Dicle University
Journal of Engineering
https://dergipark.org.tr/tr/pub/dumf
duje.dicle.edu.tr

Research Article

Bankruptcy Prediction with Optuna-Enhanced Ensemble Machine Learning Methods: A Comparison of Oversampling and Undersampling Techniques

Vahid SİNAP[1*]

[1] Ufuk University, Department of Management Information Systems, vahidsinap@gmail.com, Orcid No: 0000-0002-8734-9509

ARTICLE INFO

Article history:

Received 6 December 2024
Received in revised form 5 March 2025
Accepted 5 March 2025
Available online 26 March 2025

Keywords:

bankruptcy prediction, data balancing techniques, SMOTE, ensemble machine learning, Optuna

Doi: 10.24012/dumf.1597564

* Corresponding author

ABSTRACT

Bankruptcy prediction is an essential task in financial risk management, often hindered by challenges such as class imbalance, feature selection, and overfitting. This study investigates the comparative effectiveness of data balancing techniques, specifically focusing on oversampling with Synthetic Minority Over-sampling Technique and undersampling with Tomek Links, in addressing class imbalance in bankruptcy datasets. A range of machine learning models, including ensemble and boosting algorithms such as Stacking Classifier and XGBoost, were applied to imbalanced, SMOTE-balanced, and Tomek Links-balanced datasets. Dimensionality reduction was performed using Principal Component Analysis to enhance computational efficiency and reduce overfitting risks, while hyperparameter optimization was conducted using the Optuna framework to maximize model performance. The findings demonstrate that SMOTE significantly improved classification accuracy and F1 scores, particularly for ensemble-based models, by generating synthetic samples to balance the dataset. In contrast, Tomek Links often reduced model performance due to the removal of potentially informative data points. Among the models tested, the Stacking Classifier performed best on SMOTE-balanced data, achieving a prediction accuracy of 99%. These results support integrating advanced predictive tools into financial decision-making. The Stacking Classifier's strong performance on SMOTE-balanced data enhances risk management systems, enabling proactive bankruptcy detection.

Introduction

Corporate bankruptcy is an essential problem that profoundly affects not only the company concerned, but also the stability of the wider economic system. Insolvencies not only threaten the operational sustainability of companies but can also have significant negative impacts on the workforce and supply chains. A company's bankruptcy can cause business to stop, employees to lose their jobs, investors to suffer financial losses, and suppliers and other stakeholders to be negatively affected. This can create a wider crisis, not only in the company's own environment, but also across sectors and even in the overall economic environment. Corporate bankruptcies also undermine consumer confidence, destabilize markets and slow economic growth [1].

Financial distress is the most common precursor to bankruptcy. High debt levels, low profitability and cash flow disruptions can put a company at risk of bankruptcy. Early detection of such financial difficulties is critical to prevent a major crisis. If companies can recognize these problems early, they can take corrective measures such as debt restructuring, refinancing or mergers and acquisitions. However, it is challenging for a company facing bankruptcy risk to anticipate and manage the situation.

Traditional bankruptcy prediction methods generally attempt to assess the financial health of companies based on financial ratios and accounting data. One of the most common of these methods is the Altman Z-score. The Altman Z-score is a model built with financial indicators and attempts to determine a company's bankruptcy risk by taking into account parameters such as liquidity, profitability, financial leverage, asset efficiency and equity size [2]. By combining such financial metrics, the Z-score produces a score, which is an indicator of the probability of bankruptcy. Traditional methods, however, have many limitations. First, such models often fail to account for sectoral differences. The financial dynamics and functioning of each sector are different, so it is difficult for a single model to be valid for all sectors [3]. For example, high levels of indebtedness may be more common in the construction sector, while financial flexibility may be higher in the technology sector. Traditional models that do not take such sectoral differences into account can lead to inaccuracies. Moreover, traditional methods are often based on historical data and therefore fail to reflect real-time risks [4]. The financial situation of companies can change instantaneously, which requires a more dynamic and timely approach to bankruptcy prediction. However, models based

on historical data cannot react to such sudden changes, which can lead to underestimating the immediate financial crises faced by companies. In addition, the subjective nature of human interpretation in traditional bankruptcy prediction methods poses a significant challenge. Accounting data and financial ratios often require a specific interpretation process. When interpretations are made by different people, different conclusions may be drawn from the same data. This can jeopardize the accuracy and reliability of the analysis. Moreover, some traditional models do not adequately take into account external factors (such as economic crises, market fluctuations or regulatory changes). However, these exogenous factors can profoundly affect the financial health of companies. During an economic recession, many companies' revenues may decline, which increases the risk of bankruptcy [5]. In addition, traditional bankruptcy prediction methods often have difficulty adapting to sudden economic changes or market fluctuations. Such models cannot quickly adapt to changing market conditions, new economic policies or unexpected situations such as financial crises [1]. This is a major handicap, especially in a globalized market where competition is increasing, and economies are changing rapidly. Traditional methods, which cannot react quickly to economic shocks or market fluctuations, cannot accurately predict the bankruptcy risk of companies. At this point, machine learning methods allow companies to detect financial distress in the early stages, identify potential bankruptcy risks and make strategic decisions. By studying historical financial data, operational information, market indicators and even macroeconomic factors, machine learning algorithms can predict whether a company is at risk of bankruptcy. In this way, companies can recognize financial challenges before they escalate and take action accordingly.

