Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DEVELOPING AN INTELLIGENT SYSTEM FOR PREDICTING BANKRUPTCY

SAMAR ALY[1], MARCO ALFONSE[2], MOHAMED I. ROUSHDY[3], ABDEL-BADEEH M. SALEM[4]
[1,2,4]Computer Science Department, Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt
[3]Faculty of Computers & Information Technology, Future University in Egypt
E-mail: [1]samar.aly@cis.asu.edu.eg, [2]marco_alfonse@cis.asu.edu.eg, [3]mohamed.roushdy@fue.edu.eg, [4]absalem@cis.asu.edu.eg

ABSTRACT

Bankruptcy prediction is a very important metric for making economic decisions. It is a very critical element in credit risk analysis. Machine learning based classifiers have been heavily utilized in predicting bankruptcy. In terms of machine learning, predicting bankruptcy with imbalanced dataset is a very big challenge. Despite the current existing models for bankruptcy prediction, finding a model that achieves a high-performance measurement with imbalanced datasets is still an interesting point of research. The reason behind this challenge is the fact that imbalanced dataset leads to misclassification results. This paper aims to develop a robust machine learning based model for predicting bankruptcy with solving imbalanced dataset problem. Four different re-sampling strategies were applied to solve the imbalanced class distribution problem based on three popular datasets. The used datasets were downloaded from the University of California, Irvine (UCI) machine learning repository. The developed model based on different single and ensemble machine learning classifiers. The overall experimental results showed that the best performance of the developed model with the Polish dataset was 97% for accuracy and 95.4% for AUC (Area Under the Curve). Moreover, the best performance with the Australian dataset was 88.4% for accuracy and 92% for AUC. And the best performance with the German dataset was 81.5% for accuracy and 83.4% for AUC ratio.

Keywords: *Bankruptcy prediction, Artificial Intelligence, Ensemble Techniques, Machine Learning Techniques, Imbalanced Dataset*

1. INTRODUCTION

Bankruptcy prediction is an essential issue in determining credit risk for financial institutions [1]. Since the financial crisis of 2007, managing credit risk has become a top priority for financial institutions and financial regulators [2]. The severe economic and social consequences followed the financial crisis were due to several big bankruptcies [3]. Therefore, the high social and economic costs of corporate bankruptcies have attracted attention of researchers. They have been trying to understand the causes of bankruptcy and eventually prediction of business distress. The purpose of the bankruptcy prediction is to assess the financial condition of an institution and its future perspectives within the context of long-term operation in the market [4].

Recently, several studies apply Artificial Intelligence (AI) techniques to predict bankruptcy problem instead of the traditional statistical techniques [3]. Almost all studies proved that AI techniques show better performance than classical statistical techniques to predict bankruptcy [5], [6], [7], [8], [9], [10], [11].

Ensemble classifiers present outstanding ability to efficiently predict the bankruptcy of institutions [2], [12], [1], [13]. Since taking different perspectives in any decision-making problem is better than a single perspective, the ensemble techniques could boost their performance by combining varied machine learning classifiers with different characteristics [14]. Among these classifiers are Extreme gradient boosting (XGBoost) [9], Cluster-based Boosting (CBoost) technique [15], Adaptive Boosting (AdaBoost) [12], Bagging (BA) and Boosting (BO) [16]. While some studies use a single based classifier, other studies use multiple classifiers in the ensemble strategy to predict bankruptcy problem [17].

The developed model mainly focused on the machine learning classifiers to show its power in predicting bankruptcy problem. In this paper, two single based machine learning classifiers (Support Vector Machine (SVM), Decision Trees (DTs)) were applied to predict bankruptcy problem. Four ensemble classifiers (XGBoost, AdaBoost, Bagging and Categorical Boosting (CatBoost)) were applied to predict bankruptcy problem. The main aim of this study is to be able to efficiently classify the financial institutions to prevent bankruptcy. The datasets used are Polish enterprises dataset [18], Australian credit bankruptcy [19] and German enterprises dataset [20]. This paper is structured as follows: Section 2 presents a literature review from several studies; Section 3 presents the developed model for predicting bankruptcy; Section 4 presents the applied machine learning classifiers for data classification; Section 5 presents the performance evaluation of the developed model; Section 6 presents the experimental results; and Section 7 shows our conclusions and future work.

2. LITERATURE REVIEW

Recently, the bankruptcy problem has attracted the attentions of many studies. Bankruptcy prediction models created a common alert to avoid the consequences of the bankruptcy problem [21]. The pioneers in applying models to predict the bankruptcy problem were William H. Beaver (1966) [22], Altman (1968) [23] and Ohlson (1980) [24]. They made the early contributions in predicting the bankruptcy problem by using traditional statistical techniques [3]. After that, the supervised machine learning techniques efficiently proved a high performance in predicting the bankruptcy problem [13]. This section presents the most recent research papers in bankruptcy prediction. Its review data are from the web of science database. It presents the latest models in predicting bankruptcy based on machine learning techniques. Moreover, it presents the latest solutions to solve the problem of imbalanced datasets to improve prediction performance.

Various single based machine learning classifiers such as SVM [6, 25], DT [8, 26], Random Forest 11, 45]. They are applied to obtain a balanced dataset by having an equivalent number of minority and majority classes [46].

However, various studies proved that oversampling technique outperforms the under-sampling technique [47, 38, 48, 26, 49]. In 2002, Chawla,et al. [50] presented an improved (RF) [27], Genetic Algorithm (GA) [28] , Artificial Neural Network (ANN) [7, 29], Convolutional Neural Network (CNN) [30] and Anti Colony Optimization (ACO) model [31] show good performance ratio.

However, these studies could not determine the most superior model in predicting the bankruptcy problem because each model can prove a high performance based on the datasets used, financial ratios and situation [8]. From this perspective, integrating multiple single based machine learning classifiers to produce a robust classifier become the most superior technique in bankruptcy prediction. After that, researchers turn their studies to the ensemble classifiers to improve the performance of the prediction. They achieved a high-performance using ensemble classifiers.

BA and BO are the most popular ensemble classifiers used to improve the performance of the single machine learning classifiers [17]. They operate by collecting weak classifiers to generate a powerful ensemble classifier. The main difference between the BA and BO is the preparation of the training dataset [13]. The BA technique showed a good performance in many studies to predict bankruptcy problem [2, 16, 25]. The BO technique has many techniques that depend on it which improve an enhancement in predicting bankruptcy problem. The AdaBoost [14, 32, 33, 1], Gradient Boosting (GBoost) [34, 27], XGBoot [35, 36, 11, 37, 38], CatBoost [39-41] are the most commonly used BO techniques.

Recently, several studies are interested in re-sampling the dataset as a pre-processing step to avoid the imbalanced dataset problem [26]. The balanced datasets ensure a more reliable results while determining whether the financial institution will be bankrupt or not which brings a good perspective for the future in the financial market [42]. Among the data sampling strategies are the oversampling and the under-sampling. The re-sampling techniques (oversampling and under-sampling) are applied by many studies [15, 43, 44,

oversampling technique which is the Synthetic Minority Oversampling Technique (SMOTE). The SMOTE technique enhances the performance of random oversampling by generating non overlapped synthetic observations in the minority class [11]. After that, the SMOTE is integrated with the Tomek link to obtain a superior

performance in the re-sampling step which prevents missing important information in the testing dataset [49], [51].

3. DEVELOPED MODEL FOR PREDICTING BANKRUPTCY

3.1 Overview

The developed model in this paper consists of two main phases: the pre-processing phase and the application of machine learning techniques phase. The pre-processing phase includes applying different re-sampling techniques to balance the imbalanced selected datasets. The application of machine learning was used to predict the problem of bankruptcy. Three imbalanced datasets (Polish [18], Australian [19] and German credit [20] enterprises) were selected with different features from the UCI machine learning repository. Pre-processing the selected datasets is very important because the machine learning techniques cannot produce efficient results with imbalanced datasets. Constructing a model with an imbalanced dataset may be biased to the majority class which could mislead the classification results. Most of credit datasets have imbalanced ratio between bankrupt and non-bankrupt classes. The developed pre-processing phase ensures more efficient and robust results in predicting bankruptcy problem. Oversampling and under-sampling are the most common re-sampling techniques to solve the problem of imbalanced dataset. In this research, both oversampling and under-sampling approaches were applied to gain a better performance with the used datasets. Most credit datasets have missing values. Removing instances that have missing values could lead to losing useful information. The Polish and Australian datasets have missing values, which were filled during the pre-processing phase and before applying the re-sampling strategies. The selected datasets were split into training and testing datasets. In the training step, 80% of the dataset was used to build the machine learning classifiers. Hence, 20% of the dataset was used to test the prediction performance of the developed model. The training datasets were the input for the re-sampling step. Four re-sampling techniques (oversampling, SMOTE, under-sampling and SMOTETomek link) were applied to balance the training datasets.

The machine learning techniques were applied to classify the input training datasets to bankruptcy and non-bankruptcy classes. Six machine learning techniques were applied in the developed model to predict the bankruptcy problem. The applied machine learning classifiers were SVM, DT, BA, AdaBoost, XGBoost and CatBoost. The accuracy, AUC, precision and recall performance metrics were used to evaluate the developed model.

Figure 1 presents the framework of overall system.

