Hindawi
Complexity
Volume 2019, Article ID 8460934, 12 pages
https://doi.org/10.1155/2019/8460934

Research Article

A Hybrid Approach Using Oversampling Technique and Cost-Sensitive Learning for Bankruptcy Prediction

Tuong Le,[1] **Minh Thanh Vo**,[2] **Bay Vo**,[3] **Mi Young Lee**,[1] **and Sung Wook Baik**[1]

[1]*Digital Contents Research Institute, Sejong University, Seoul, Republic of Korea*
[2]*Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam*
[3]*Faculty of Information Technology, Ho Chi Minh City University of Technology (HUTECH), Ho Chi Minh, Vietnam*

Correspondence should be addressed to Sung Wook Baik; sbaik@sejong.ac.kr

Received 30 April 2019; Accepted 11 July 2019; Published 5 August 2019

Guest Editor: Thiago C. Silva

The diagnosis of bankruptcy companies becomes extremely important for business owners, banks, governments, securities investors, and economic stakeholders to optimize the profitability as well as to minimize risks of investments. Many studies have been developed for bankruptcy prediction utilizing different machine learning approaches on various datasets around the world. Due to the class imbalance problem occurring in the bankruptcy datasets, several special techniques would be used to improve the prediction performance. Oversampling technique and cost-sensitive learning framework are two common methods for dealing with class imbalance problem. Using oversampling techniques and cost-sensitive learning framework independently also improves predictability. However, for datasets with very small balancing ratios, combining two above techniques will produce the better results. Therefore, this study develops a hybrid approach using oversampling technique and cost-sensitive learning, namely, HAOC for bankruptcy prediction on the Korean Bankruptcy dataset. The first module of HAOC is oversampling module with an optimal balancing ratio found in the first experiment that will give the best overall performance for the validation set. Then, the second module uses the cost-sensitive learning model, namely, CBoost algorithm to bankruptcy prediction. The experimental results show that HAOC will give the best performance value for bankruptcy prediction compared with the existing approaches.

1. Introduction

Machine learning and data mining [1–9], which is the process of learning in order to look for patterns in observations or data and make better decisions in the future based on the training samples, is widely used in various fields such as cybernetics [10–14], engineering [15–18], bioinformatics [19], medical informatics [20], economics [21–27], etc. Especially in economics, there are many issues for optimizing profits in the business such as customer lifetime value modeling (CLVM), churn customer modeling (CCM), dynamic pricing, customer segmentation, recommendation systems, etc. CLVM [23] is one of the most important models for eCommerce business. These models can identify, understand, and retain the most valuable customers in your business. With the obtained results from these models, the business managers may make a better business strategy to optimize profitability. CCM [24] can help the companies determine their customers who will stop using their services. The outputs of these models, the customer list, are important inputs of an algorithmic retention strategy because they help optimize discount offers, marketing campaigns, and other targeted marketing initiatives. Dynamic pricing models [25] are for flexibly pricing products based on several factors such as the level of interest of the target customer, demand of the market at the time of purchase, and whether the customer has engaged with a marketing campaign. Meanwhile, customer segmentation models [26, 27] group customers into personas based on specific variations among them using several clustering and classification algorithms. Recommendation systems are another major way by which machine learning proves its business value. Recommendation systems sift through large quantities of data to predict how likely any given customer is to purchase an item or enjoy a piece of content and then suggest those things to the user. The result is a customer experience that encourages better engagement

and reduces churn. Bank lending and systemic risk [28, 29] is another issue in economics sector that attracted a lot of attention. This model will find empirical evidence against diversification as a mean to reduce systemic risk.

Bankruptcy prediction is also a hot topic in the field of business attracted by many scientists on computer science as well as economics around the world. In computer science domain, bankruptcy prediction, a predictive machine learning model, is to analyze the financial statement of a firm to make predictions for its fate in the future. Based on the obtained results from this task, investors and managers will devise appropriate strategies for companies that are going bankrupt. Many studies have been developed in recent years to predict the firm bankruptcy using various approaches [30–32]. In 2015, Kim et al. [30] introduced an efficient boosting algorithm, namely, GMBoost, using geometric mean for dealing with the problem of imbalanced data occurring in bankruptcy datasets. This algorithm calculates the error of majority class and the error of minority class separately. Then geometric mean value of these values will be determined to calculate the weight values for the next phase. Next, a novel approach [31] utilizing eXtreme Gradient Boosting (XGB) with synthetic features was proposed for bankruptcy prediction. In this study, the synthetic features proposed are automatically generated by random selection of two existing features and random selection of the arithmetical operation which help to improve the prediction performance. Recently, Barboza et al. [32] performed and evaluated several existing classification models including SVC (linear and RBF kernels), artificial neural networks (ANN), logistic regression, boosting, Random Forest, and Bagging, for forecasting bankruptcy companies. The authors use a balanced bankruptcy dataset that includes 449 bankruptcy firms and 449 non-bankruptcy firms from 1985 to 2005 for training the above classifiers. The trained models will be evaluated by an imbalanced bankruptcy dataset collected between 2006 and 2013 that consists of 133 bankruptcy cases and 13,300 non-bankruptcy cases. The experimental results in this study indicate that three classifiers including boosting, bagging, and random forest provide better results for bankruptcy prediction.

In many datasets on various domains, class distribution is commonly imbalanced called by class imbalance problem. The minority class in these datasets consists of a small number of data points while the majority class has a very large number of data points. Specifically, the number of bankruptcies is extremely small compared to the normal companies in bankruptcy datasets. The traditional classification models have a big bias towards majority class in such datasets. It is the cause of reduced performance of the above models. Therefore, many methods are given to deal with class imbalance problem which are grouped into the four following categories [33]. *(1) Algorithm level approaches* adapt existing classifiers to bias the learning toward the minority class [34, 35] without changing training data. *(2) Data level approaches* change the class distribution by resampling the data space [36, 37] to improve the predictive performance. There are three subcategories in this group including undersampling, oversampling, and hybrids techniques. Undersampling techniques balance the data distribution by removing the real data samples in majority class while oversampling techniques add the synthetic data samples to minority class. Meanwhile, hybrids techniques combine both undersampling and oversampling techniques. *(3) Cost-sensitive learning framework* is the hybrid methods that combine data and algorithm level approaches. These frameworks add costs to data samples (data level) and modify the learning process to accept costs (algorithm level) [38, 39]. The classifier in this group is biased toward the minority class by assuming higher misclassification costs for this class and seeking to minimize the total cost errors of both classes. *(4) Ensemble-based methods* usually consist of a combination of an ensemble learning algorithm and one of the techniques above, specifically, data level and cost-sensitive ones [40]. By combining data level approach to the ensemble learning algorithm, the new hybrid method usually preprocesses the data before training each classifier, whereas cost-sensitive ensembles, instead of modifying the base classifier in order to accept costs in the learning process, guide the cost minimization via the ensemble learning algorithm. The above four methods are used depending on the datasets to improve performance.