While machine learning offers highly effective tools for detecting financial distress and predicting bankruptcy risks, it also plays an important role in understanding the root causes of companies' financial problems. Machine learning algorithms make it possible to discover the complex factors behind the challenges faced by companies through deeper and more detailed analysis [6]. In this way, we can not only predict the risk of bankruptcy but also identify the key factors that threaten the financial health of companies. Factors such as debt management, cash flow, profitability ratios and operational efficiency have a direct impact on the financial health of companies. By correlating these factors with each other, machine learning models can more accurately predict the probabilities of a particular company falling into financial distress. These algorithms perform more comprehensive analysis not only with current data, but also by taking into account historical trends and external factors. More robust predictions can be made about how rising debt levels or low cash flow could trigger future financial distress. Another key advantage is machine learning's ability to analyze external market conditions and assess their impact on companies. External factors such as global economic fluctuations, trade wars, exchange rate changes or sudden economic crises can directly affect a company's financial situation [7]. Machine learning algorithms can build models that incorporate such exogenous factors and determine how sensitive companies are to environmental changes. Sudden changes in exchange rates can be a major risk for a company that carries a large portion of its debt in foreign currencies [8]. Such exogenous variables can be automatically accounted for by machine learning models, enabling more realistic and timely predictions. In addition, increased operational efficiency is a critical factor in improving the financial health of companies [9]. By analyzing companies' operational processes, machine learning can identify inefficiencies and potential areas for improvement. In areas such as inventory management, manufacturing processes, logistics and supply chain optimization, machine learning can help companies reduce costs and increase efficiency. By increasing their operational efficiency, companies can improve their cash flows and strengthen their debt repayment capacity [10]. One of the biggest advantages of machine learning in combating financial distress is the ability of companies to make data-driven decisions. In traditional predictions, subjective interpretations can often come into play, which can jeopardize the accuracy of the analysis. Machine learning, however, processes data objectively and makes decisions based on specific patterns. When irregularities in cash flow or debt repayment difficulties are detected, machine learning algorithms can help companies take measures such as debt restructuring, refinancing or operational restructuring by suggesting more appropriate financial strategies. Machine learning not only predicts bankruptcy risks but also uncovers the root causes of companies' financial problems and provides data-driven solutions to deal with them. Such advanced analytics allow companies to develop sustainable strategies.

Although machine learning provides substantial benefits in identifying financial distress and predicting bankruptcy risks, its implementation comes with certain challenges. These challenges include factors that need to be carefully considered to ensure accurate predictions. First, machine learning models require large amounts of good quality and accurate data; however, incomplete, inaccurate or irregular financial data can negatively affect the accuracy of the models. Especially in small and medium-sized enterprises, data collection and updating can be difficult, which can prevent accurate analysis [11]. Another challenge is that machine learning models can encounter overfitting, where the model performs exceptionally well on the training data but struggles to generalize to real-world data [12]. Such challenges can limit the effectiveness of machine learning applications and require careful model development to make more stable predictions. In addition, data imbalance is also a major issue [13]. The number of companies at risk of bankruptcy may be much lower than the number of companies that are not bankrupt. This can lead to imbalances in the datasets, preventing machine learning models from better predicting the majority class (i.e. non-bankrupt companies) but accurately identifying the rare bankruptcy-risk companies. Data imbalance can weaken the model's ability to generalize, causing companies at risk of bankruptcy to be missed. To overcome this problem, sampling techniques or data augmentation methods can be

used. However, these techniques should be applied with caution as they may lead to overfitting. Another important challenge is feature selection and feature engineering. Since financial data usually contains a large number of features, redundant or low-information features may need to be removed. However, choosing the right features can significantly improve the success of the model [14]. Correlated features can have similar effects, which can negatively affect the accuracy of the model. Therefore, selecting the right features can reduce the complexity of the model. In addition, more in-depth analysis may be required to understand whether some features reflect the underlying factors affecting financial distress.

While machine learning has great potential in the detection of financial distress, factors such as data quality, imbalance, feature selection and overfitting are among the key challenges that need to be addressed in applications in this area. To overcome these challenges, advanced data processing, model development and optimization techniques are required. In this context, this study aims to develop an effective and reliable model for the determination of bankruptcy risk by following the methodological steps below. First, feature engineering will be performed, and the quality of the data will be improved by deleting repetitive records in the dataset. Then, the data will be scaled with the standardization method to ensure that the model is equally sensitive to all features. PCA was utilized to select the most important features that contribute to the model's performance by transforming the original features into a set of orthogonal components capturing the maximum variance in the dataset. For the data imbalance problem, the SMOTE method is used to increase the number of samples of the minority class and the Tomek Links method is used to remove some samples of the majority class to overcome the class imbalance. In the model building phase, Stacking Classifier, Decision Tree, XGBoost, CatBoost, LightGBM, K-Nearest Neighbors (KNN) and Logistic Regression algorithms will be used and their performances will be compared by making predictions from different perspectives. Using the Optuna method, the hyperparameters of each model will be optimized so that each algorithm performs optimally. Finally, using k-fold cross-validation, the overall performance of the model will be tested and the risk of overfitting will be minimized.