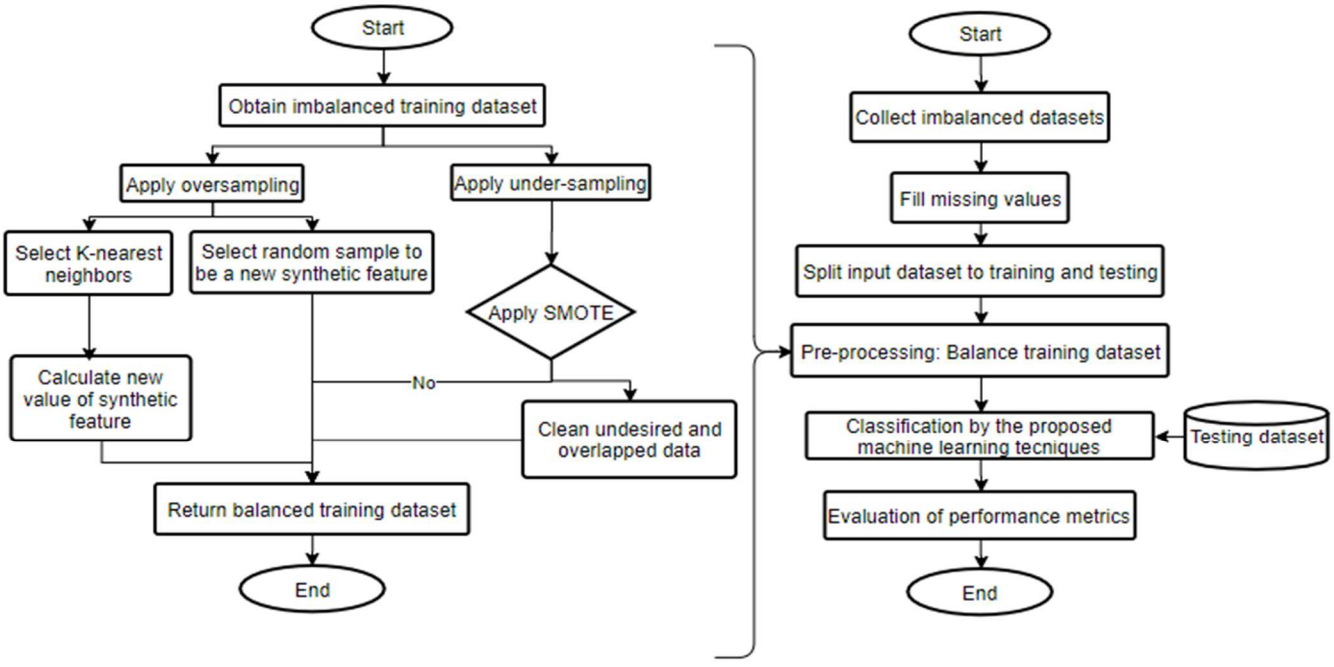


Figure 1: Framework of the developed model.

3.2 Pre-processing Techniques to Balance Datasets

In the economic domain, the non-bankrupt institutions (majority class) are much more than the bankrupt institutions (minority class). Thus, the imbalanced dataset problem misleads the classification results. So, the developed model balanced the datasets used for a better performance. The oversampling and under-sampling techniques were applied before using any machine learning classifier in the developed model. The developed model classified two classes: 0 for non-bankrupt and 1 for bankrupt.

3.2.1 Oversampling technique

The oversampling technique aims to create synthetic values in the minority class to have a good balancing ratio between minority and majority classes [46]. The balancing ratio will lead to the same number of instances of minority and majority classes [44]. The oversampling technique selects random samples from the minority class to be the new synthetic values [51]. However, this process makes a multiple copy of the samples in the minority class which could lead to the overfitting problem. The SMOTE technique is an advanced technique of the oversampling strategy to overcome the overfitting problem in the training dataset. The SMOTE technique was applied in this paper to enhance the performance of the oversampling strategy.

The SMOTE is widely used in many studies [21], [44], [46]. It depends on creating synthetic values which are similar in the feature space from the minority class. The pseudo code of the SMOTE algorithm to generate synthetic features is presented in ***Algorithm 1*** according to [48]. Firstly, SMOTE model randomly selects a value (stands for $\boldsymbol{x_i}$) from the minority class ($\boldsymbol{x_i}$ ϵ minority class). After that, it will find the k-nearest neighbors ($stands\ for\ \boldsymbol{k_i}$) to the random value ($\boldsymbol{x_i}$) [46]. It applies the Euclidian distance for each selected random value $\boldsymbol{x_i}$ to find $\boldsymbol{k_i}$. Then it will choose a random value ($\boldsymbol{x_j}$) within $\boldsymbol{k_i}$. The new synthetic feature (stands for $\boldsymbol{x_{new}}$) *in* the minority class, will be calculated as follow: $\boldsymbol{x_{new}} = \boldsymbol{x_i} + (\boldsymbol{x_j}\text{-}\boldsymbol{x_i}) \times \delta$, such that δ ϵ [0,1] is a random value to control the synthetic feature [52]. This powerful technique could create any required number of synthetic features to balance the data in the two classes. Figure 2 represents the execution of SMOTE technique based on the calculated Euclidian distance.

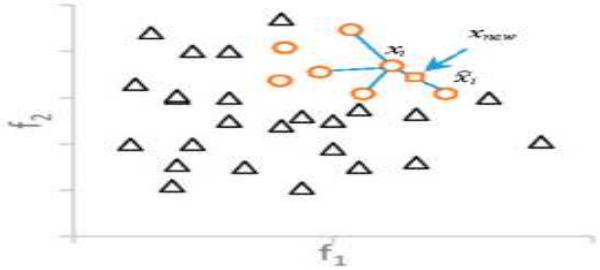


Figure 2: SMOTE technique based on the Euclidian distance.

Algorithm 1. SMOTE

Input: T: number of minority class samples; N: amount of SMOTE
N%; k: number of nearest neighbors; minority data D = $\mathbf{x_i}$ ϵ X, where i = 1, 2, 3, …, T.
Output: S: synthetic data S
N = (int)(N/100)
for i = 1 to T **do**
1. Find the k nearest (minority class) neighbors of $\mathbf{x_i}$

While N ≠ 0 **do**
1. Select one of the k nearest neighbors, x
2. Select a random number α ϵ [0,1]
3. x = $\mathbf{x_i}$+α (x-$\mathbf{x_i}$)
4. Append x to S
5. N = N – 1

end while
end for

3.2.2 Under-sampling technique

The main idea of under-sampling technique is to re-sample the majority class. Here in, deleting diverse values in the majority class to have a good balancing ratio by having the same number of samples in the two classes (majority and minority) [46]. Unfortunately, this technique may loss useful information in the majority class because of the high imbalanced ratios between the majority and minority classes [53]. Hence, some authors combine the oversampling and the under-sampling techniques to overcome the drawbacks in the under-sampling strategy [44]. The combined technique aims to apply the oversampling to increase number of minority class then apply under-sampling by reducing the number of majority class in order to produce two balanced classes.

The SMOTETomek link is another powerful re-sampling technique. It combines the oversampling SMOTE technique and the under-

sampling Tomek link [46]. Tomek link is used to eliminate undesired overlapping features to improve the performance of classification [21]. Tomek link is defined by the existence of two different data points: one of them belongs to the majority class (stands for$\mathbf{x_j}$), while the other belongs to the minority class (stands for $\mathbf{x_i}$). The Figure 3 presents the pre-processing step in the training dataset, either applying SMOTE alone or followed by under-sampling Tomek link.

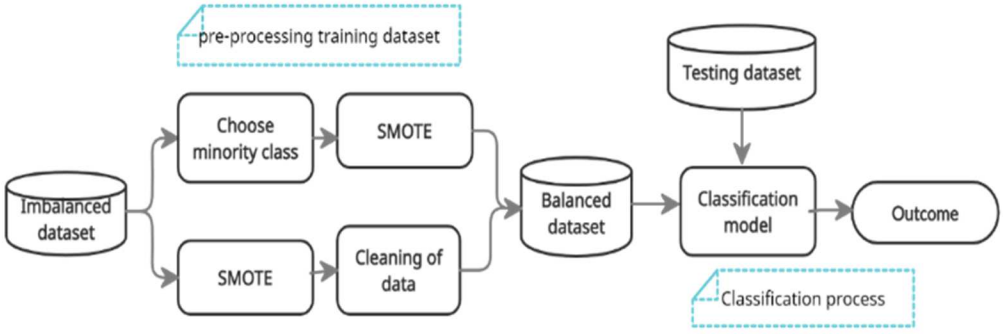


Figure 3: The SMOTE oversampling technique will be applied alone or followed by Tomek link to balance the imbalanced dataset.

4. MACHINE LEARNING FOR DATA CLASSIFICATION

Machine learning is an important application of AI. It makes a high contribution in classifications problems [25]. It shows high performance in bankruptcy prediction [3], [13], [12], [1]. In this paper, various machine learning techniques were applied to predict bankruptcy problem.

4.1 Support Vector Machine (SVM)

SVM is a supervised machine learning classifier. It is widely used in many fields such as banks failure to differentiate between two or more classes according to the target feature [47]. In this presented work, SVM was used as a binary classification model to separate between bankrupt and non-bankrupt classes. The main objective of the SVM is to classify input data by determining hyperplane with the highest margin which is the distance between the nearest point from each class and the hyperplane. The hyper plane is the boundary of classification between two classes (bankrupt and non-bankrupt) [29]. In case, the input training dataset is a linearly separable, then a linear hyperplane will be used to make classification boundary. The mathematical formula of linear SVM hyperplane for training dataset $\mathbf{x_i} \epsilon \mathbb{R}^{\mathbf{n}}$ (i= 1,2, 3, ..., n) and the target output is $y_i \epsilon$ [1, -1] as shown in equation (*1*).

$$\text{Hence H: } w^T (x) + b = 0 \qquad (1)$$

w is weight of n-dimensional vector space and b stands for bias [55]. Figure 4 presents the support vector machine model to differentiate input data points linearly into two classes by determining the hyperplane with the help of support vectors

Figure 4: Linear classification between 2 classes by the SVM with hyperplane. Support vectors are small number of data points which are used to determine hyperplane [8]. Support vectors

Euclidian distance between Xi and Xj is Ed($\mathbf{x_i}$, $\mathbf{x_j}$). The two data points $\mathbf{x_i}$ and $\mathbf{x_j}$ have a Tomek link if there exist data point $\mathbf{x_n}$ such that Ed($\mathbf{x_i}$, $\mathbf{x_n}$) < Ed($\mathbf{x_i}$, $\mathbf{x_j}$) or Ed($\mathbf{x_j}$, $\mathbf{x_n}$) < Ed($\mathbf{x_i}$, $\mathbf{x_j}$). Hence, one of these data points $\mathbf{x_i}$ and $\mathbf{x_j}$ is noisy or both of them are close to the border [54].

are located in the margin with dashed line. To define if the instance belongs to which y_i (1 or -1), it is classified as shown in equation ((2).

$$w^T(x) + b \begin{cases} \geq 1, & \text{for } y_i = +1 \\ \leq -1, & \text{for } y_i = -1 \end{cases} \quad (2)$$

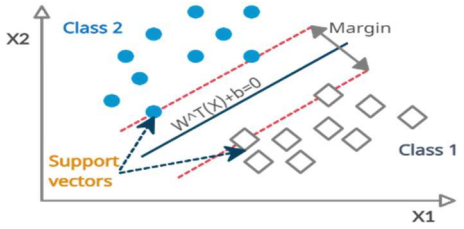


Figure 4: Linear classification between 2 classes by the SVM with hyperplane.