In 2018, Le et al. [41] first introduced the Korean Bankruptcy dataset denoted by KRBDS. In this study, the authors presented the oversampling based (OSB) framework that utilizes the oversampling techniques, a technique belonging to data level approach, for dealing with the class imbalance problem to predict the bankruptcy. This framework found that SMOTE-ENN is the best oversampling technique for KRBDS. Then, Le et al. [42] proposed a cluster-based boosting (CBoost) algorithm for dealing with the class imbalance problem. CBoost approach is considered as a cost-sensitive learning framework for dealing with the class imbalance problem. The framework, namely, RFCI, based on CBoost algorithm achieves the best AUC (The area under the receiver operating characteristics curve) with a shorter processing time compared with the first framework and several methods for bankruptcy prediction. In this study, we propose a hybrid approach, namely, HAOC, that combines the oversampling technique and cost-sensitive learning framework together for bankruptcy prediction. Our proposed approach firstly uses SMOTE-ENN to adjust class distribution of KRBDS with specific balancing ratio. Then, HAOC will use CBoost algorithm to predict the bankruptcy. The first experiment was conducted to find the best normalization technique among StandardScaler, MinMaxScaler, and RobustScaler for KRBDS. The second experiment is to find the optimal balancing ratio for oversampling phase. The comparison between HAOC with the existing approaches will be evaluated in the third experiment.

The rest of this manuscript is structured as follows. Section 2 first summarized the experimental dataset, namely, KRBDS, an oversampling technique, namely, SMOTE-ENN, and the CBoost algorithm. As the main contribution of this study, Section 2 introduces the hybrid approach for bankruptcy prediction, namely, HAOC. Two experiments were conducted to find the optimal balancing ratio and to show the effectiveness of proposed approach for bankruptcy prediction. Finally, the conclusions as well as several future

Table 1: The statistical information of KRBDS.

Feature	Description	Max	Min	Mean	Standard Deviation	Median	P25	P75
F1	Current assets	2.2×10^{11}	0	2.2×10^{7}	9.2×10^{8}	2.2×10^{6}	8.0×10^{5}	6.5×10^{6}
F2	Fixed assets, or fixed capital property	9.5×10^{10}	0	2.9×10^{7}	6.5×10^{8}	1.4×10^{6}	2.9×10^{5}	6.8×10^{6}
F3	Total assets	2.5×10^{11}	0	6.2×10^{7}	1.7×10^{9}	4.5×10^{6}	1.5×10^{6}	1.5×10^{7}
F4	Current liabilities within one year	2.1×10^{11}	-1.2×10^{6}	1.8×10^{7}	8.9×10^{8}	1.1×10^{6}	2.9×10^{5}	5.2×10^{6}
F5	Non-current liabilities.	6.5×10^{11}	-7.7×10^{5}	2.2×10^{7}	2.5×10^{9}	4.2×10^{5}	1.2×10^{4}	2.2×10^{6}
F6	Total liabilities	6.5×10^{11}	-2.1×10^{5}	4.9×10^{7}	2.9×10^{9}	2.1×10^{6}	5.5×10^{5}	8.3×10^{6}
F7	Capital	1.6×10^{10}	-2.9×10^{7}	5.1×10^{6}	1.2×10^{8}	4.0×10^{5}	1.5×10^{5}	1.0×10^{6}
F8	Earned surplus	4.8×10^{10}	-6.4×10^{11}	1.4×10^{6}	2.5×10^{9}	8.3×10^{5}	1.2×10^{5}	3.2×10^{6}
F9	Total capital	5.5×10^{10}	-6.3×10^{11}	1.3×10^{7}	2.5×10^{9}	1.7×10^{6}	5.4×10^{5}	5.5×10^{6}
F10	Total capital after liabilities	2.5×10^{11}	-4.3×10^{4}	6.2×10^{7}	1.7×10^{9}	4.5×10^{6}	1.4×10^{6}	1.5×10^{7}
F11	Sales revenue	6.0×10^{10}	-1.4×10^{9}	3.6×10^{7}	5.2×10^{8}	5.1×10^{6}	1.8×10^{6}	1.5×10^{7}
F12	Cost of sales	5.4×10^{10}	-4.7×10^{6}	2.7×10^{7}	4.2×10^{8}	3.4×10^{6}	8.6×10^{5}	1.1×10^{7}
F13	Net profit	2.5×10^{10}	-2.6×10^{10}	7.3×10^{6}	1.6×10^{8}	1.1×10^{6}	4.2×10^{5}	3.1×10^{6}
F14	Sales and administrative expenses	1.3×10^{10}	-5.2×10^{6}	5.5×10^{6}	9.6×10^{7}	8.8×10^{5}	3.4×10^{5}	2.4×10^{6}
F15	Operating profit that refers to the profits earned through business operations	2.5×10^{10}	-2.6×10^{10}	1.9×10^{6}	1.1×10^{8}	1.9×10^{5}	3.6×10^{4}	6.5×10^{5}
F16	Non-operating income	1.0×10^{10}	-4.4×10^{5}	1.6×10^{6}	5.1×10^{7}	4.3×10^{4}	8.1×10^{3}	2.2×10^{5}
F17	Non-operating expenses	3.0×10^{9}	-5.5×10^{5}	1.6×10^{6}	2.8×10^{7}	6.6×10^{4}	1.2×10^{4}	3.2×10^{5}
F18	Income and loss before income taxes	2.8×10^{10}	-2.3×10^{10}	2.0×10^{6}	1.2×10^{8}	1.6×10^{5}	3.3×10^{4}	5.8×10^{5}
F19	Net income	2.8×10^{10}	-2.3×10^{10}	1.5×10^{6}	1.2×10^{8}	1.4×10^{5}	2.9×10^{4}	5.0×10^{5}

research issues related to bankruptcy prediction are given in Section 4.

2. Materials and Methods

This section firstly introduces the experimental dataset, namely, KRBDS. Then, we summarize the oversampling technique named SMOTE-ENN and the cost-sensitive learning framework named CBoost algorithm. Finally, the proposed approach, namely, HAOC, will be introduced.

2.1. The Experimental Dataset. KRBDS was first introduced by Le et al. [41] that was provided by a Korean financial company. From the financial statements released by Korean companies from 2016 to 2017, nineteen financial features that have frequently been used in the previous bankruptcy prediction studies including assets, liabilities, capital, profit, etc. were extracted. Assets are any resources owned by the business such as buildings, equipment, and stocks while a liability is defined as any type of borrowing from persons or banks for improving their business. In addition, capital is any economic resource used by entrepreneurs and businesses to buy what they need to make their products or to provide their services. Meanwhile, profit is a financial benefit that is realized when the amount of revenue gained from a business activity exceeds the expenses, costs, and taxes needed to sustain the activity. These values are extremely important in finance to consider the company's performance, especially bankruptcy prediction. These features and some statistical information including maximum, minimum, and mean are shown and described in Table 1.

There are 307 bankrupted firms and 120,048 normal firms in KRBDS which has the balancing ratio of 0.0026. This ratio is extreme small for the normal classifier to predict bankruptcy correctly. Therefore, we need to develop several specific techniques to improve the performance.

2.2. Oversampling Technique with MOTE-ENN. Resampling technique belonging to data level approaches for dealing with class imbalance problem is the most common approach by adjusting the class distribution. Resampling technique consists of three subcategories including oversampling techniques, undersampling techniques, and hybrids techniques as illustrated in Figure 1. Undersampling technique balances the data distribution by removing the real data samples in majority class while oversampling technique accomplishes that purpose by adding the synthetic data samples to minority class. Meanwhile, hybrids methods combine both undersampling and oversampling techniques.