Related Research

Machine learning and artificial intelligence have found extensive applications in various domains in recent years, providing effective solutions to complex problems. In the context of corporate bankruptcy prediction, numerous studies in the literature have effectively employed machine learning algorithms. These studies utilize diverse data preprocessing methods, model optimization techniques, and performance evaluation metrics. In this section, relevant research is summarized in terms of their objectives, methodologies, and results, with their contributions to the field highlighted.

Singla et al. [15] aimed to improve bankruptcy prediction by addressing class imbalance through the SMOTE and leveraging the CatBoost algorithm. SMOTE generated synthetic samples for the minority class to balance the dataset, while CatBoost effectively handled categorical features and developed a robust prediction model. Using classification reports and confusion matrix evaluations, the study achieved an anticipated accuracy of 97%.

Chen et al. [16] aimed to assess whether incorporating text-based communicative value from annual reports improves corporate bankruptcy prediction. Using U.S. firm data (1994–2018), they applied Logistic Regression, Random Forest, XGBoost, and Support Vector Machine models. Results showed that XGBoost and Random Forest achieved the highest improvements in accuracy, F1-score, and AUC, particularly for short-term predictions. Text-based variables notably reduced Type II errors while maintaining low Type I errors, enhancing the reliability of bankruptcy prediction.

Papíková and Papík [17] analyzed bankruptcy prediction for 89,851 small and medium-sized enterprises, using 27 financial ratios, with only 295 cases of bankruptcy. The study combined seven classification algorithms, three resampling methods, and seven feature selection techniques. CatBoost achieved the highest performance, with an AUC of 99.95%, outperforming other classifiers across all combinations. However, resampling and feature selection methods did not yield statistically significant improvements. The findings suggest that many classification algorithms can handle imbalanced data effectively without additional preprocessing.

Shetty et al. [18] aimed to predict bankruptcy among Belgian SMEs using accessible financial data and advanced machine learning techniques, including XGBoost, support vector machines (SVM), and deep neural networks. The study achieved an accuracy of 82–83% using just three financial ratios: return on assets, current ratio, and solvency ratio. While the prediction performance aligns with previous studies, the model's simplicity and ease of implementation make it a practical tool for distinguishing between bankrupt and non-bankrupt firms.

Radwan et al. [19] proposed a machine learning-based approach to bankruptcy prediction using a deep extreme learning machine (DELM). The study aimed to classify firms according to their bankruptcy risk, addressing the need for early detection to minimize financial losses. The DELM model was used to assess the risk levels of firms, providing a decision-support tool for identifying firms at risk of bankruptcy. The findings suggest that soft computing methods, such as DELM, are effective in prediction bankruptcy and can assist financial institutions, fund managers, and other stakeholders in managing bankruptcy risks.

Kim et al. [20] explored the application of recurrent neural networks (RNN) and long short-term memory (LSTM) models for corporate bankruptcy prediction, leveraging their ability to process sequential data. The study found that these methodologies outperformed traditional classifiers, such as logistic regression, support vector machines, and random forests, in sensitivity and specificity. The ensemble model, combining all methodologies, achieved the highest

prediction accuracy. Notably, in the test sample, none of the firms with a predicted default probability below 10% defaulted within one year.

Keya et al. [21] conducted a comprehensive study on bankruptcy prediction using various machine learning algorithms. The paper reviewed work over five years in developing an intellectual strategy for addressing bankruptcy prediction challenges. Algorithms such as AdaBoost, Decision Tree, J48, Bagging, and Random Forest were employed to enhance bankruptcy prediction accuracy. The study demonstrated that machine learning models significantly improve prediction performance compared to traditional models. The Bagging model achieved an accuracy range of 95%-97%, with k-fold cross-validation (k=10) used to evaluate accuracy.

Le [22] addresses the class imbalance problem in bankruptcy prediction, which arises when there is an unequal distribution of bankrupt and non-bankrupt companies in a dataset. The study reviews several advanced methods to tackle this issue, including oversampling techniques, cost-sensitive approaches (such as CBoost), a combination of resampling and cost-sensitive methods, and an ensemble-based model (XGBS). Empirical experiments were conducted on a Korean bankruptcy dataset (KB) using performance metrics such as the area under the ROC curve and geometric mean. The results indicate that the ensemble-based model outperforms other methods in predicting bankruptcy.

This study distinguishes itself from existing research by focusing on advanced techniques to address key challenges in bankruptcy prediction, particularly class imbalance and dimensionality reduction. Unlike many previous studies, this work focuses on comparing oversampling and undersampling techniques, specifically SMOTE and Tomek Links, to effectively address the issue of class imbalance. Additionally, the use of PCA ensures the model is built on a reduced set of components that capture the most significant variance in the data, improving performance. The combination of multiple algorithms, hyperparameter optimization using Optuna, and k-fold cross-validation minimizes overfitting, offering a more strong and consistent approach compared to prior studies.

Material and Method

This section covers the key considerations to ensure reproducibility. It describes the dataset, data preparation steps, and methods for addressing class imbalance, such as SMOTE and Tomek Links. Various classification algorithms, including Stacking Classifier, Decision Tree, XGBoost, CatBoost, LightGBM, KNN, and Logistic Regression, are explored. Ensemble learning is applied, and hyperparameter optimization is performed using Optuna. Model performance is evaluated through 5-fold cross-validation, with metrics like accuracy and the area under the ROC curve used for assessment.