In non-linear separable features case, the SVM kernel function is used to map input point into high dimensional feature space. To classify data successfully the hyperplane should be with the highest margin to minimize the misclassification errors which are overfitting [56]. The multiplier α which is called Lagrange procedure is used to determine the hyperplane as shown in equation ((3) according to [55]:

$$\min_{w,b}\max_{\alpha} \left(\frac{1}{2}w^T w - \sum_{i=1}^{n} \alpha_i\{y_i[w^T x_i + b] - 1\}\right) \quad (3)$$

4.2 Decision Tree (DT)

DT is a supervised machine learning predictive classifier. DT classifier separates feature space by a recursive operation called divide and conquer, which produces prediction rules from training the dataset [29]. It starts with the topmost root node, and it transfers information of the samples to the branches which contain decision rules. Leaves are terminal nodes that contain the label of the target class (bankrupt or non-bankrupt). The remaining non-terminal inner nodes represent features, and the branches represent the deduction of the features to build the decision rules that lead to the final classification result [47]. Figure 5 presents the building structure of the DT [57]. The developed model applied the Gini index strategy to determine the split points. The Gini index technique is usually used to measure the efficiency of the separation which means if a random selected feature is classified correctly or not. Its probability lies between 0 and 1, 1 means purity of classification and 0 means random distribution of variables between classes. It is represented as shown in equation ((4) by [58].

$$G = \sum_{l=1}^{L} \hat{p}_{ml}\,(1 - \hat{p}_{ml}) \quad (4)$$

Here L means the number of observations and $\hat{p}_{ml}$ means if the proportion of observation l in partition Pm.

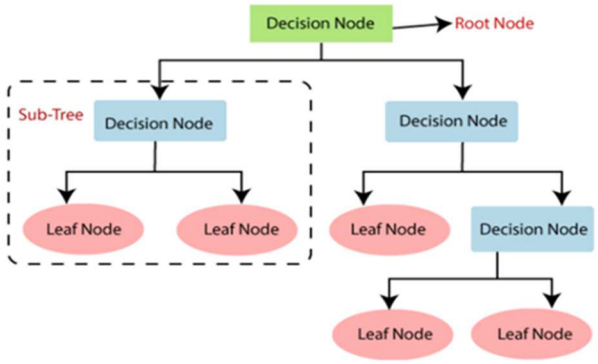


Figure 5: The building structure of the DT technique.

4.3 Ensemble Classifiers

The ensemble classifier integrates multiple weak learning classifiers with different hypotheses together for a better performance which is better than applying any integrated classifier alone [13]. It becomes a brilliant technique in the bankruptcy problem because it often shows a high performance than a single based classifier [59]. There are two strategies of ensemble classifier: parallel ensemble classifier and sequential ensemble classifier, as represented in

. The parallel ensemble classifier combines various classifiers to generate the desired ensemble classifier with different hypotheses from each classifier independently. However, in the sequential ensemble classifier, the first classifier tries to generate the desired model and the second classifier tries to minimize errors from the output of first classifier, and so on [14].

4.3.1 Bagging

BA model overcomes the overfitting problem. Moreover, it can increase the performance and the stability by integrating various and independent classifiers. It also stands for "bootstrap with variations from the given training dataset [13]. After that variations, different datasets are generated. However, some samples may be used more than once, while some other samples may not be used [2]. The ensemble classifier is generated by training single classifiers in parallel, then the output

aggregation" [25]. It combines various classifiers which are trained using bootstrap sampling. And hence, re-samples the samples uniformly

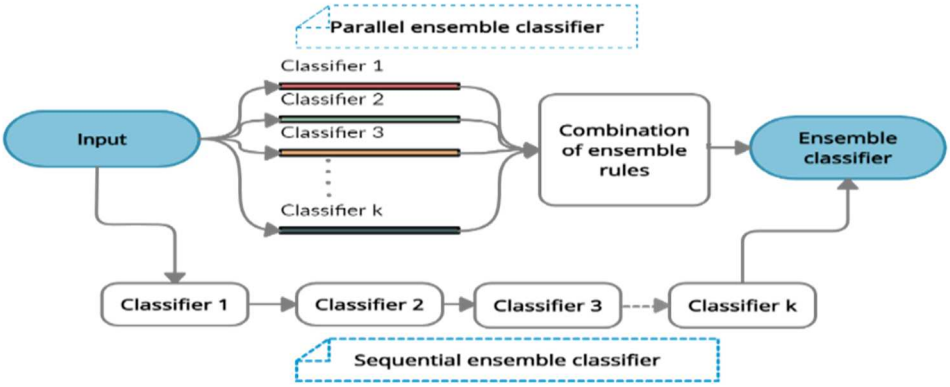


Figure 6: The structure of the parallel and sequential ensemble classifiers.

decision can be obtained by voting on most efficient learning methods. In this study, the BA technique was applied based on the random forest as a base classifier for BA.

4.3.2 Boosting

BO is a sequential ensemble classifier. It can reduce the overall misclassification error by using auxiliary models to avoid errors [9]. It combines weak classifiers with initial weight in the learning with modified dataset. It increases the weight of the misclassified samples, while decreases the weight of the correctly classified samples [2]. Each classifier focuses on the wrong classified instances from the previous classifier to fix misclassification cases and enhances the performance.

4.3.2.1 AdaBoost

AdaBoost technique is the earlier algorithm of BO ensemble classifier and is commonly used in the bankruptcy prediction problem. It depends on combining weak classifiers to produce a stronger classifier. The whole dataset is used to train each classifier in the AdaBoost technique. During each iteration, the AdaBoost makes a progress in the classification prediction by increasing the weight of the misclassified samples, while it decreases the weight of the correctly classified samples [45]. Each classifier on the AdaBoost has its own accuracy and score based on the used training dataset. Each classifier gives a vote when a new sample is trained. After that, the final class label is determined by the greatest number of votes from the combined classifiers [26]. The pseudo code of AdaBoost algorithm is shown in ***Algorithm 2*** [25].

The loop in step 2 presents the iterative operation of generating each tree and using the weights of instances from the previous iteration. Within the loop in step a, the classifier is fitted to the training dataset using the weighted instances. At every iteration, the tree uses the misclassified instances from the previous iteration. Within the loop in step b, the weighted error of the misclassified instances is calculated in the newest tree. Within the loop in step c, calculating the weight score of the newest tree using step b, this is an important step in combining all classifiers to determine the final label. Within the loop in step d, updating the weight of instances according to the fitting classifier. The final step 3 is used to sum the weights of all classifiers to determine final prediction [45].

4.3.2.2 XGBoost

The XGBoost is an advanced GBoost technique. It was recently proposed in 2016 by [35]. It appears outstanding performance and speed in many machine learning applications. XGBoost and GBoost techniques usually use a group of Classification and Regression Tress (CART). The XGBoost structures parallel DT, while GBoost structures sequential DT [11]. Let the given dataset D= $\{x_j , y_j\}$, where x_i is a sample of m features and $y_j$is the target label [14]. Both XGBoost and GBoost use additive function $f_n(x)$ to predict each sample in the given dataset. The prediction function is presented as shown inequation **(** (5)**.**

$$P_j = \sum_{j=1}^{N} f_n(x_j), f_n \epsilon \mathcal{F} \quad (5)$$

Where $f_n(x)$ indicates the prediction according to the N-th boost ($\mathcal{F}$ is the number of possible CARTs), and N refers to the number of samples in the training dataset. The XGBoost enhances the GBoost technique to improve the classification performance. It applies the objective function consists of loss function and a regularization term instead of applying a loss function only in the GBoost. The objective function is presented as shown in equation (

$$Obj(\Theta)=\sum_{j=1}^{N} \text{Ł}\ (y_j, P_j^{(r)}) + \sum_{r=1}^{R} \Omega\ (f_r) \quad (6)$$

) by [60].

$$Obj(\Theta)=\sum_{j=1}^{N} \text{Ł}\ (y_j, P_j^{(r)}) + \sum_{r=1}^{R} \Omega\ (f_r) \quad (6)$$

Algorithm 2. The pseudo code of the AdaBoost

1. Initialize the weights of instances $\mathbf{w_i}$= 1/N; i= 1,2,3, …, N
2. **For** t= 1 to T repeat steps from a to d
   a. Fit classifier $\mathbf{C_t}$ (**x**) to the training dataset using $\mathbf{w_i}$ (weights).
   b. Calculate weighted error of the newest DT $\mathbf{err_t} = \frac{\sum_{i=1.}^{N} \mathbf{w_i}\ \mathbf{I}\ (y_i \neq \mathbf{C_t}(\mathbf{x_i}))}{\sum_{i=1.}^{N} \mathbf{w_i}}$, I= 1 when error rate is acceptable, and 0 otherwise
   c. At each iteration calculate $\boldsymbol{\alpha_t}$= log[(1- $\mathbf{err_t}$)/ $\mathbf{err_t}$].
   d. Update weights for i= 1 to N $\mathbf{w_i}$= $\mathbf{w_i}$. exp [$\boldsymbol{\alpha_t}$.I($\mathbf{y_i} \neq \mathbf{C_t}(\mathbf{x_i})$)] and renormalized to $\mathbf{w_i}$ to sum to 1
3. **Output** C(x) = sign [$\sum_{t=1.}^{T} \boldsymbol{\alpha_t}\mathbf{C_t}(\mathbf{x})$]; The final classifier