The advantage of these techniques is to balance the class distribution for improving the predictive performance. However, there is no absolute advantage of one resampling method

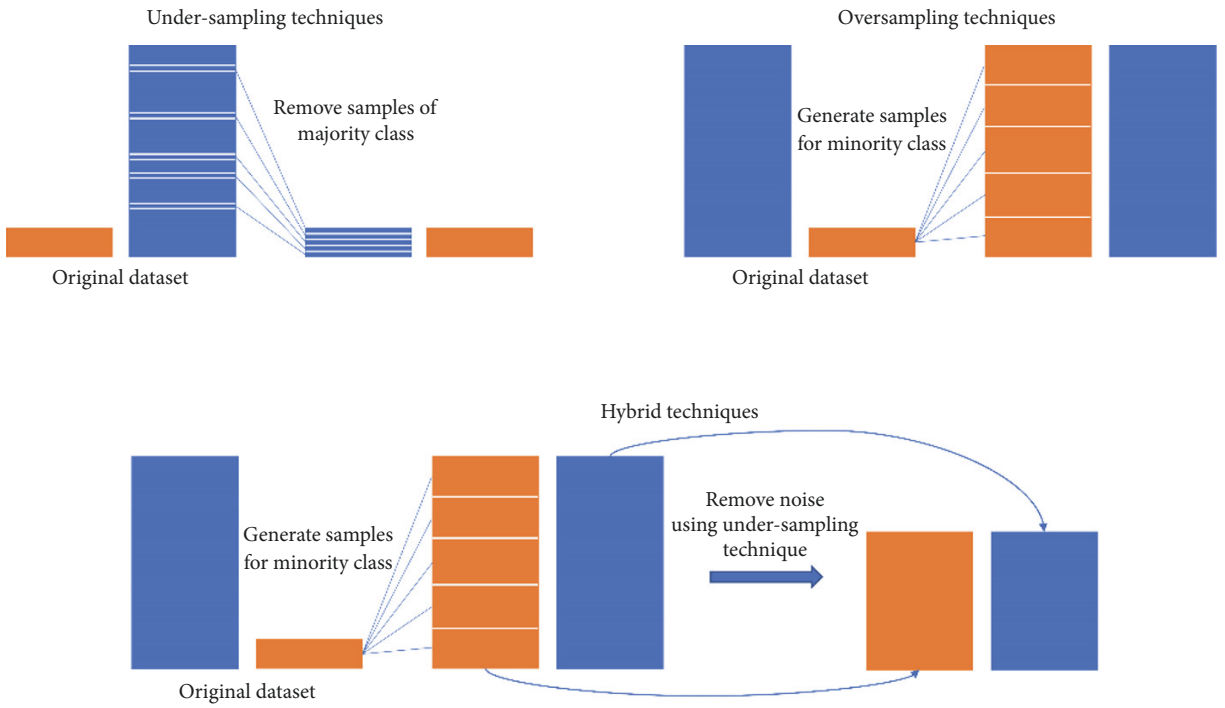

Figure 1: The illustration of oversampling, undersampling, and hybrids techniques.

over another. Application of these techniques depends on the use case it applies to and the dataset itself. Meanwhile, the disadvantage of undersampling techniques is that they can remove potentially useful data samples that could be important for the induction process. When the number of samples in the minority class is too small compared to that of samples in the majority class like KRBDS, undersampling techniques became ineffective. In this case, many samples in majority class are deleted. In addition, the main disadvantage with oversampling is that, by making exact copies of existing examples, it makes overfitting likely. A second disadvantage of oversampling is that it increases the number of training examples. Thus, the systems increase training time and the amount of memory required to hold the training set.

In 2018, Le et al. [41] conducted the oversampling framework that presents the empirical evaluation of oversampling techniques for bankruptcy prediction on KRBDS. Several oversampling techniques such as Synthetic Minority Oversampling Technique (SMOTE) [36], Borderline-SMOTE [44], Adaptive Synthetic (ADASYN) sampling approach [45], SMOTE-ENN [46], and SMOTE-Tomek [46] were used to improve the bankruptcy prediction performance. The experiments conducted in this study found that SMOTE-ENN is the best oversampling technique for KRBDS. This approach is summarized as follows.

The SMOTE algorithm was first proposed by Chawla et al. [36] in 2002 that generates synthetic minority samples based on the feature similarities between the original minority samples. Firstly, SMOTE determines the k-nearest neighbors (NNs) which is denoted by $\mathcal{K}_{\mathbf{x}_i}$ for each minority sample $\mathbf{x}_i \in \chi_{min}$.

Figure 2(a) demonstrates the three NNs of $\mathbf{x}_i$ that connect with $\mathbf{x}_i$ by a line. To generate a synthetic data sample ($\mathbf{x}_{new}$) for $\mathbf{x}_i$, SMOTE randomly selects an element $\hat{\mathbf{x}}_i$ in $\mathcal{K}_{\mathbf{x}_i}$ and $\hat{\mathbf{x}}_i$ in χ_{min}. The feature vector of $\mathbf{x}_{new}$ is the sum of the feature vectors of $\mathbf{x}_i$ and the value, which can be obtained by multiplying the vector difference between $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ with a random value δ from 0 to 1 ($\delta \in [0, 1]$), as the following equation:

$$\mathbf{x}_{new} = \mathbf{x}_i + \left(\hat{\mathbf{x}}_i - \mathbf{x}_i\right) \times \delta \tag{1}$$

where $\hat{\mathbf{x}}_i$ is an element in $\mathcal{K}_{\mathbf{x}_i}$: $\hat{\mathbf{x}}_i \in \chi_{min}$.

According to (1), the synthetic sample is a point along the line segment joining $\mathbf{x}_i$ and the randomly selected $\hat{\mathbf{x}}_i \in \mathcal{K}_{\mathbf{x}_i}$. Figure 2(b) shows a toy example of the SMOTE algorithm. The new sample $\mathbf{x}_{new}$ is in the line between $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$.

Then, SMOTE-ENN will apply the neighborhood cleaning rule based on the edited nearest neighbor (ENN) [46] to clean unwanted overlapping between classes, which removes samples that differ from two samples in the three nearest neighbors. Figure 3 shows the example of an ENN. Generally, SMOTE-ENN also uses SMOTE for the oversampling step and then uses ENN to remove the overlapping examples as shown in Figure 4.