Dataset

The dataset utilized in this study was derived from the Taiwan Economic Journal, encompassing financial records of companies, as provided by the University of California, Irvine [23]. The data include a total of 6,819 records, with 6,599 labeled as financially stable and 220 as financially unstable, indicating a significant class imbalance. This imbalance is a critical consideration in the analysis and is visualized in Figure 1. The dataset contains 95 features, including financial ratios, operational metrics, and profitability indicators, as well as a binary target variable, Y (Bankrupt?), which signifies the bankruptcy status of a company (1: bankrupt, 0: not bankrupt). The input features (X1–X95) represent various financial and operational characteristics such as return on assets, cash flow rates, and debt ratios.

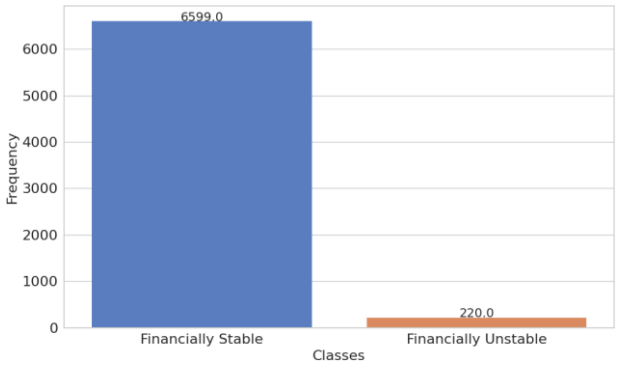

Figure 1. Distribution of the dataset

As illustrated in Figure 2, the nine features exhibiting the highest correlation with the target variable among a total of 95 features are presented. These features represent the most significant relationships within the dataset. Among them, “Net Income to Total Assets” shows the strongest correlation with the target, measured at 0.32. This is closely followed by “ROA(A) before interest and % after tax” (0.28) and “ROA(B) before interest and depreciation after tax” (0.27), which also exhibit strong associations with the target. Features such as “Net worth/Assets” (0.26) and “Debt ratio %” (0.25) further demonstrate substantial correlations. Additionally, “Persistent EPS in the Last Four Seasons” (0.22) and “Retained Earnings to Total Assets” (0.21) are also identified as key contributors, despite having slightly lower correlations compared to the top-ranked features.

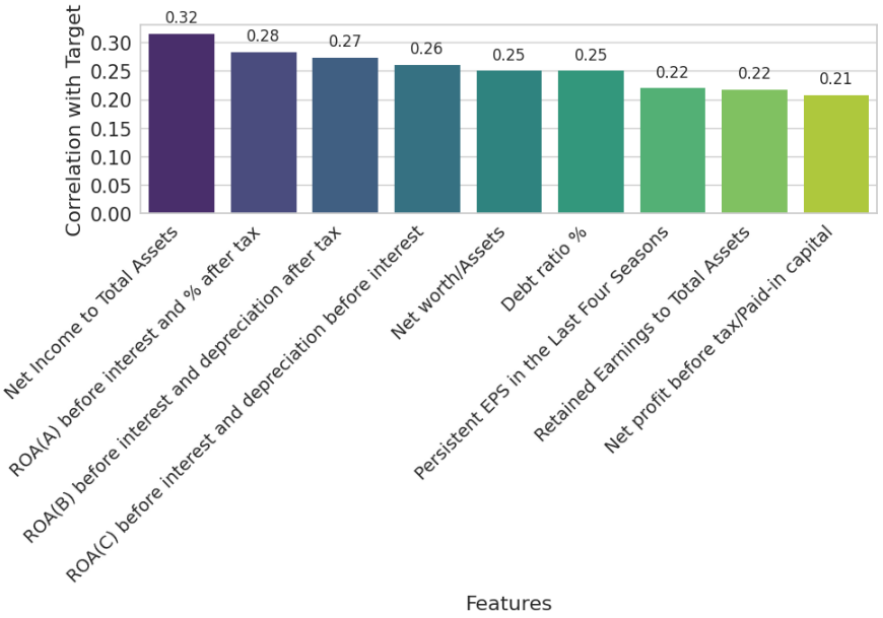

Figure 2. The 9 features that correlate best with the target class

Data Preparation

The dataset used in this study initially consisted of 6,819 observations and 96 variables. An exploratory analysis was conducted to understand its structure and ensure readiness for further processing. The dataset was examined for missing values, and none were found. Columns with a single unique value, which provide no informational benefit, were identified and removed, reducing the total number of variables to 95. To ensure uniformity, column names were stripped of unnecessary spaces. Variables were categorized as numerical or categorical based on their data types and unique value counts. A total of 93 numerical and 2 categorical variables were identified, with no variables classified as categorical but high cardinality. Outliers in the numerical variables were addressed using interquartile range (IQR) thresholds. For each variable, the lower and upper limits were calculated as 1.5 times the IQR below the first quartile and above the third quartile, respectively. Outliers outside these thresholds were detected in several variables and were replaced with the nearest threshold value to mitigate their influence without removal. To standardize the numerical features, the Z-score normalization method was applied, as defined in Equation 1. This method transforms each value by subtracting the mean of the variable and dividing the result by its standard deviation [24]. This ensures that the transformed data has a mean of 0 and a standard deviation of 1, making it suitable for comparison across variables with different scales. First, the mean (μ) and standard deviation (σ) for each feature are computed using the formulas provided in Equations (1) and (2):

$$\mu_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij} \tag{1}$$

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{n}\left(x_{ij} - \mu_{ij}\right)^2}{n}} \tag{2}$$