Where Ł(.) represents the loss function, Ω(.) represents the regularization term and Θ is the structured CART.Moreover, R refers to the number of iterations of the XGBoost [11]. The regularization term $\sum \Omega(f)$ is an important to measure the complicity of XGBoost. Moreover, the regularizations term helps in avoiding the problem of overfitting by smoothing the weight of learners. The main objective of XGBoost technique is to detect the $f_r$that reduces the objective function [14]. The XGBoost uses the algorithm of the greedy search to optimize the objective function which is describe by [35]. The Taylor expansion formula is used with the objective function at iteration r to optimize the XGBoost technique as shown in equation **(**

$$Obj^{(r)}=\sum_{j=1}^{N} [\ q_j f_r\ (x_j) + \frac{1}{2}\ p_j f_r^{\,2}\ (x_j))\ ] + \sum_{r=1}^{R} \Omega\ (f_r)$$

(7)**.**

$$Obj^{(r)}=\sum_{j=1}^{N} [\ q_j f_r\ (x_j) + \frac{1}{2}\ p_j f_r^{\,2}\ (x_j))\ ] + \sum_{r=1}^{R} \Omega\ (f_r)$$

(7)

Where the first and second order of gradient statistics on the loss function, respectively are represented by q_j and p_j [27].

4.3.2.3 CatBoost

The CatBoost is a robust open-source library based on the GBoost technique, and it achieves a high performance in various machine learning applications [27]. It uses DT as a base ensemble classifier. Prokhorenkov, et al. [39] showed that CatBoost outperforms other GBoost techniques (XGBoost and Light GBoost). Prokhorenkov, et al. [39] introduced a solution to solve the prediction shift problem of the GBoost in the CatBoost technique to not use the same samples in the training process[40]. The CatBoost technique modifies the order boosting of the GBoost technique to solve the problem of the prediction shift [41]. Thus, the CatBoost firstly permutes the training samples randomly. The independent s+1 random permutations of the training samples are presented by $\sigma_1, \sigma_2, \ldots, \sigma_s$. The CatBoost constructs n models ($i = M_1, M_2, \ldots, M_n$) within each R iteration of the boosting. The first i samples in the random permutation are used to train the $M_i$model of the r^{th} iteration. Moreover, the M_i is used to calculate the gradients on the $(r+1)^{th}$ iteration of the i+1 samples [39]. The GBoost libraries except the CatBoost cannot deal with categorical data directly. So, they usually need a preprocessing step to convert categorical dataset to their target statistic (encode each categorical feature to a numerical value). Another main function of the catBoost that CatBoost can handle categorical features without need to a preprocessing step and keep the most information without loss [37]. The CatBoost constructs a balanced (symmetric) DT with the same structure at each level to avoid overfitting problem on handling categorical features by obtain multiple labels for the same category. So that, the CatBoost do random permutations to have different datasets to compute the numerical feature without overfitting problem. At each permutation, the information of the samples before i are used to calculate the target statistics of sample i. After using several permutations to calculate the target statistics of the feature, the averaged value for each sample is computed to be the final target [41].

Table 1: Meta-parameters for each applied machine learning classifier.

Classifier	Parameters
SVM	Linear SVM and penalty parameter C = 1
DT	Criterion = Gini index tree with maximum depth = 3, random state = 100 and Min. samples in leaf = 5
BA	Base learner is RF with random state = 42, the number of estimators = 100, bootstrap=True and number of jobs = -1
AdaBoost	Base learner is DT with maximum depth = 3, random state = 100 and Min. samples in leaf = 5
XGBoost	Base learner is CART with Maximum depth of trees = 3, estimators =300, random state = 0 and learning rate = 1.0
CatBoost	Base learner is DT with random state = 42, estimators = 100 and verbose=0

5. PERFORMANCE EVALUATION

The efficiency of the used machine learning classifiers was validated on three popular datasets using various performance measurements. To assure a reliable performance of the machine learning classifiers, the datasets were selected to have varying number of features and samples. The developed model used all features of the selected datasets. The selected datasets, evaluation metrics

All the experimental results were obtained using Python 3.8, on a PC with 2.6 GHz, Intel CORE i7, 8 GB RAM and NVIDIA GEFORCE GTX 1650 Max-Q using Windows 10 operating system. **Error! Reference source not found.** presents the setting of the parameters for each machine learning classifier in the developed mode.
and the achieved results are presented in the following subsections.

5.1. Datasets Used

Three imbalanced datasets were used from the UCI machine learning repository to measure the performance of the developed model. The datasets used are Polish enterprises dataset [18], Australian credit dataset [19] and German credit dataset [20]. The polish dataset is a large dataset that consists of 5 years of real-world data from the financial markets. It lists the bankrupt institutions from 2000 to 2012 and still working institutions from 2007 to 2013 [25]. The large training dataset helps us to ensure a good and reliable performance. Hence, the five years were grouped in one file to have 43,405 records as applied in [31]. **Error! Reference source not found.** presents the description of the three datasets used.

Table 2: The description of the datasets used.

Properties/ Dataset	Polish	Australian	German
Attributes	64	14	24
Instances	43,405	690	1,000
Bankrupt institutions	2,091	383	300
Non-Bankrupt institutions	41,314	307	700
Missing values	Yes	Yes	No

To replace the missing values in the Polish and Australian datasets, the average value for column of each attribute were calculated and the missing cell were replaced with the calculated average value.

5.2. Metrics

There are various performance measurements to evaluate the model's performance of prediction [5]. In this paper, four performance measurements were used, which are accuracy, AUC, recall and precision. The developed model depends on the four performance measures for more accurate and robust measurements.

5.2.1. Accuracy

The accuracy is the most commonly used measurement in evaluating the performance of models. However, the accuracy metric is misled measure to evaluate machine learning classifiers [37]. The accuracy is the percentage of the correct predicted labels. The overall accuracy is the ratio of the true predicted labels to the total number of the predicted labels,whether correct or wrong labels [61].

5.2.2. Precision

The precision indicates the sharpness of the model not to classify non-bankrupt institution as bankrupt institution [34]. It is calculated by the ratio of the number of correct cases to the total number of cases of bankrupt and non-bankrupt. It is calculated for each class (bankrupt and non-bankrupt).

5.2.3. AUC

The AUC ratio is widely used with binary classification problems [14]. Most studies depend on the AUC ratio as a robust measurement when comparing models. It reflects the ability of the model to distinguish between bankrupt and non-bankrupt cases. Its value lies between 0 and 1 [9]. If the value of the AUC is between 0.9 and 1 refers to outstanding performance, good performance is between 0.7, and 0.9 and inferior performance is less than 0.7 [37]. Equation ($AUC = \frac{precision\ for\ non-bankrupt + precision\ for\ bankrupt}{2}$ (8)) shows the formula of the AUC measure.

$$AUC = \frac{precision\ for\ non-bankrupt + precision\ for\ bankrupt}{2} \quad (8)$$

5.2.4. Recall

Recall is known as a sensitivity value. It is used to determine the total number of correct results and should be assessed for each class [11].

Figure 7 presents the confusion metric of the four used performance measurements which are accuracy, AUC, precision, and recall.

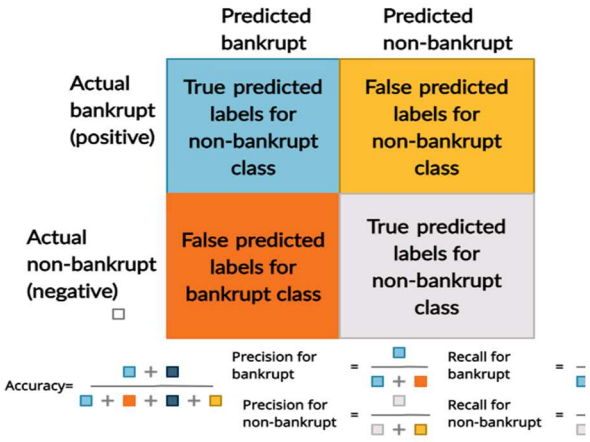


Figure 7: The confusion matrix of the presented performance measurements (accuracy, AUC, precision and recall).

6. RESULTS AND DISCUSSION

This section presents the results of our experiments. The selected datasets were split into training and testing datasets to measure the performance of the developed model as recommended by [37]. The random state used with the oversampling, SMOTE and under-sampling techniques was 42. However, a random state equal to zero was used with the SMOTETomek link. The random states in this paper were chosen to achieve high-performance with the selected datasets. As we noticed that the value of the random state could change the performance of the models in some datasets.

The developed model focused on the precision ratio of the non-bankrupt institutions to make sure that non-bankrupt is not classified as bankrupt. Moreover, this paper presented only the recall ratio of the bankrupt institutions to know the correct number of the classified bankrupt. A comparison between the applied machine learning classifiers with the selected re-sampling strategies across the selected datasets were made. As mentioned before, the accuracy measurement alone is not enough with binary classification problems. For this reason, the model with the highest performance was determined based on four performance measurements. The experimental results of the developed model are shown in Table 3-Table 6. The technique which shows the highest performance measurements according to the used performance metrics is highlighted in underline and bold. Table 3 reveals the evaluation measures of the developed model with the oversampling strategy across the selected datasets.

Table 3: The performance measurements of the developed techniques across selected datasets after applying oversampling strategy.