2.3. Cluster-Based Boosting Algorithm. Recently, Le et al. [42] proposed CBoost algorithm that is based on the cost-sensitive learning framework for dealing with the class imbalance problem occurring in bankruptcy datasets effectively. CBoost algorithm first clusters the majority class in the bankruptcy datasets, i.e., the non-bankruptcy firms, by applied k-mean clustering with $k = 45$ which is considered as the best k value

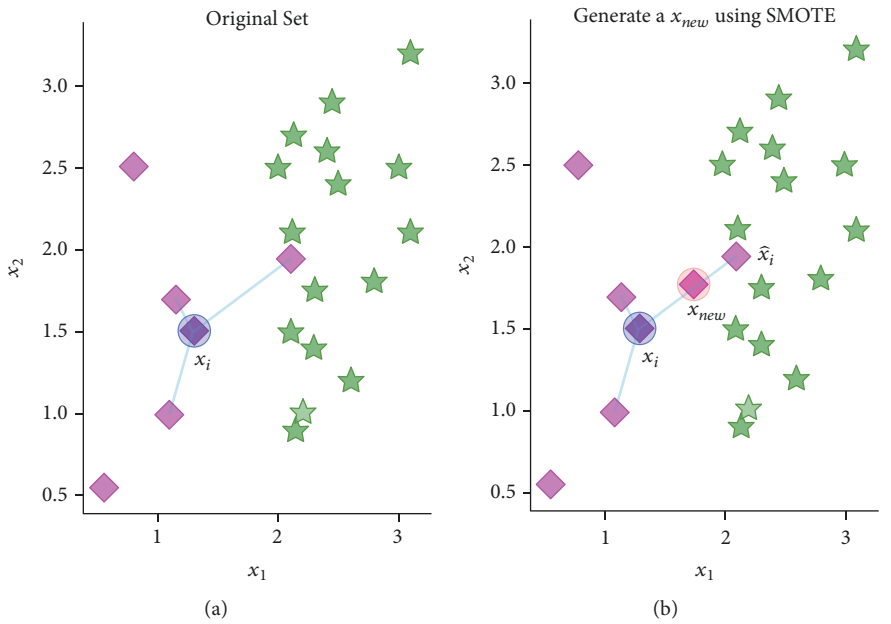

Figure 2: A toy example of the three-nearest neighbors for the $\mathbf{x}_i$ (a); and generate $\mathbf{x}_{new}$ using SMOTE (b).

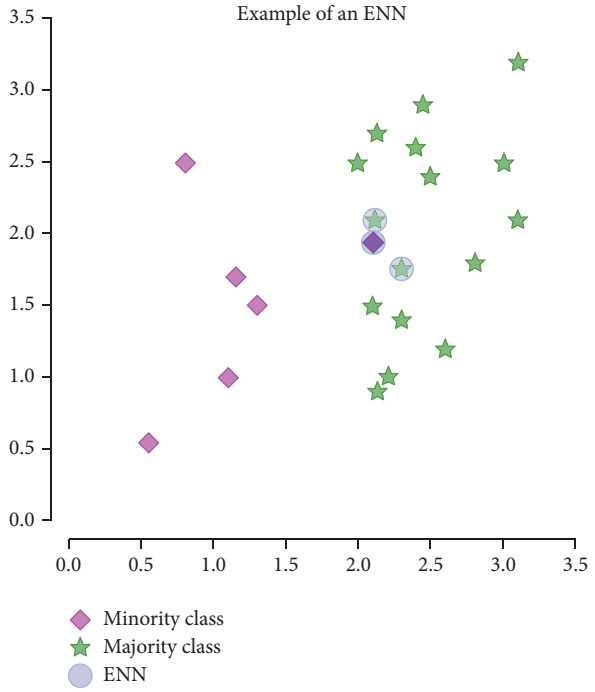

Figure 3: Example of an ENN.

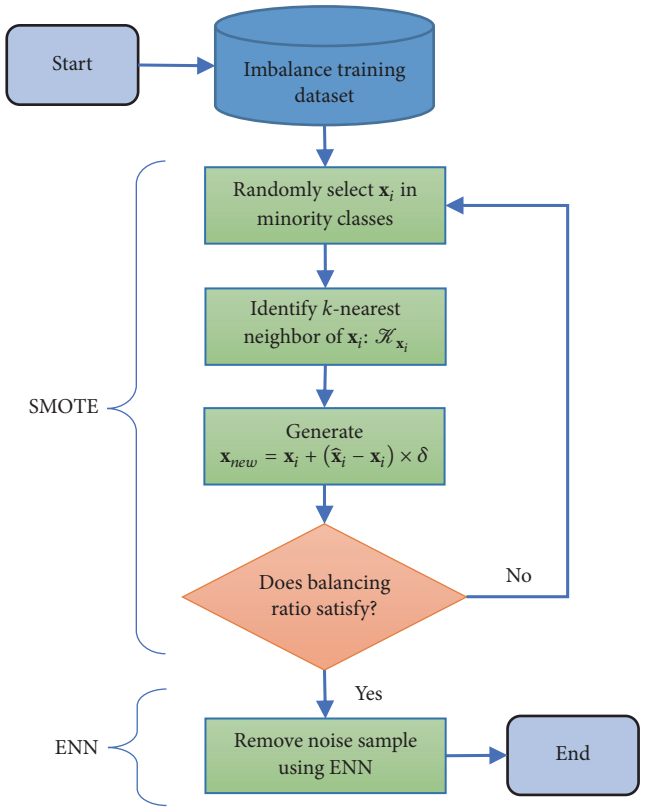

Figure 4: The flowchart of SMOTE-ENN algorithm.

based on the experimental results in [42]. Then, for each sample belonging to the majority class the algorithm will determine the distance from this sample to the nearest center point. Let d_{max} be the maximum value of the distances of data samples in class of bankruptcy firms. CBoost algorithm then assigns the values of each data sample in the minority class equal to d_{max}. Then, CBoost algorithm will determine the initial weights denoted by W_1 as follows:

$$W_1(i) = \ln\left(\frac{1}{d(\mathbf{x}_i)}\right) \tag{2}$$

where $d(\mathbf{x}_i)$ refers to the distance between data point $\mathbf{x}_i$ and the nearest center point for the majority class and $d(\mathbf{x}_i) = d_{max}$ for the minority class. Equation (2) makes it so that the samples in the majority class closed the center points and the samples in the minority class will have higher weight values compared to the further samples in majority class. CBoost will then normalize these values by the following equation:

$$W_1(i) = \frac{W_1(i)}{\sum_{i=1}^{m} W_1(i)} \tag{3}$$

where m is the total number of data points in the training set. This step will ensure that

$$\sum W_1(i) = 1 \tag{4}$$

The initial weight W_1 helps the weak classifier classify more accurately the samples in the majority class close to the center points as well as the samples in the minority class. Therefore, it will improve the overall performance for class imbalance problem like bankruptcy dataset.

For each iteration, CBoost identifies the weak learner denoted by $h_t(\mathbf{x})$ that produces the lowest classification error denoted by ϵ_t, calculates the weight for this classifier denoted by α_t, and determines the next weight W_{t+1} for the next iteration as follows.

$$h_t = \underset{h_j \in \mathcal{H}}{\operatorname{argmin}} \epsilon_j = \sum_{i=1}^{m} W_t(i) \left[y_i \neq h_j(\mathbf{x}_i) \right]$$

$$\alpha_t = \eta \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{5}$$

$$W_{t+1}(i) = \frac{W_t(i) \exp\left(-\alpha_t y_i h_t(\mathbf{x}_i)\right)}{Z_t}$$

where Z_t is normalization factor. Finally, the algorithm will combine all weak learners to make the final classifier H as follows.

$$H(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right) \tag{6}$$

where $h_t(\mathbf{x})$ is the weak learner at the iteration t-th and α_t is its weight.