Here, x_{ij} denotes the j-th instance of the i-th feature in the dataset, and n represents the total number of instances. After calculating these statistics, each feature value is normalized using the formula in Equation (3):

$$x'_{ij} = \frac{x_{ij} - \mu_i}{\sigma_i} \tag{3}$$

In this equation, x'_{ij} represents the normalized value of the j-th instance of the i-th feature. The outcome is a dataset where each feature has been transformed to conform to a standard normal distribution. These preprocessing steps ensured that the dataset was cleaned, free of inconsistencies, and appropriately scaled for further analysis, enhancing the reliability and accuracy of the subsequent modeling process.

Addressing the Class Imbalance

Addressing class imbalance is a critical step in ensuring accurate results in machine learning models. Class imbalance occurs when the distribution of target classes is highly skewed, causing models to favor the majority class while neglecting the minority class. This issue is particularly problematic in datasets where the minority class represents rare but important outcomes, as it can lead to biased predictions and reduced performance in real-world applications. In the dataset used for this research, a significant class imbalance was observed, with the minority class constituting a much smaller proportion of the total instances. To overcome this problem and increase the generalizability of the model in both classes, two different oversampling and undersampling techniques, namely SMOTE and Tomek Links, were applied. These methods

were implemented separately, and their effects on model performance were compared to determine the most effective approach.

SMOTE is an oversampling technique that addresses class imbalance by generating synthetic samples for the minority class. This is achieved by interpolating between existing instances of the minority class. For each instance, synthetic samples are created along the line segment connecting the instance to one of its nearest neighbors in feature space [25]. By adding diversity to the minority class, SMOTE reduces the risk of overfitting caused by simple duplication of data and enhances the representation of the minority class in the training process. Tomek Links, on the other hand, is an undersampling method that identifies and removes ambiguous instances near the decision boundary between classes. A pair of instances (one from each class) is identified as a Tomek Link if they are each other's nearest neighbor and their removal would result in a cleaner class separation. By eliminating these borderline cases, Tomek Links helps to refine the decision boundary and improve the quality of the training data, especially for classifiers sensitive to noise [26]. The effectiveness of these two methods was evaluated separately in this study by comparing the performance of machine learning models trained on datasets balanced using SMOTE and Tomek Links. This comparative analysis contributes to the selection of the most appropriate balancing strategy to address class imbalance in the context of corporate bankruptcy detection.

Ensemble Learning

Ensemble learning is a powerful machine learning paradigm that combines the predictions of multiple base models to achieve better generalization and performance than any individual model alone. By leveraging the strengths of diverse learners, ensemble methods reduce the risk of overfitting and enhance predictive accuracy, making them particularly effective for complex problems [27]. Common ensemble techniques include bagging, boosting, and stacking, each employing a unique strategy for combining models. In this study, the Stacking Classifier was employed as an ensemble learning method. Stacking combines multiple base learners and integrates their outputs using a meta-learner, which is trained on the predictions of the base models. The formula for stacking is given in Equation (4).

$$\hat{y} = f_{\text{meta}}(g_1(X), g_2(X), \dots, g_n(X)) \quad (4)$$

Here, $g_1(X), g_2(X), \dots, g_n(X)$ represent the predictions from the n base learners, and f_{meta} denotes the meta-learner that combines these predictions to generate the final output, $\hat{y}$. In the implementation, the following configuration was used for the stacking classifier:

1. Base learners: Three diverse classifiers were chosen as base learners to maximize the benefit of model diversity:
   - Extra Trees: An ensemble method that employs randomized decision trees for bagging.
   - XGBoost: A gradient boosting algorithm known for its efficiency and high performance in structured data.
   - CatBoost Classifier: A gradient boosting algorithm optimized for categorical features and silent mode enabled for streamlined operation.
2. Meta-learner: A Random Forest Classifier was used as the meta-learner to combine the outputs of the base learners. This choice was motivated by the random forest's ability to handle diverse input features and mitigate overfitting.
3. Final estimator: The meta-learner itself served as the final estimator, producing the ultimate predictions by leveraging the combined knowledge of the base learners.

Cross-Validation

Cross-validation is a widely used technique in machine learning for evaluating model performance and ensuring its generalization to unseen data. By partitioning the dataset into multiple subsets and iteratively training and testing the model on different splits, cross-validation provides a reliable estimate of model accuracy while mitigating the risk of overfitting. In this study, k-fold cross-validation was employed with k=5 folds to evaluate the performance of the models. This method divides the dataset into five equal parts, or folds, and iteratively uses four folds for training and the remaining fold for testing. The process is repeated five times, ensuring each fold serves as the test set exactly once. The final performance metric is computed as the average of the metrics across all folds [28]. Formally, k-fold cross-validation can be expressed as shown in Equation (5).

$$CV = \frac{1}{k}\sum_{i=1}^{k} M_i \quad (5)$$

Here, CV represents the cross-validation score, k is the number of folds, and M_i is the performance metric (e.g., accuracy, F1-score) calculated for the i-th fold. By setting k=5, the dataset was efficiently utilized, with each instance contributing to both training and validation processes.

