 1789-1796. DOI: 10.1016/j.eswa.2014.10.016

Wang C., Yang B. & Wang H. Q. (2020). Multi-Objective Master Production Schedule for Balanced Production of Manufacturers, *International Journal of Simulation Modelling*, *19*(4), 678-688.

Yang, B., Chen, W. & Lin, C. (2017). The Algorithm and Simulation of Multi-Objective Sequence and Balancing Problem for Mixed Mode Assembly Line, *International Journal of Simulation Modelling, 16*(2), 357-367. DOI: 10.2507/IJSIMM16(2)CO10

Yee, W. S., Ng, H., Yap, T. T. V., Goh, V. T., Ng, K. H. & Cher, D. T. (2022). An evaluation study on the predictive models of breast cancer risk factor classification, *Journal of Logistics, Informatics and Service Science*, *9*(3), 129-145. DOI: 10.33168/LISS.2022.0310