In short, CBoost is a greedy algorithm that finds one weak learner at an iteration, optimizes the weight of this learner, and updates the weighted distribution for the next iteration. The algorithm combines all weak learners as in (5) to create the final classifier. The flowchart of CBoost algorithm is shown in Figure 5.

2.4. The Hybrid Approach for Bankruptcy Prediction on KRBDS. The balancing ratio of KRBDS is very small which leads to a reduction in performance of oversampling and cost-sensitive learning independently. Therefore, this study proposes a hybrid approach that combines oversampling technique and cost-sensitive learning (HAOC) for bankruptcy prediction on KRBDS to improve the overall performance.

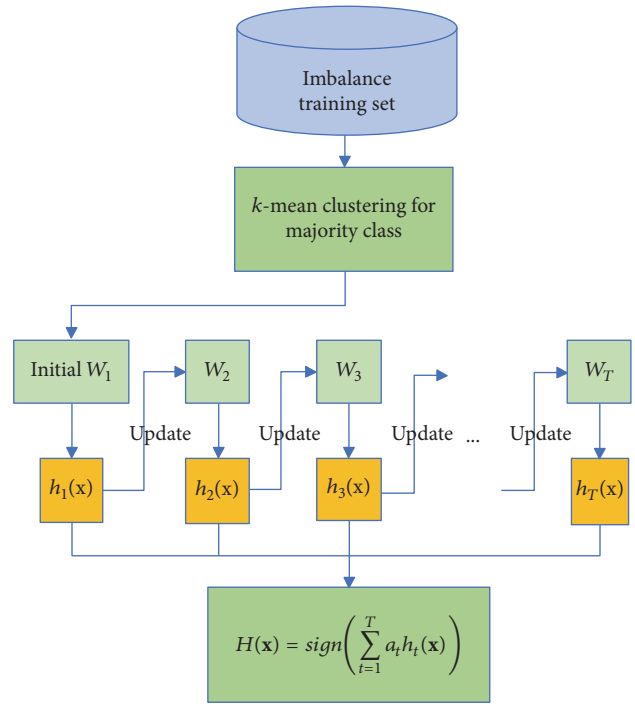

Figure 5: The flowchart of CBoost algorithm.

The flowchart of HAOC is presented in Figure 6. KRBDS is first normalized by using a normalization module that uses the best normalization technique in the first experiment (Data preprocessing). Next, the fivefold cross-validation module will be used to split the KRBDS into five parts, in which four parts were used for training and the remaining part was used for testing alternately.

The training set will be put into the found optimal balancing ratio module. This module will divide the training set into two subsets: the training set and validation set. Using these sets, this module tries various balancing ratios for SMOTE-ENN and will find the optimal balancing ratio for the KRBDS which will be presented in the first experiment. The training set will be balanced by SMOTE-ENN with the best balancing ratio that was found in the previous step. After this phase, the resample training set will be utilized to train the CBoost algorithm for bankruptcy prediction later. The testing set will be used to evaluate the proposed approach.

3. Experimental Results

3.1. Experiment Setup. The experimental methods were implemented in Python 2.7 environment and performed on a computer with Intel Core i7-2600 CPU (3.40 GHz × 2 cores), 8 GB RAM that runs with Ubuntu 16.04 LTS. In addition, SMOTEENN was implemented by the imbalanced-learn package [47] and Bagging, AdaBoost, Random Forest, and MLP were in Scikit-learn package [48]. The imbalanced-learn package is an open-source Python toolbox which consists of several methods for dealing with the problem of class imbalance while Scikit-learn package is a free software machine learning library for the Python programming language.

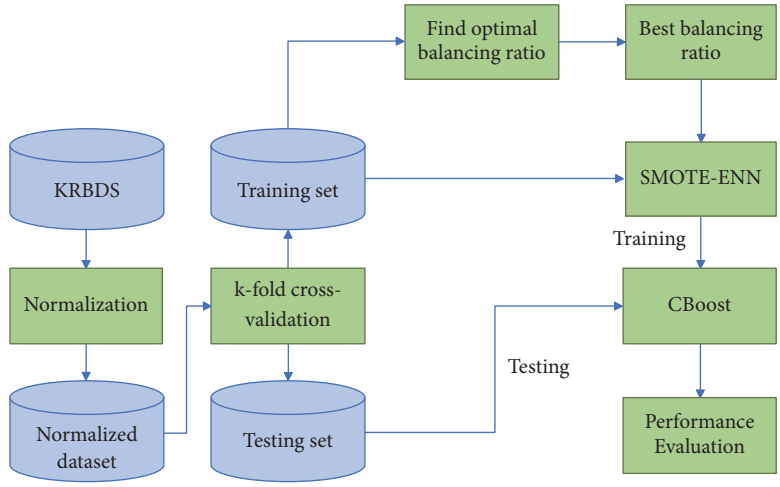

Figure 6: The flowchart of HAOC.

To show the effectiveness of the proposed approach, we compare the performance among the state-of-the-art methods and HAOC for bankruptcy prediction on KRBDS. The first four approaches are Bagging (BG), AdaBoost (AB), Random Forest (RF), and Multilayer Perceptron (MLP) which were recommended by Barboza et al. [32]. These approaches were used to predict bankruptcy directly; i.e., there is no resample approach applied to adjust the class distribution. The 5^{th} to 8^{th} approaches combine undersampling method based on clustering technique [43] with BG, AB, RF, and MLP classifiers. The 9^{th}-12^{th} approaches are oversampling method using SMOTE-ENN (with balancing ratio = 1) combined with BG, AB, RF, and MLP classifiers to predict bankruptcy. The 13^{th} approach is RFCI introduced by Le et al. [42] and the 14^{th} approach is the proposed approach (HAOC). Moreover, the study employs the fivefold cross-validation in 10 times with different configurations of folds for each run to get the average performance.

Next, we use GridSearchCV in Scikit-learn package [48] to turn several parameters of Bagging, AdaBoost, Random Forest, and MLP. We turned the *n_estimators* (150) and *max_samples* (0.2) for Bagging, *learning_rate* (0.1) for AdaBoost, *max_depth* (5) for Random Forest, and *max_iter* (150), *learning_rate_init* (0.01), and *hidden_layer_sizes* (50, 5) for MLP.

3.2. Evaluation Metrics. This study uses two evaluation metrics including AUC (Area under the ROC Curve) and G-mean (Geometric Mean) to compare the performance among the experimental methods. A ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots True Positive Rate (TPR) and False Positive Rate (FPR) computed as follows.

$$TPR = \frac{TP}{TP + FN}$$
$$FPR = \frac{FP}{FP + TN} \quad (7)$$

where TP, FN, FP, and TN are true positives, false negatives, false positives, and true negatives, respectively. Lowering the classification threshold classifies more items as positive, thus increasing both false positives and true positives. AUC (Area under the ROC Curve) provides an aggregate measure of performance across all possible classification thresholds. If an algorithm has a larger AUC than that of another algorithm, this algorithm is better.