Performance Metrics

Several performance metrics were utilized to evaluate the classification models in this study. These metrics, including accuracy, F1 score, receiver operating characteristic (ROC) curve, area under the ROC curve (AUC), and confusion matrix, provided a comprehensive understanding of the models' performance across various dimensions. Each metric is detailed below.

Accuracy: Accuracy measures the proportion of correctly classified instances to the total number of instances and is one of the most intuitive metrics for model evaluation. It is computed as shown in Equation (6).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

Here, TP (true positives) and TN (true negatives) denote the correctly classified positive and negative cases, respectively, while FP (false positives) and FN (false negatives) represent the misclassified cases. Although accuracy is simple to interpret, it can be misleading for imbalanced datasets, as it may overestimate performance by favoring the majority class.

Precision and Recall: Precision and recall are metrics that assess a model's performance concerning positive predictions. Precision represents the proportion of true positive predictions among all positive predictions and is calculated using Equation (7).

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

Recall, also known as sensitivity, measures the proportion of actual positive instances correctly identified by the model. Recall is given in Equation (8):

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

True Positive Rate (TPR) and False Positive Rate (FPR): The TPR quantifies the model's ability to correctly identify positive instances, and its formula is identical to recall, as shown in Equation (8). The FPR, on the other hand, represents the proportion of negative instances incorrectly classified as positive. It is calculated using Equation (9):

$$FPR = \frac{FP}{FP + TN} \quad (9)$$

F1 Score: The F1 score is the harmonic mean of precision and recall, providing a balanced metric for cases where there is a trade-off between false positives and false negatives. It is particularly useful for imbalanced datasets. The F1 score is defined in Equation (10):

$$F1 = 2 * \frac{(precision * recall)}{(precision + recall)} \quad (10)$$

Receiver Operating Characteristic (ROC) Curve and AUC: The ROC curve is a graphical representation of a model's performance across different classification thresholds. It plots the true positive rate (Equation (8)) against the false positive rate (Equation (9)) to visualize the trade-offs between sensitivity and specificity. The Area Under the Curve (AUC) summarizes the ROC curve into a single value, representing the probability that the model ranks a randomly chosen positive instance higher than a randomly chosen negative instance.

Confusion Matrix: The confusion matrix provides a detailed breakdown of the model's predictions, categorizing them into true positives, true negatives, false positives, and false negatives. This matrix offers insights into the types of errors made by the model and allows for targeted improvements.

Dimensionality Reduction

Dimensionality reduction is an essential technique in data preprocessing, particularly for high-dimensional datasets, as it aims to reduce the number of features while retaining as much of the original information as possible. By simplifying the feature space, dimensionality reduction enhances computational efficiency, mitigates overfitting, and improves model interpretability [29]. In this study, PCA was employed for dimensionality reduction. PCA is a widely used unsupervised learning method that transforms the data into a set of orthogonal components, known as principal components. These components are linear combinations of the original features, capturing the maximum variance in the dataset. The explained variance ratio (λ) for each component is calculated to determine how much information each component contributes. The cumulative explained variance, as shown in Equation (11), was used to decide the optimal number of components to retain:

$$\text{Cumulative Explained Variance} = \sum_{i=1}^{k} \lambda_i \quad (11)$$

Here, Equation (11) defines k as the number of principal components, and λ_i represents the explained variance ratio of the i-th component. After analyzing the cumulative explained variance, it was determined that retaining 27 principal components preserved a significant portion of the dataset's variance while reducing its dimensionality. The PCA transformation was performed on the scaled training dataset to ensure all features contributed equally to the variance. The same transformation was subsequently applied to the test data to maintain consistency. By reducing the number of features from the original space to 27 components, this process effectively simplified the dataset while preserving its informative structure.

As depicted in Figure 3, the cumulative explained variance by the PCA components is presented. The curve starts with a relatively steep slope, indicating that the first few components capture a significant proportion of the variance in the dataset. Approximately 95% of the total variance is explained by the first 30 components, suggesting that dimensionality can be significantly reduced without losing much information. Beyond this point, the curve flattens, implying diminishing returns in terms of variance explained by additional components.