Datasets	Technique	Performance measures			
		Accuracy	AUC	Precession	Recall
Polish	SVM	0.257	0.658	0.97	0.88
	DT	0.807	0.823	0.98	0.68
	BA	0.965	0.948	0.97	0.42
	AdaBoost	0.932	0.913	0.99	0.75
	XGBoost	0.968	0.932	0.98	0.67
	CatBoost	**0.967**	**0.945**	**0.99**	**0.72**
Australian	SVM	0.862	0.903	0.91	0.86
	DT	0.862	0.894	0.89	0.80
	BA	0.877	0.925	0.91	0.84
	AdaBoost	0.797	0.856	0.83	0.71
	XGBoost	0.870	0.889	0.88	0.78
	CatBoost	**0.877**	**0.926**	**0.91**	**0.84**
German	SVM	0.72	0.800	0.88	0.78
	DT	0.74	0.770	0.80	0.49
	BA	0.775	0.825	0.84	0.61
	AdaBoost	0.745	0.719	0.83	0.59
	XGBoost	**0.815**	**0.834**	**0.86**	**0.66**
	CatBoost	0.795	0.832	0.87	0.71

The presented results used the random selection of samples in the minority class, which has increased the balancing ratio between the bankrupt and non-bankrupt classes. With the Polish and Australian datasets, the CatBoost technique showed the highest performance, which was followed by the XGBoost. Nevertheless, in terms of accuracy and AUC, the XGBoost slightly outperformed the CatBoost with the German dataset. In this developed model, the GBoost techniques (XGBoost and CatBoost) proved high predictive results in the bankruptcy problem with oversampling as demonstrated by [27].

The oversampling strategy produces larger number of samples, which increases the training ability of the XGBoost and CatBoost techniques. Such large number of samples creates well-structured boosting trees in XGBoost and CatBoost techniques, which increases their ability to differentiate between bankrupt and non-bankrupt institutions without overfitting.

Some additional observations are also worth mentioning. The performance of the other techniques (BA, AdaBoost and DT) is not bad, but still not the best. For example, the performance of the BA technique is very close to the CatBoost with the Australian dataset. The BA and CatBoost accuracy and bad AUC ratio. SVM could not determine the perfect hyperplane, due to the large number of features in the Polish dataset. Nevertheless, SVM proved better performance with small dataset (Australian and German) across small number of features than large dataset (Polish).

After applying the oversampling technique, the SMOTE as oversampling strategy were applied to enhance the performance. Table 4 presents the performance measurements of the SMOTE across the selected datasets.

Table 4: The performance measurements of the developed techniques across selected datasets after applying SMOTE strategy.

Datasets	Technique	Performance measures			
		Accuracy	AUC	Precession	Recall
Polish	SVM	0.256	0.658	0.97	0.88
	DT	0.764	0.823	0.98	0.76
	BA	0.962	0.934	0.97	0.49
	AdaBoost	0.949	0.917	0.98	0.64
	XGBoost	0.965	0.932	0.98	0.64
	CatBoost	**0.970**	**0.954**	**0.98**	**0.66**
Australian	SVM	0.870	0.898	0.86	0.75
	DT	0.841	0.894	0.93	0.90
	BA	0.870	0.921	0.90	0.82
	AdaBoost	0.790	0.838	0.85	0.76
	XGBoost	0.891	0.895	0.89	0.80
	CatBoost	**0.877**	**0.928**	**0.90**	**0.82**
German	SVM	0.745	0.769	0.82	0.58
	DT	0.63	0.723	0.81	0.66
	BA	0.805	0.828	0.85	0.63
	AdaBoost	0.755	0.716	0.82	0.54
	XGBoost	0.80	0.824	0.86	0.68
	CatBoost	**0.795**	**0.844**	**0.84**	**0.61**

made many versions of the training samples to overcome the overfitting problem on the Australian dataset, which has small number of features. As regards the SVM technique on the Polish dataset, despite its high precision and recall ratio, it has a very low

Using the SMOTE together with the CatBoost technique showed the highest performance measurement across all selected datasets. So, it is obvious that the CatBoost showed better performance with the SMOTE more than with the random selection of samples on the German dataset. This is because the SMOTE technique applies the K-nearest neighbors with K= 1, which enhances the performance over the random selection of samples used in the original oversampling technique. The SMOTE enhanced the accuracy and AUC ratio of the CatBoost across the selected datasets, nevertheless, it reduces the precision and recall ratio. This finding matches with [40] who proved that CatBoost with default parameters is better that XGBoost with tuned parameters across various datasets.

Datasets	Technique	Performance measures			
		Accuracy	AUC	Precession	Recall
Polish	SVM	0.201	0.648	0.97	0.91
	DT	0.795	0.821	0.98	0.70
	BA	0.817	0.911	0.99	0.85
	AdaBoost	0.844	0.921	0.99	0.83
	XGBoost	0.862	0.933	0.99	0.83
	CatBoost	**0.880**	**0.955**	**0.99**	**0.87**
Australian	SVM	0.870	0.895	0.89	0.80
	DT	0.862	0.889	0.89	0.80
	BA	**0.870**	**0.922**	**0.90**	**0.82**
	AdaBoost	0.797	0.854	0.84	0.73
	XGBoost	0.877	0.888	0.89	0.80
	CatBoost	0.870	0.920	0.89	0.80
German	SVM	0.70	0.795	0.89	0.81
	DT	0.695	0.784	0.93	0.88
	BA	0.73	0.816	0.89	0.80
	AdaBoost	0.62	0.654	0.77	0.53
	XGBoost	0.71	0.781	0.87	0.76
	CatBoost	**0.745**	**0.822**	**0.89**	**0.78**

Some additional observations are also worth mentioning. The performance of the SVM technique on the Polish dataset does not change by applying the SMOTE. But the SMOTE only improved the accuracy of the SVM technique on the Australian and German datasets. The SMOTE improved only the recall ratio of the DT technique on the Polish dataset. The SMOTE has also improved the precision and recall ratio of the DT on the Australian and German datasets,

nevertheless, it minimized the accuracy. There is no doubt that the improvement in the precision and recall ratio means the more reliable results. The SMOTE improves the performance measurement of the BA technique on the German dataset, nevertheless, it reduces the performance on the Australian dataset. Moreover, it reduces the AUC of the BA technique on the Polish dataset, also improves the recall ratio. The SMOTE improves the accuracy and AUC but reduces the precision and recall ratio of the AdaBoost technique on the Polish dataset, nevertheless, the opposite is observed on the Australian dataset. The SMOTE also improved the accuracy of the AdaBoost on the German dataset. The XGBoost with the original oversampling is slightly better than SMOTE on the Polish dataset. The SMOTE also improved the accuracy and reduces the recall ratio of the XGBoost on the Polish dataset. Moreover, it improved the performance measurement of the XGBoost on the Australian dataset. It minimized the accuracy and AUC ratio of the XGBoost on the German dataset, nevertheless, it improved the recall ratio.

It is obvious that the accuracy of the most models using the under-sampling strategy is very less than the accuracy using the oversampling strategies.

With the Polish dataset, the developed machine learning models have bad performance in terms of AUC ratio except BO techniques (AdaBoost, XGBoost and CatBoost). The BO techniques with the largest dataset used (Polish) across under-sampling can have less misclassification errors by adding auxiliary methods with few samples. The performance of the developed models was remarkably bad with under-sampling in terms of accuracy compared with oversampling. With the under-sampling, the dataset used lost large number of samples in the majority class, while most credit datasets have few numbers of bankrupt institutions. So, the large Polish dataset with under-sampling has a better recall ratio and same precision ratio.

With the Australian dataset, the developed models showed a bad performance in terms of AUC ratio with under-sampling more than with oversampling. The presented models only showed better performance in terms of accuracy with SVM and XGBoost. The SVM has a better performance in terms of accuracy with Australian dataset, because Australian has a small number of features and has continuous and nominal attributes. The regularization term of the objective function of the tuned XGBoost helped it to improve its performance in terms of accuracy. The developed models with under-sampling on Australian dataset in terms of precision showed a slight difference more than with oversampling. The developed models showed a bad performance in terms of recall with under-sampling on Australian dataset except XGBoost and AdaBoost. The BA technique outperformed the developed models with under-sampling and showed a high ability to overcome overfitting problem with RF as a base classifier. The performance of the CatBoost is below the performance of BA with a slight difference of 0.002%.

After applying the SMOTE technique, the under-sampling strategy were applied to show its performance.

Table 5 presents the performance measurements of the under-sampling strategy across the selected datasets.

Table 5: The performance measurements of the developed techniques across selected datasets after applying under-sampling strategy.

With the German dataset, the developed models showed a remarkable low performance in terms of accuracy and AUC ratio using the under-sampling technique more than oversampling. Nevertheless, the developed models improved the performance in terms of precision and recall ratio using under-sampling except AdaBoost. In binary classification bankruptcy problem, the under-sampling did not improve accuracy and AUC, which are the two significant performance measures. Because the under-sampling strategy lost much information from the majority class, so the number of samples used for training became small. The CatBoost presented the best performance on the Polish and German datasets, because they have large number of features and CatBoost randomly permutes the training

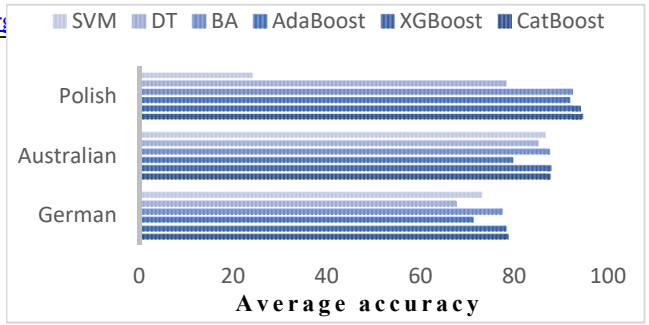
samples. So, the CatBoost has various training datasets and has a good ability for training.