From ROC, Youden index which is the vertical distance between the 45-degree line and the point on the ROC curve was used to determine the optimal cut-off threshold. The Youden index is determined as follows.

$$J = sensitivity + specificity - 1 \quad (8)$$

The optimal cut-off threshold corresponds to the point with the maximum value of J. From that threshold, sensitivity and specificity, respectively, will be determined. G-mean is the root of the product of classwise sensitivity. This measure tries to maximize the accuracy on each of the classes while keeping these accuracies balanced. For binary classification G-mean is the squared root of the product of the sensitivity and specificity. Similar to the AUC, the algorithm with a larger G-mean is better.

3.3. Data Preprocessing. In this section, we apply some normalization techniques including StandardScaler, MinMaxScaler, and RobustScaler to the original features. StandardScaler normalizes the original features to create standardized features by removing the mean and scaling to unit variance. MinMaxScaler transforms the features by scaling each feature to a given range while RobustScaler scales the features using statistics that are robust to outliers. HAOC is then used to predict the bankruptcy from the normalized features. The performance results in Table 2 show that the StandardScaler is the best normalization technique for KRBDS. Therefore, we apply the StandardScaler for the next experiments. Please note that the settings of StandardScaler were found only using training data and then we used these settings for the training and testing data.

Table 2: Performance results of HAOC using several normalization techniques for KRBDS.

No	Normalization technique	Normalization formula	AUC
1	None	None	50.0±0.0
2	StandardScaler	$\mathbf{x}' = \frac{\mathbf{x} - \mathbf{x}_{mean}}{\mathbf{x}_{stdev}}$	*87.1±0.6*
3	MinMaxScaler	$\mathbf{x}' = \frac{\mathbf{x} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}$	73.0±4.0
4	RobustScaler	$\mathbf{x}' = \frac{\mathbf{x} - \mathbf{x}_{Q_1}}{\mathbf{x}_{Q_3} - \mathbf{x}_{Q_1}}$	50.0±0.0

Table 3: The overall results of all experimental approaches for KRBDS.

No	Method	Resample approach	Classifier	AUC	G-mean	Average Rank	p-value
1	BG	None	Bagging	78.8±0.4	70.8±0.8	9.0	3.9×10^{-5}
2	AB	None	AdaBoost	84.9±0.8	78.2±0.6	7.0	0.0023
3	RF	None	Random Forest	86.2±0.6	79.9±0.6	4.7	***0.069***
4	MLP	None	MLP	86.7±0.8	80.1±1.0	2.6	***0.487***
5	USC-BG	Under-sampling method based on clustering technique (USC) [43]	Bagging	65.1±1.6	53.6±4.9	11.2	1.2×10^{-7}
6	USC-AB		AdaBoost	59.7±3.0	56.3±5.0	12.9	5.6×10^{-10}
7	USC-RF		Random Forest	64.7±1.0	62.6±1.9	11.9	1.5×10^{-8}
8	USC-MLP		MLP	46.9±2.7	36.5±3.7	14.0	1.1×10^{-11}
9	OSE-BG	Oversampling method using SMOTE-ENN (OSE) [41]	Bagging	83.9±0.3	77.4±0.3	7.8	5.1×10^{-4}
10	OSE-AB		AdaBoost	85.4±0.7	78.5±0.4	6.2	0.009
11	OSE-RF		Random Forest	86.6±0.7	80.2±1.0	3.3	***0.285***
12	OSE-MLP		MLP	72.8±2.1	69.8±1.8	10.0	3.3×10^{-6}
13	RFCI [42]	Under-sampling method using IHT concept	CBoost	86.6±0.7	79.1±3.5	3.1	***0.336***
14	HAOC	Oversampling method using SMOTE-ENN (with balancing ratio = 0.08)	CBoost	*87.1±0.6*	*81.1±0.8*	*1.3*	-

3.4. Finding the Optimal Balancing Ratio. This section is conducted to find the optimal balancing ratio of HAOC for KRBDS. Using different balancing ratios from 0.003 to 1 for oversampling module, we obtain the AUCs for the valuation sets shown as Figure 7 in five folds. According to the results, we found that the balancing ratio at 0.08 gives the best average AUC for validation sets. Therefore, we use this value for our proposed approach in the final experiment.

3.5. Performance Results. Figure 8 shows the box plot in terms of AUC of the experimental approaches for KRBDS in five folds. We can easily see found that CUS_BG, CUS_AB, CUS_RF, CUS_MLP, and OSE_MLP did not achieve good results. The remaining approaches get more positive results.

Figure 9 presents the box plot in term of G-mean of all the experimental approaches which indicate that AB, RF, NLP, OSE_RF, RFCI, and HAOC are the best methods in terms of G-mean.

Table 3 presents the average AUCs and G-mean of these approaches with standard deviation. According to these

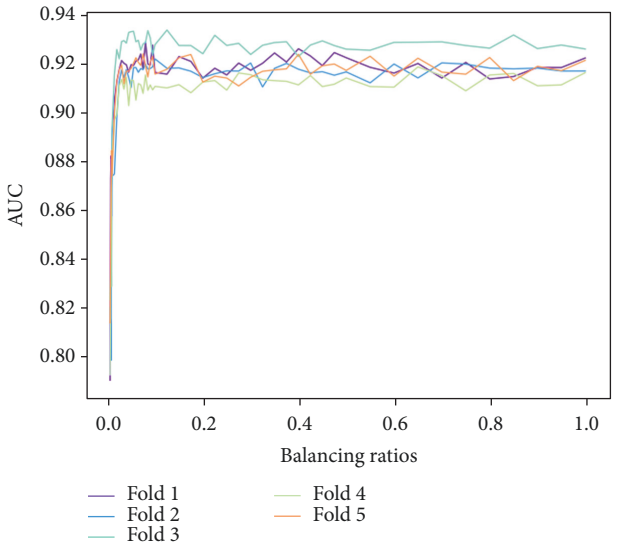

Figure 7: Performance of HAOC in terms of AUC for validation sets in five folds.

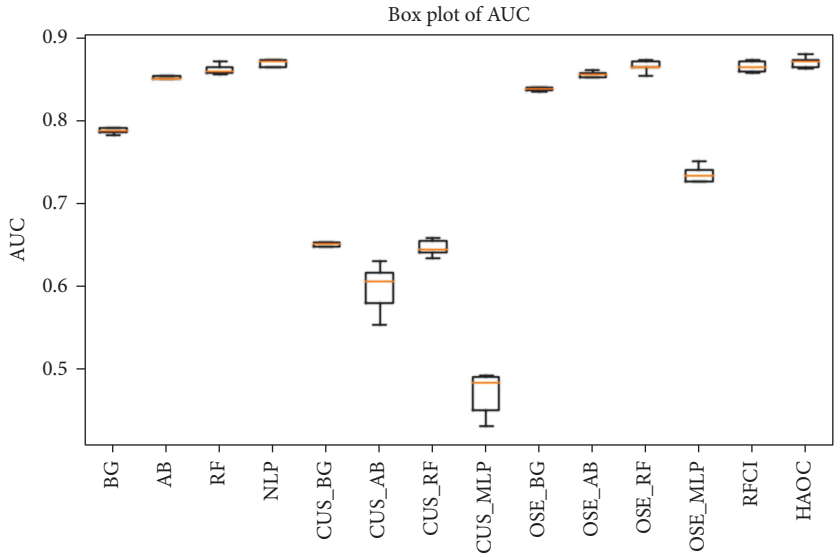

Figure 8: The box plot in terms of AUC of experimental approaches for KRBDS in five folds.