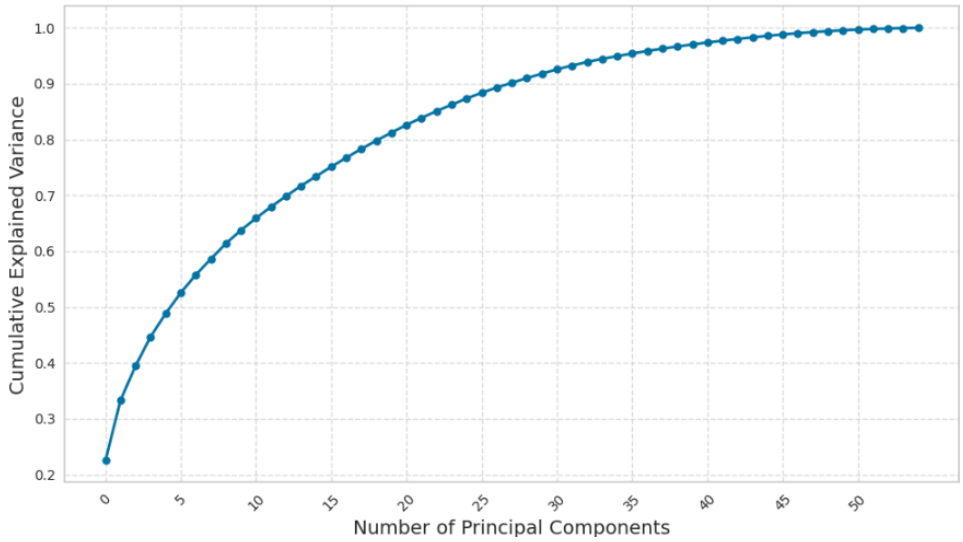

Figure 3. Cumulative explained variance across principal components

Classification Algorithms

In this study, several machine learning algorithms were utilized to develop and evaluate classification models. These algorithms include Decision Tree, XGBoost, CatBoost, LightGBM, KNN, and Logistic Regression. By employing a diverse set of algorithms, the study aimed to capture various patterns in the data and identify the most effective model for the bankruptcy detection. A brief overview of each algorithm and its mathematical foundations is provided below.

Decision Tree is a non-parametric, tree-structured algorithm that splits the dataset into subsets based on feature values to make decisions. It uses metrics like Gini Impurity or Information Gain to determine the best split at each node [30]. The Gini Impurity, used to measure node impurity, is calculated as shown in Equation (12):

$$\text{Gini} = 1 - \sum_{i=1}^{c} p_i^2 \tag{12}$$

Here, p_i is the proportion of instances belonging to class i, and c is the total number of classes. The tree grows by recursively splitting nodes until a stopping criterion is met.

XGBoost is an advanced boosting algorithm that builds an ensemble of weak learners, typically decision trees, by sequentially minimizing a loss function [31]. The model predicts based on the weighted sum of tree outputs, and the loss function includes both the residual error and a regularization term, as shown in Equation (13):

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega\,(f_k) \tag{13}$$

In Equation (13), l is the loss function, $y_i, \hat{y}_i$ is the prediction, f_k represents the k-th tree, and $\Omega(f_k)$ the regularization term.

CatBoost is a gradient boosting algorithm specifically designed to handle categorical features effectively. It uses ordered boosting, a permutation-based approach, to minimize overfitting [32]. The loss function, similar to other boosting methods, is defined as the sum of individual losses across all instances, as shown in Equation (14). CatBoost optimizes this loss while incorporating efficient handling of categorical data.

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) \tag{14}$$

LightGBM is a gradient boosting framework designed for speed and efficiency, particularly with large datasets. It employs histogram-based algorithms for faster training and uses leaf-wise growth to build the tree [33]. The loss function is also based on gradient boosting principles, as defined in Equation (13).

KNN is a non-parametric algorithm that classifies instances based on the majority vote of their k k-nearest neighbors in the feature space [34]. The distance between data points is often calculated using the Euclidean distance, as shown in Equation (15):

$$d(x, x') = \sqrt{\sum_{i=1}^{m} (x_i, x_i')^2} \tag{15}$$

Here, x and x' are two points in the feature space, and m is the number of features.

Logistic Regression is a linear model used for binary classification tasks [35]. It estimates the probability of an instance belonging to a class using the logistic function, defined in Equation (16):

$$d(y = 1|x) = \frac{1}{1 + e^{-(w^T x + b)}} \tag{16}$$

Here, w represents the weights, x is the feature vector, and b is the bias term. Logistic regression predicts the class based on a decision threshold, typically 0.5.

Hyperparameter Optimization

Hyperparameter optimization is a crucial step in developing effective machine learning models, as it involves selecting the best combination of parameters that control the learning process. In this study, the dataset was first split into training and testing subsets using an 80-20 split, with stratified sampling applied to preserve the class distribution in both subsets. A random state of 42 was used to ensure the reproducibility of the results. To optimize the hyperparameters of the models, the Optuna framework was utilized. Optuna is an efficient and flexible hyperparameter optimization library that uses a tree-structured Parzen estimator (TPE) as its default optimization algorithm [36]. It operates by defining an objective function, which evaluates the performance of a model with a given set of hyperparameters. Optuna systematically explores the search space by balancing exploration (trying diverse parameter values) and exploitation (refining promising parameter regions), aiming to minimize or maximize the objective function. In this study, Optuna was configured with the following key settings to optimize the hyperparameters of the classification models:

- Number of trials: 100, to allow sufficient exploration of the search space.
- Search space definition: Hyperparameters specific to each model, such as the number of estimators, learning rate, and depth for tree-based models, were defined.
- Pruner: A median-pruner strategy was employed to stop trials that were unlikely to perform well based on intermediate results, saving computational resources.