After applying the under-sampling technique, the SMOTETomek link strategy were applied as a combination of oversampling and under-sampling strategies. Table 6 presents the performance measurements of the SMOTETomek link strategy across the selected datasets.

Table 6: The performance measurements of the developed techniques across selected datasets after applying SMOTETomek link strategy.

Datasets	Technique	Performance measures			
		Accuracy	AUC	Precession	Recall
Polish	SVM	0.252	0.658	0.97	0.89
	DT	0.767	0.782	0.98	0.72
	BA	0.952	0.936	0.98	0.57
	AdaBoost	0.951	0.920	0.98	0.61
	XGBoost	0.966	0.941	0.98	0.64
	CatBoost	**0.969**	**0.945**	**0.98**	**0.66**
Australian	SVM	0.862	0.901	0.94	0.90
	DT	0.841	0.912	0.92	0.88
	BA	**0.884**	**0.920**	**0.91**	**0.84**
	AdaBoost	0.804	0.847	0.83	0.71
	XGBoost	0.876	0.891	0.91	0.84
	CatBoost	0.884	0.919	0.91	0.84
German	SVM	0.76	0.772	0.83	0.58
	DT	0.64	0.733	0.92	0.88
	BA	0.79	0.825	0.85	0.63
	AdaBoost	0.73	0.685	0.81	0.54
	XGBoost	**0.805**	**0.840**	**0.88**	**0.73**
	CatBoost	0.81	0.844	0.85	0.63

With the Polish dataset, the SMOTETomek link proved an enhancement of the developed models in terms of accuracy and AUC more than under-sampling. As a performance measurement, this work considers the AUC ratio rather than the accuracy when comparing the different models, because the AUC is more reliable than the accuracy in binary classification problems. Hence, the presented results are very logical. Because the under-sampling may remove some useful information from the majority class, which results in small training dataset. So, the accuracy of the models will be better but the AUC ratio, which is more reliable and robust, will be minimized. Thus, the SMOTETomek link is presented to get rid of the un-reliable results of the under-sampling by linking the oversampling with under-sampling Tomek link. Most of the developed models presented the same performance in terms of precision with oversampling,



under-sampling and SMOTETomek link and some models had only 0.01% difference. However, the developed models presented a low recall ratio with SMOTETomek link more than under-sampling. The oversampling showed better performance of the developed models in terms of accuracy and AUC ratio more than SMOTETomek link except AdaBoost and XGBoost. Only the BO techniques improved the performance in terms of recall with SMOTETomek link more than oversampling. The CatBoost with SMOTETomek link outperformed the presented models followed by XGBoost then AdaBoost.

With the Australian dataset, the SMOTETomek link improved the developed model in terms of accuracy and AUC more than under-sampling. The SMOTETomek link improved only the performance of the DT and XGBoost in terms of accuracy and AUC ratio more than oversampling. Nevertheless, the SMOTETomek link improved the performance of the developed models in terms of precision and recall more than oversampling and under-sampling. The high precision and recall ratio present most of correctly labeled results in the testing step. The BA technique outperformed the developed models with SMOTETomeklink followed by CatBoost with a slight difference 0.001%.

With the German dataset, the SMOTETomek link improved only the developed ensemble model in terms of accuracy and AUC more than oversampling and under-sampling. Nevertheless, the SMOTETomek link minimized the performance of the developed models in terms on precision and recall ratio more than under-sampling except AdaBoost. However, the SMOTETomek link improved only the developed models in terms of precision and recall more than oversampling except SVM, AdaBoost and

ISSN: 1992-8645 www.jatit.org

CatBoost. The XGBoost technique outperformed the developed models with SMOTETomek link followed by CatBoost.

The overall classification performance of the applied techniques across the four applied re-sampling techniques was illustrated in terms of average accuracy and average AUC ratio in **Error! Reference source not found.** and

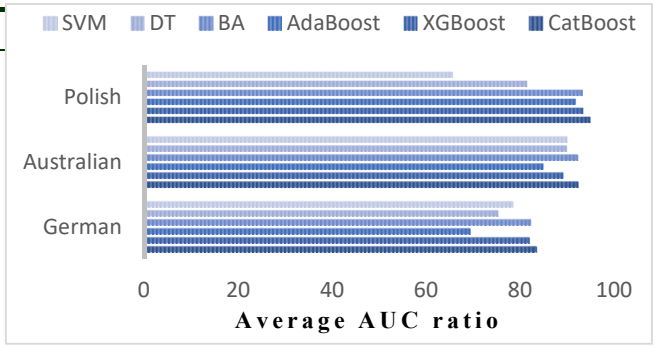


Figure 8: Comparison between the presented techniques in terms of average accuracy.

Figure 9, respectively. **Error! Reference source not found.** and

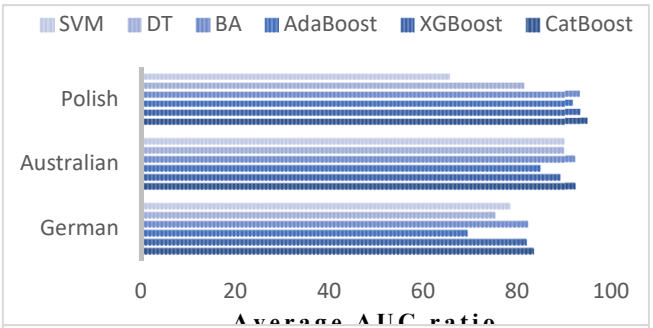


Figure 8: Comparison between the presented techniques in terms of average accuracy.

Figure 9 include the classification performance across the three datasets used.

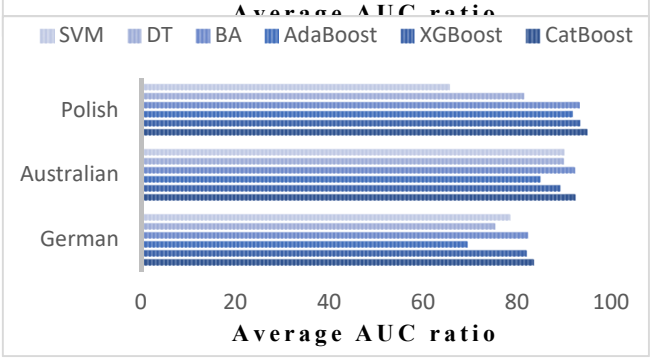


Figure 8: Comparison between the presented techniques in terms of average accuracy.

Figure 9: Comparison between the presented techniques in terms of average AUC ratio.

Error! Reference source not found. and

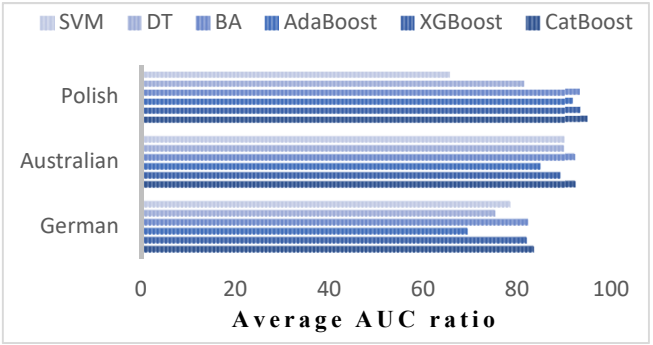


Figure 8: Comparison between the presented techniques in terms of average accuracy.

Figure 9 illustrate that CatBoost technique showed the better experimental results than other applied techniques across the selected datasets. Also, **Error! Reference source not found.** and

ISSN: 1992-8645 www.jatit.org

Figure 8: Comparison between the presented techniques in terms of average accuracy.

Figure 9 illustrate that the overall classification performance of the applied techniques is influenced by the size of the dataset used. It is obvious that the BA and BO techniques have a better performance with the large datasets. Also, this study noticed that the SVM and DT techniques work better with small datasets than large datasets. The experimental outcomes in **Error! Reference source not found.** and

Figure 8: Comparison between the presented techniques in terms of average accuracy.

Authors	Technique	Accuracy	AUC	Feature selection
Xia, et al. (2017) **[14]**	SVM	85.54	0 .625	√
	DT	84.51	0 .606	
	BA	86.42	0 .630	
	AdaBoost	85.64	0 .620	
	XGBoost	87.38	0 .647	
Xia, et al. (2020) **[27]**	SVM	85.78	0.898	√
	DT	82.14	0.817	
	XGBoost	86.82	0.936	
	CatBoost	87.07	0.936	
Bentéjac, et al. (2020)**[41]**	XGBoost	86.94	-	χ
	CatBoost	87.54	-	
Zhang, et al. (2021) **[12]**	AdaBoost	81.15	0.898	√
	XGBoost	84.05	0.933	

Authors	Technique	Accuracy	AUC	Feature selection
Xia, et al. (2017) **[14]**	SVM	76.07	0 .271	√
	DT	72.65	0 .129	
	BA	75.19	0 .223	
	AdaBoost	73.61	0 .214	
	XGBoost	76.85	0 .297	
Xia, et al. (2020) **[27]**	SVM	70.67	0.708	√
	DT	68.92	0.636	
	XGBoost	77.02	0.795	
	CatBoost	77.63	0.802	
Bentéjac, et al. (2020) **[41]**	XGBoost	79.00	-	χ
	CatBoost	77.30	-	
Zhang, et al. (2021) **[12]**	AdaBoost	75.16	0.765	√
	XGBoost	74.00	0.763	

[12] in terms of accuracy and AUC with the German dataset.