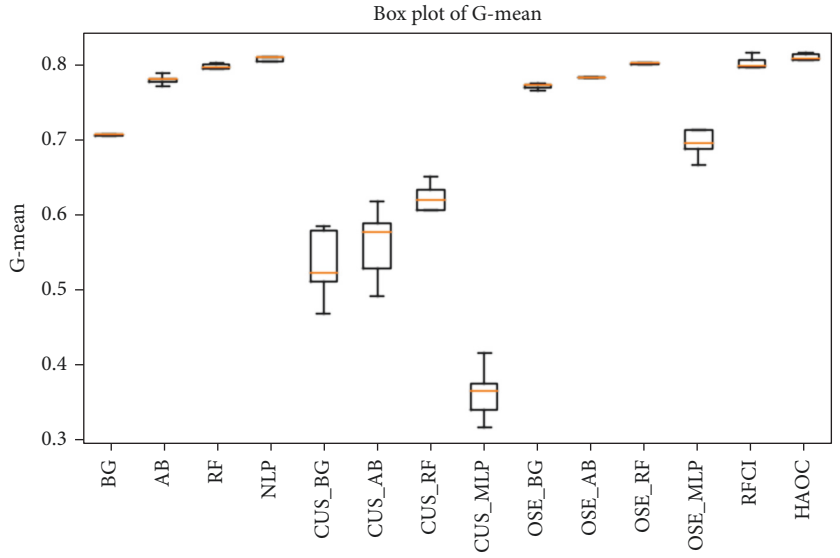

Figure 9: The box plot in terms of G-mean of experimental approaches for KRBDS in five folds.

results, Bagging without resample approach gives poor results at 78.8 in AUC, respectively. Meanwhile, AdaBoost, Random Forest, and MLP show acceptable results at 84.9, 86.2, and 86.7 in AUC. In addition, the undersampling method based on clustering technique (UCS) [43] is responsible for reducing the performance of classification algorithms including Bagging, MLP, RF, and AB. Therefore, UCS is not suitable for KRBDS when its balancing ratio is very small. The 9^{th}–12^{th} approaches, OSE-BG, OSE-AB, OSE-RF, and OSE-MLP, give the overall AUC at 83.9, 85.4, 86.6, and 72.8, respectively. Meanwhile, RFCI [42] that uses the cost-sensitive learning algorithm, namely, CBoost, achieved 86.6 in AUC. Our proposed method outperforms the other approaches when achieving the overall AUC at 87.1. Moreover, Table 3 also reports the G-mean of all experimental approaches. According to these results, HAOC achieves the best value of G-mean while OSE-RF obtains the second value. Besides, RFCI, MLP, RF, and OSE-RF also have good results. In general, the proposed approach has the best values which balance between AUC and G-mean for KRBDS.

In addition, we employ the MULTIPLETEST package [49] for conducting multiple comparisons involving all possible pairwise experimental methods whose results are also presented in Table 3. The average rank of the proposed method is 1.3 which is the best rank in terms of AUC. Also, it can be noted that the results of our proposal do not have statistical differences against those results obtained by Random Forest, MLP, OSE-RF, and RFCI when the p-values are greater than 0.05. In addition, the p-values (≤ 0.05) show that the differences in the results of HAOC

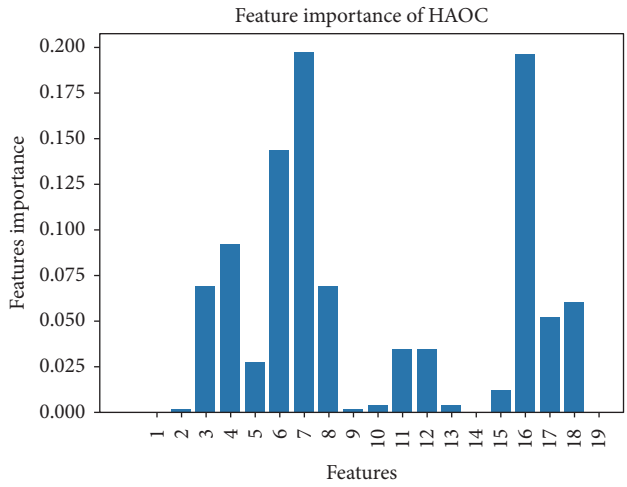

Figure 10: Feature importance of HAOC approach.

against the remaining tested classifiers are statistically significant.

Finally, Figure 10 presents the feature importance of HAOC approach on KRBDS. We can easily see that F3 (total assets), F4 (current liabilities within one year), F6 (total liabilities), F7 (capital), F8 (earned surplus), and F16 (nonoperating income) are the most important features. On the contrary, F1 (current assets), F2 (fixed assets, or fixed capital property), F9 (total capital), F10 (total capital after liabilities), F13 (net profit), F14 (sales and administrative expenses), and F19 (net income) are unimportant features and therefore they can be removed in the proposed model.

4. Conclusions

This study proposed a hybrid approach using oversampling technique and cost-sensitive learning framework for bankruptcy prediction on the Korean Bankruptcy dataset. In the first phase, the training set will be balanced by an oversampling module that utilizes the SMOTE-ENN algorithm with an optimal balancing ratio. Then, the second module uses the cost-sensitive learning framework, namely, CBoost, for bankruptcy prediction. Two experiments were conducted in this study to show the effectiveness of the proposed approach. The first experiment is to find the optimal balancing ratio that will give the best overall performance for bankruptcy prediction on the training set. Using the optimal balancing ratio that was found in the first experiment, we evaluate the performance in terms of AUC and G-mean between our proposed approach and the existing approaches. The results indicate that HAOC outperforms the existing approaches for bankruptcy prediction on KRBDS.

In the future, we will focus on how to find the optimal feature selection methods using evolutionary algorithms. In addition, several advanced methods for forecasting bankruptcy from multiple information sources to improve performance will be studied.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the National Program for Excellence in SW (2015-0-00938) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

References

[1] T. H. Cupertino, M. Guimarães Carneiro, Q. Zheng, J. Zhang, and L. Zhao, "A scheme for high level data classification using random walk and network measures," *Expert Systems with Applications*, vol. 92, pp. 289–303, 2018.

[2] T. C. Silva and L. Zhao, *Machine Learning in Complex Networks*, Springer, 2016.

[3] T. Le, B. Vo, P. Fournier-Viger, M. Y. Lee, and S. W. Baik, "SPPC: a new tree structure for mining erasable patterns in data streams," *Applied Intelligence*, vol. 49, no. 2, pp. 478–495, 2019.

[4] T. Le, B. Vo, and S. W. Baik, "Efficient algorithms for mining top-rank- k erasable patterns using pruning strategies and the subsume concept," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 1–9, 2018.

[5] T. Le, A. Nguyen, B. Huynh, B. Vo, and W. Pedrycz, "Mining constrained inter-sequence patterns: a novel approach to cope with item constraints," *Applied Intelligence*, vol. 48, no. 5, pp. 1327–1343, 2018.