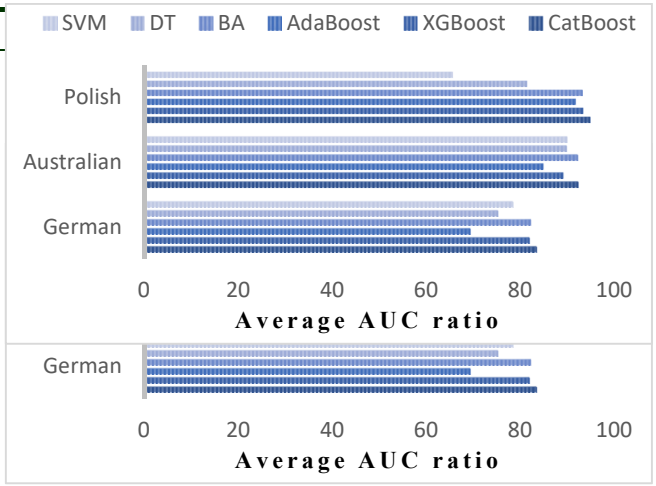


Figure 9 proved that the classification results with re-sampling techniques are highly affected. It is notice that applying re-sampling techniques enhances the overall classification performance of the applied techniques with more reliable results. Some previous studies were presented in Table 7 and Table 8 using some applied techniques in this study. All these previous studies do not apply any re-sampling technique with using imbalanced datasets. Table 7 compares between thedeveloped model and these previous studies [14], [27], [41] and [12] in terms of accuracy and AUC with the Australian dataset. Table 8 compares between the developed model and these studies [14], [27], [41] and

Table 7: Performance evaluation across the Australian dataset.

Table 8: Performance evaluation across the German dataset.

The developed model has an enhancement in the performance with the used re-sampling strategies on the Australian and German datasets more than [14]. Moreover, the developed model achieved an enhancement by applying SVM and DT on both Australian and German datasets more than [27]. With the German dataset, the developed model showed a better performance by XGBoost with the used re-sampling strategies except with under-sampling more than [27]. The CatBoost with the developed model showed a better performance with the Australian dataset more than [27], however the contradiction

performance was happened with the German dataset. XGBoost and CatBoost were applied by [41] with the imbalanced dataset. The common performance metric between this study and [41] is only accuracy. The developed model improved the accuracy with the used re-sampling strategies more than [41] on the Australian dataset. The developed model improved the accuracy with the used re-sampling strategies except with under-sampling on the German dataset.

In terms of accuracy and recall, Table 9 presents the comparison between this study and [31] with the Polish, Australian and German datasets. The developed model by [31] did not balance the imbalanced datasets used.

Table 9: The performance measurement across the Polish dataset.

With the Polish dataset, the developed model improved the accuracy of BA more than [31]. The AdaBoost has the same performance in terms of accuracy in the developed model with SMOTE and SMOTETomek link such as [31].
With the Australian dataset, the developed model showed a better performance with using the SVM, DT and BA more than [31].

With the German dataset, the SVM technique with the developed model showed a better performance with the oversampling and under-sampling more than [31]. The DT technique showed a better performance with under-sampling and SMOTETomek link more than [31]. The BA technique showed a better performance with only under-sampling more than [31]. The AdaBoost technique improved only the accuracy with oversampling and SMOTE more than [31].

7. CONCLUSIONS AND FUTURE WORK

In this paper, a critical issue for the financial market which is the bankruptcy problem was discussed. This developed work attempted to find powerful classification model to predict the bankruptcy problem. The machine learning techniques showed powerful performance in many studies. So, several machine learning techniques were applied in this paper to predict the bankruptcy problem. This paper presented empirical analysis of some single-based machine learning classifiers and ensemble-based machine learning classifiers (AdaBoost, BA, XGBoost and CatBoost) to predict the bankruptcy problem. The presented results were based on three selected datasets (Polish, Australian and German companies). This developed model made a comparison between the applied techniques based on the selected datasets in terms of some performance measurements. Most of the available datasets are highly imbalanced datasets, hence, the machine learning classifiers cannot produce a reliable result from these imbalanced datasets. Due to the imbalanced dataset problem, the developed model solved the imbalanced dataset problem before applying any machine learning technique to ensure more reliable results. This is achieved using re-sampling techniques, which produce good balanced distribution ratio between the classes. This study used the application of four re-sampling techniques to balance the selected datasets. In most cases, the results showed that

Authors	Dataset	Technique	Accuracy	Recall
J. Uthayakumar1, et al. (2020) **[31]**	Polish	SVM	95.18	95.18
		DT	95.18	95.18
		BA	95.16	95.20
		AdaBoost	95.14	95.38
	Australian	SVM	84.92	77.96
		DT	83.47	80.44
		BA	84.05	81.67
		AdaBoost	85.65	78.88
	German	SVM	73.2	74.65
		DT	72.4	77.31
		BA	74.6	78.3
		AdaBoost	74.59	75.20

ensemble classifiers enhance the performance than single-based classifier on the selected datasets. Furthermore, the single-based classifiers showed ineffective performance with small datasets such as the Australian and German datasets compared to large datasets such as the Polish dataset.

The empirical experimental results showed that the best model with the Polish dataset consists of handling the missing values, then re-sample the training dataset using SMOTE technique followed by CatBoost classification technique. The CatBoost classifier with SMOTE showed high ability in handling categorical datasets to overcome the overfitting problem, which is the case with polish dataset. Moreover, the empirical results showed that the best model with Australian dataset consists of handling and filling missing values, then re-sample training dataset using SMOTETomek link technique followed by BA classification technique. Moreover, the empirical results showed that the best model with German dataset consists of re-sample training dataset

using oversampling technique followed by XGBoost classification technique. The experimental results showed that the performance of the classification techniques on the datasets differs from one re-sampling technique to another. The results proved the efficiency of oversampling techniques than under-sampling techniques with various datasets. This study is based on datasets that have been used in several studies. It showed that the efficiency of each machine learning-based model depends on the size and features of the used datasets and the applied pre-processing steps. The developed system depends on all the features of the used datasets, so it will not work well with only some of the features. Moreover, it does not perform well with small size datasets and without the suggested pre-processing steps. Each model shows its strength based on the dataset used and the applied re-sampling technique. The proposed future work attempts to depend on larger datasets to ensure the presented results in this paper.

REFERENCES:

[1] Du Jardin P., *Forecasting corporate failure using ensemble of self-organizing neural networks.* European Journal of Operational Research, 2021. **288**(3): p. 869-885.https://doi.org/10.1016/j.ejor.2020.06.020

[2] Abellán J. and Castellano J. G., *A comparative study on base classifiers in*

[7] Business and Finance, 2018. **44**: p. 16-25.https://doi.org/10.1016/j.ribaf.2017.07.104

[8] Alaka H. A., Oyedele L. O., Owolabi H. A., Kumar V., Ajayi S. O., Akinade O. O., and Bilal M., *Systematic review of bankruptcy prediction models: Towards a framework for tool selection.* Expert Systems with Applications, 2018. **94**: p. 164-184.https://doi.org/10.1016/j.eswa.2017.10.040

[9] Carmona P., Climent F., and Momparler A., *Predicting failure in the U.S. banking sector: An extreme gradient boosting approach.* International Review of Economics & Finance, 2019. **61**: p. 304-323.https://doi.org/10.1016/j.iref.2018.03.008

[10] Gregova E., Valaskova K., Adamko P., Tumpach M., and Jaros J., *Predicting Financial Distress of Slovak Enterprises: ensemble methods for credit scoring.* Expert Systems with Applications, 2017. **73**: p. 1-10.https://doi.org/10.1016/j.eswa.2016.12.020

[3] Cao Y., Liu X., Zhai J., and Hua S., *A two-stage Bayesian network model for corporate bankruptcy prediction.* International Journal of Finance & Economics, 2020.https://doi.org/10.1002/ijfe.2162

[4] Lemma W. Senbet T. Y. W., *Corporate Financial Distress and Bankruptcy: A Survey.* Foundations and Trends in Finance , 2013. **5**(4): p. 243-335.https://doi.org/10.2139/ssrn.2034646

[5] Liang D., Lu C.-C., Tsai C.-F., and Shih G.-A., *Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study.* European Journal of Operational Research, 2016. **252**(2): p. 561-572.https://doi.org/10.1016/j.ejor.2016.01.012

[6] Antunes F., Ribeiro B., and Pereira F., *Probabilistic modeling and visualization for bankruptcy prediction.* Applied Soft Computing, 2017. **60**: p. 831-843.https://doi.org/10.1016/j.asoc.2017.06.0437.Le H. H. and Viviani J.-L., *Predicting bank failure: An improvement by implementing a machine-learning approach to classical financial ratios.* Research in International

Comparison of Selected Traditional and Learning Algorithms Methods. Sustainability, 2020. **12**(10).https://doi.org/10.3390/su12103954

[11] Dastile X., Celik T., and Potsane M., *Statistical and machine learning models in credit scoring: A systematic literature survey.* Applied Soft Computing, 2020. **91**.https://doi.org/10.1016/j.asoc.2020.106263

[12] Zhang W., Yang D., Zhang S., Ablanedo-Rosas J. H., Wu X., and Lou Y., *A novel multi-stage ensemble model with enhanced outlier adaptation for credit scoring.* Expert Systems with Applications, 2021. **165**.https://doi.org/10.1016/j.eswa.2020.113872

[13] Chen Z., Chen W., and Shi Y., *Ensemble learning with label proportions for bankruptcy prediction.* Expert Systems with Applications, 2020.

146.https://doi.org/10.1016/j.eswa.2019.113155

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

[15] Le T., Hoang Son L., Vo M., Lee M., and Baik S., *A Cluster-Based Boosting Algorithm for Bankruptcy Prediction in a Highly Imbalanced Dataset.* Symmetry, 2018. **10**(7).https://doi.org/10.3390/sym10070250

[16] Lin W.-C., Lu Y.-H., and Tsai C.-F., *Feature selection in single and ensemble learning-based bankruptcy prediction models.* Expert Systems, 2019. **36**(1).https://doi.org/10.1111/exsy.12335

[17] Liang D., Tsai C.-F., Dai A.-J., and Eberle W., *A novel classifier ensemble approach for financial distress prediction.* Knowledge and Information Systems, 2017. **54**(2): p. 437-462.https://doi.org/10.1007/s10115-017-1061-1

[18] Newman A. a. a. D. *Polish companies bankruptcy data Data Set*. 2007 [cited 2021 April/22]; Available from: https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data.