[6] T. Kieu, B. Vo, T. Le, Z. Deng, and B. Le, "Mining top-k co-occurrence items with sequential pattern," *Expert Systems with Applications*, vol. 85, pp. 123–133, 2017.

[7] B. Vo, T. Le, F. Coenen, and T.-P. Hong, "Mining frequent itemsets using the n-list and subsume concepts," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 2, pp. 253–265, 2016.

[8] B. Vo, T. Le, G. Nguyen, and T. Hong, "Efficient algorithms for mining erasable closed patterns from product datasets," *IEEE Access*, vol. 5, pp. 3111–3120, 2017.

[9] G. Nguyen, T. Le, B. Vo, and B. Le, "EIFDD: An efficient approach for erasable itemset mining of very dense datasets," *Applied Intelligence*, vol. 43, no. 1, pp. 85–94, 2015.

[10] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of Internet of things for smart home: challenges and solutions," *Journal of Cleaner Production*, vol. 140, pp. 1454–1464, 2017.

[11] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[12] N. P. Nguyen and S. K. Hong, "Sliding mode Thau observer for actuator fault diagnosis of quadcopter UAVs," *Applied Sciences*, vol. 8, no. 10, article 1893, 2018.

[13] N. P. Nguyen and S. K. Hong, "Fault-Tolerant control of quadcopter uavs using robust adaptive sliding mode approach," *Energies*, vol. 12, no. 1, article 95, 2019.

[14] N. Nguyen and S. Hong, "Fault diagnosis and fault-tolerant control scheme for quadcopter UAVs with a total loss of actuator," *Energies*, vol. 12, no. 6, article 1139, 2019.

[15] T. N. Nguyen, S. Lee, H. Nguyen-Xuan, and J. Lee, "A novel analysis-prediction approach for geometrically nonlinear problems using group method of data handling," *Computer Methods Applied Mechanics and Engineering*, vol. 354, pp. 506–526, 2019.

[16] T. N. Nguyen, C. H. Thai, A. Luu, H. Nguyen-Xuan, and J. Lee, "NURBS-based postbuckling analysis of functionally graded carbon nanotube-reinforced composite shells," *Computer Methods Applied Mechanics and Engineering*, vol. 347, pp. 983–1003, 2019.

[17] T. N. Nguyen, C. H. Thai, H. Nguyen-Xuan, and J. Lee, "NURBS-based analyses of functionally graded carbon nanotube-reinforced composite shells," *Composite Structures*, vol. 203, pp. 349–360, 2018.

[18] T. N. Nguyen, C. H. Thai, H. Nguyen-Xuan, and J. Lee, "Geometrically nonlinear analysis of functionally graded material plates using an improved moving Kriging meshfree method based on a refined plate theory," *Composite Structures*, vol. 193, pp. 268–280, 2018.

[19] D. Le and V. Pham, "HGPEC: a Cytoscape app for prediction of novel disease-gene and disease-disease associations and evidence collection based on a random walk on heterogeneous network," *BMC Systems Biology*, vol. 11, no. 1, article 61, 2017.

[20] D. J. Hemanth, J. Anitha, and L. H. Son, "Brain signal based human emotion analysis by circular back propagation and deep kohonen neural networks," *Computers and Electrical Engineering*, vol. 68, pp. 170–180, 2018.

[21] D. M. Fazio, T. C. Silva, B. M. Tabak, and D. O. Cajueiro, "Inflation targeting and financial stability: Does the quality of institutions matter?" *Economic Modelling*, vol. 71, pp. 1–15, 2018.

[22] T. Le, B. Vo, H. Fujita, N. Nguyen, and S. W. Baik, "A fast and accurate approach for bankruptcy forecasting using squared logistics loss with GPU-based extreme gradient boosting," *Information Sciences*, vol. 494, pp. 294–310, 2019.

[23] A. Vanderveld, A. Pandey, A. Han, and R. Parekh, "An engagement-based customer lifetime value system for E-commerce," in *Proceedings of the the 22nd ACM SIGKDD International Conference*, pp. 293–302, San Francisco, Calif, USA, August 2016.

[24] B. Zhu, B. Baesens, and S. K. vanden Broucke, "An empirical comparison of techniques for the class imbalance problem in churn prediction," *Information Sciences*, vol. 408, pp. 84–99, 2017.

[25] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, "Decentralized cloud-SDN architecture in smart grid: a dynamic pricing model," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1220–1231, 2018.

[26] X. Chen, Y. Fang, M. Yang, F. Nie, Z. Zhao, and J. Z. Huang, "PurTreeClust: a clustering algorithm for customer segmentation from massive customer transaction data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 3, pp. 559–572, 2018.

[27] H. V. Long, L. H. Son, M. Khari et al., "A new approach for construction of geodemographic segmentation model and prediction analysis," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 9252837, 10 pages, 2019.

[28] T. C. Silva, M. D. S. Alexandre, and B. M. Tabak, "Bank lending and systemic risk: A financial-real sector network approach with feedback," *Journal of Financial Stability*, vol. 38, pp. 98–118, 2018.

[29] B. M. Tabak, T. C. Silva, and A. Sensoy, "Financial Networks," *Complexity*, vol. 2018, Article ID 7802590, 2 pages, 2018.

[30] M. Kim, D. Kang, and H. B. Kim, "Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1074–1082, 2015.

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

[32] F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications*, vol. 83, pp. 405–417, 2017.

[33] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*, Springer, 2018.

[34] Y. Lin, Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," *Machine Learning*, vol. 46, no. 1-3, pp. 191–202, 2002.

[35] B. Liu, Y. Ma, and C. Wong, "Improving an association rule-based classifier," *PKDD*, pp. 293–317, 2000.

[36] 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.

[37] T. Le and S. W. Baik, "A robust framework for self-care problem identification for children with disability," *Symmetry*, vol. 11, no. 1, article 89, 2019.

[38] N. V. Chawla, D. A. Cieslak, L. O. Hall, and A. Joshi, "Automatically countering imbalance and its empirical relationship to cost," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 225–252, 2008.

[39] C. Ling, V. Sheng, and Q. Yang, "Test strategies for cost-sensitive decision trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 1055–1067, 2006.

[40] 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.

[41] T. Le, M. Y. Lee, J. R. Park, and S. W. Baik, "Oversampling techniques for bankruptcy prediction: Novel features from a transaction dataset," *Symmetry*, vol. 10, no. 4, article 79, 2018.

[42] T. Le, L. H. Son, M. T. Vo, M. Y. Lee, and S. W. Baik, "A cluster-based boosting algorithm for bankruptcy prediction in a highly imbalanced dataset," *Symmetry*, vol. 10, no. 7, article 250, 2018.

[43] W. Lin, C. Tsai, Y. Hu, and J. Jhang, "Clustering-based undersampling in class-imbalanced data," *Information Sciences*, vol. 409-410, pp. 17–26, 2017.

[44] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in *Proceedings of the International Conference on Intelligent Computing (ICIC '05)*, vol. 3644 of *Lecture Notes in Computer Science*, pp. 878–887, August 2005.

[45] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: adaptive synthetic sampling approach for imbalanced learning," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 1322–1328, June 2008.

[46] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behaviour of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[47] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

[48] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[49] S. García and F. Herrera, "An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons," *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.