[19] Newman A. a. a. D. *Statlog (Australian Credit Approval) Data Set*. 2007 August/31 [cited 2021; Available from: https://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval).

[20] Newman A. a. a. D. *Statlog (German Credit Data) Data Set*. 2007 September/1 [cited 2021; Available from: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

[21] Smiti S. and Soui M., *Bankruptcy Prediction Using Deep Learning Approach Based on Borderline SMOTE.* Information Systems Frontiers, 2020. **22**(5): p. 1067-1083.https://doi.org/10.1007/s10796-020-10031-6

[22] Beaver W. H., *Financial Ratios as Predictors of Failure.* Journal of Accounting Research, 1966. **4**: p. 71-111.https://doi.org/10.2307/2490171

[23] Altman E. I., *Financial Ratios, Discriminant Analysis and The Prediction of Coroporate Bankruptcy*. The Journal of FINANCE, 1968. 23(4): p 589-609.https://doi.org/10.1111/j.1540-6261.1968.tb00843.x

[24] Ohlson J. A., *Financial Ratios and the Probabilistic Prediction of Bankruptcy.* 1980. **18**(1): p. 109-131.https://doi.org/10.2307/2490395

[25] Barboza F., Kimura H., and Altman E., *Machine learning models and bankruptcy prediction.* Expert Systems with Applications, 2017. **83**: p. 405-417.https://doi.org/10.1016/j.eswa.2017.04.006

[26] Chaabane I., Guermazi R., and Hammami M., *Enhancing techniques for learning decision trees from imbalanced data.* Advances in Data Analysis and Classification, 2020. **14**(3): p. 677-745.https://doi.org/10.1007/s11634-019-00354-x

[27] Xia Y., Zhao J., He L., Li Y., and Niu M., *A novel tree-based dynamic heterogeneous ensemble method for credit scoring.* Expert Systems with Applications, 2020. **159**.https://doi.org/10.1016/j.eswa.2020.113615

[28] Chou C.-H., Hsieh S.-C., and Qiu C.-J., *Hybrid genetic algorithm and fuzzy clustering for bankruptcy prediction.* Applied Soft Computing, 2017. **56**: p. 298-316.https://doi.org/10.1016/j.asoc.2017.03.014

[29] Choi H., Son H., and Kim C., *Predicting financial distress of contractors in the construction industry using ensemble learning.* Expert Systems with Applications, 2018. **110**: p. 1-10.https://doi.org/10.1016/j.eswa.2018.05.026

[30] Hosaka T., *Bankruptcy prediction using imaged financial ratios and convolutional neural networks.* Expert Systems with Applications, 2019. **117**: p. 287-299.https://doi.org/10.1016/j.eswa.2018.09.039

[31] J U., Metawa N., Shankar K., and Lakshmanaprabu S. K., *Financial crisis prediction model using ant colony optimization.* International Journal of Information Management, 2020. **50**: p. 538-556.https://doi.org/10.1016/j.ijinfomgt.2018.12.001

[32] Schapire Y. F. a. R. E., *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.* journal of computer

and system sciences 55, , 1997. **55**(1): p. 119-139 https://doi.org/10.1006/jcss.1997.1504

[33] Mousavi M. M. and Lin J., *The application of PROMETHEE multi-criteria decision aid in financial decision making: Case of distress prediction models evaluation.* Expert Systems with Applications, 2020. **159**.https://doi.org/10.1016/j.eswa.2020.113438

[34] Zelenkov Y., Fedorova E., and Chekrizov D., *Two-step classification method based on genetic algorithm for bankruptcy forecasting.* Expert Systems with Applications, 2017. **88**: p. 393-401.https://doi.org/10.1016/j.eswa.2017.07.025

[35] Chen T., Guestrin C., *XGBoost: A Scalable Tree Boosting System*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016. p. 785–794.

[36] Son H., Hyun C., Phan D., and Hwang H. J., *Data analytic approach for bankruptcy prediction.* Expert Systems with Applications, 2019. **138**.https://doi.org/10.1016/j.eswa.2019.07.033

[37] Jabeur S. B., Gharib C., Mefteh-Wali S., and Arfi W. B., *CatBoost model and artificial intelligence techniques for corporate failure prediction.* Technological Forecasting and Social Change, 2021. **166**.https://doi.org/10.1016/j.techfore.2021.120658

[38] Kou G., Xu Y., Peng Y., Shen F., Chen Y., Chang K., and Kou S., *Bankruptcy prediction for SMEs using transactional data and two-stage multiobjective feature selection.* Decision Support Systems, 2021. **140**.https://doi.org/10.1016/j.dss.2020.113429

[39] Prokhorenkova L., Gusev G., Vorobev A., Dorogush A. V., Gulin A. CatBoost: unbiased boosting with categorical features. in 32nd Conference on Neural Information Processing Systems (NIPS). 2018.https://doi.org/arXiv:1706.09516

[40] Dorogush A. V., Ershov V., Gulin A., *CatBoost: gradient boosting with categorical features support.* Workshop on Machine Learning Systems, 2018.https://doi.org/arXiv:1810.11363

[41] Bentéjac C., Csörgő A., and Martínez-Muñoz G., *A comparative analysis of gradient boosting algorithms.* Artificial Intelligence Review, 2020. **54**(3): p. 1937-1967.https://doi.org/10.1007/s10462-020-09896-5

[42] Zięba M., Tomczak S. K., and Tomczak J. M., *Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction.* Expert Systems with Applications, 2016. **58**: p. 93-101.https://doi.org/10.1016/j.eswa.2016.04.001

[43] Tsai C. F., *Two-stage hybrid learning techniques for bankruptcy prediction*.* Statistical Analysis and Data Mining: The ASA Data Science Journal, 2020. **13**(6): p. 565-572.https://doi.org/10.1002/sam.11482

[44] Sun J., Lang J., Fujita H., and Li H., *Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates.* Information Sciences, 2018. **425**: p. 76-91.https://doi.org/10.1016/j.ins.2017.10.017

[45] Abdouli N. O. S. S. A., *Handling the Class Imbalance Problem in Binary Classification*. 2014, Masdar Institute of Science and Technology: Masdar Institute of Science and Technology.

[46] Le T., Lee M., Park J., and Baik S., *Oversampling Techniques for Bankruptcy Prediction: Novel Features from a Transaction Dataset.* Symmetry, 2018. **10**(4).https://doi.org/10.3390/sym10040079

[47] Kim H., Cho H., and Ryu D., *Corporate Default Predictions Using Machine Learning: Literature Review.* Sustainability, 2020. **12**(16).https://doi.org/10.3390/su12166325

[48] Faris H., Abukhurma R., Almanaseer W., Saadeh M., Mora A. M., Castillo P. A., and Aljarah I., *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, 2020. **9**(1): p. 31-53.https://doi.org/10.1007/s13748-019-00197-9

[49] Al Majzoub H., Elgedawy I., Akaydın Ö., and Köse Ulukök M., *HCAB-SMOTE: A Hybrid Clustered Affinitive Borderline SMOTE Approach for Imbalanced Data Binary Classification.* Arabian Journal for Science and Engineering, 2020. **45**(4): p. 3205-3222.https://doi.org/10.1007/s13369-019-04336-1

[50] Nitesh V. Chawla K. W. B., Lawrence O. Hall, W. Philip Kegelmeyer, *SMOTE: Synthetic Minority Over-sampling Technique.* Journal of Artificial Intelligence Research 2002. **16** p. 321–357.https://doi.org/10.1613/jair.953

[51] Batista G. E. a. P. A., Prati R. C., and Monard M. C., *A study of the behavior of several methods for balancing machine learning training data.* ACM SIGKDD Explorations Newsletter, 2004. **6**(1): p. 20-29.https://doi.org/10.1145/1007730.1007735

[52] Yan Y., Liu R., Ding Z., Du X., Chen J., and Zhang Y., *A Parameter-Free Cleaning Method for SMOTE in Imbalanced Classification.* IEEE Access, 2019. **7**: p. 23537-23548.https://doi.org/10.1109/access.2019.2899467

[53] Nekooeimehr I. and Lai-Yuen S. K., *Cluster-based Weighted Oversampling for Ordinal Regression (CWOS-Ord).* Neurocomputing, 2016. **218**: p. 51-60.https://doi.org/10.1016/j.neucom.2016.08.071

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

[55] Gogas P., Papadimitriou T., and Agrapetidou A., *Forecasting bank failures and stress testing: A machine learning approach.* International Journal of Forecasting, 2018. **34**(3): p. 440-455.https://doi.org/10.1016/j.ijforecast.2018.01.009

[56] Shrivastav S. K. and Ramudu P. J., *Bankruptcy Prediction and Stress Quantification Using Support Vector Machine: Evidence from Indian Banks.* Risks, 2020. **8**(2).https://doi.org/10.3390/risks8020052

[57] Jaiswal S. *Javatpoint*. 2011; Available from: https://www.javatpoint.com/.James G., Witten D., Hastie T., Tibshirani

[58] R., *An Introduction to Statistical Learning*. 1 ed. Springer Texts in Statistics. 2013: Springer, New York. XIV, 426.

[59] Wang G., Ma J., Chen G., and Yang Y., *Financial distress prediction: Regularized sparse-based Random Subspace with ER aggregation rule incorporating textual disclosures.* Applied Soft Computing, 2020. **90**.https://doi.org/10.1016/j.asoc.2020.106152

[60] Le T. and Baik S., *A Robust Framework for Self-Care Problem Identification for Children with Disability.* Symmetry, 2019. **11**(1).https://doi.org/10.3390/sym11010089

[61] Delen D., Kuzey C., and Uyar A., *Measuring firm performance using financial ratios: A decision tree approach.* Expert Systems with Applications, 2013. **40**(10): p. 3970-3983.https://doi.org/10.1016/j.eswa.2013.01.012