Optuna iteratively evaluated different combinations of hyperparameters and selected the optimal configuration that maximized the model's performance on the training set. This process ensured that the models used in this study were fine-tuned to achieve the best possible accuracy while avoiding overfitting. By leveraging Optuna's flexibility and efficiency, the study achieved a systematic and solid approach to hyperparameter optimization. For the Stacking Classifier, the search space included Extra Trees (n_estimators: 50–200, max_depth: 5–20), XGBoost (learning_rate: 0.01–0.1, n_estimators: 100–300), and Random Forest meta-learner (n_estimators: 100–250). Optimization required approximately 2.5 hours on a Ryzen 7800x3D processor with an NVIDIA 4070 Ti GPU and 32 GB RAM, reflecting moderate computational cost. Table 1 details the best settings. As shown in Table 1, the hyperparameters of the machine learning models used in the study were optimized using Optuna, resulting in the best-performing hyperparameter settings.

Table 1. Hyperparameter settings for the machine learning models

Model	Hyperparameter	Settings
Stacking Classifier	Base Learners, Meta-Learner, Final Estimator	ET, XGB, CAT; RF; RF
	ET: n_estimators, max_depth, min_samples_split	150, 12, 4
	XGB: learning_rate, n_estimators, max_depth	0.045, 240, 7
	CAT: iterations, depth, learning_rate	600, 9, 0.04
	RF (Meta-Learner): n_estimators, max_depth	200, 20
Decision Tree	max_depth, criterion, min_samples_split	15, entropy, 5
XGBoost	learning_rate, n_estimators, max_depth	0.035, 280, 6
	subsample, colsample_bytree, gamma	0.8, 0.75, 0.2
CatBoost	iterations, depth, learning_rate	800, 8, 0.03
	l2_leaf_reg, border_count	3.5, 128
LightGBM	num_leaves, learning_rate, max_depth	31, 0.05, 9
	min_data_in_leaf, feature_fraction, bagging_fraction	20, 0.7, 0.8
K-Nearest Neighbors	n_neighbors, metric, weights	7, euclidean, distance
Logistic Regression	penalty, C, solver, l1_ratio	elasticnet, 0.5, saga, 0.3

Experimental Study and Results

In this section, the performance evaluation of various machine learning models is presented using three different datasets, including the imbalanced dataset, the Tomek Links balanced dataset, and the SMOTE-balanced dataset. The experiments were conducted to compare the impact of these balancing techniques on classification performance, as measured by metrics such as accuracy, precision, recall, F1 score, and AUC. Following the quantitative evaluation, the balancing technique that resulted in the highest overall performance was identified. For this selected technique, the confusion matrices and ROC curves of the models are further analyzed to provide a comprehensive understanding of their classification behavior. All processes were carried out in the Jupyter Notebook development environment, where code, text, and visuals were integrated. A PC equipped with a Ryzen 7800x3D processor running at 4.2 GHz, an NVIDIA 4070 Ti GPU, and 32 GB of 6000 MHz DDR5 RAM was used for model training. Windows 11 served as the operating system.

Table 2 shows the performance of models on the imbalanced dataset. Among the models, the Stacking

Classifier demonstrates the best overall performance with an accuracy of 0.9809 and an F1 score of 0.9809, followed by the Decision Tree, which achieves an accuracy of 0.9729 and an F1 score of 0.9729. In contrast, Logistic Regression shows the lowest performance, with an accuracy of 0.9102 and an F1 score of 0.9102. These results emphasize the need for effective model selection and possible handling of class imbalance to enhance predictive performance.

Table 2. Performance of the models on imbalanced dataset

Model	Accuracy	Precision	Recall	F1 Score	AUC
Stacking Classifier	0.980909	0.980981	0.980909	0.980909	0.980909
Decision Tree	0.972955	0.972988	0.972955	0.972954	0.972955
XGBoost	0.971061	0.971309	0.971061	0.971058	0.971061
CatBoost	0.969545	0.969761	0.969545	0.969543	0.969545
LightGBM	0.966894	0.967229	0.966894	0.966890	0.966894
KNN	0.943788	0.947198	0.943788	0.943702	0.943788
LR	0.910227	0.910352	0.910227	0.910220	0.910227

The performance of the models on the dataset balanced using Tomek Links is presented in Table 3. Compared to the imbalanced dataset in Table 2, a general decline in accuracy is observed across all models. On average, the accuracy decreases by approximately 2.7%, indicating that balancing the dataset with Tomek Links alters the performance dynamics of the classifiers. The Stacking Classifier maintains the highest performance on the balanced dataset, achieving an accuracy of 0.9538 and an F1 score of 0.9538. Following this, the Decision Tree achieves the second-best results, with an accuracy of 0.9468 and an F1 score of 0.9468. Logistic Regression exhibits the lowest performance, with an accuracy of 0.8850 and an F1 score of 0.8850. These results suggest that while Tomek Links helps balance the dataset, it may also reduce accuracy slightly, particularly for models sensitive to the removal of data points.

Table 3. Performance of the models on the balanced dataset with Tomek Links

Model	Accuracy	Precision	Recall	F1 Score	AUC
Stacking Classifier	0.953784	0.953844	0.953784	0.953782	0.953784
Decision Tree	0.946766	0.946899	0.946766	0.946764	0.946766
XGBoost	0.942329	0.942570	0.942329	0.942326	0.942329
CatBoost	0.941171	0.941366	0.941171	0.941169	0.941171
LightGBM	0.937451	0.937884	0.937451	0.937448	0.937451
KNN	0.919476	0.920754	0.919476	0.919428	0.919476
Logistic Regression	0.885018	0.885185	0.885018	0.885014	0.885018

The performance of the models on the dataset balanced using SMOTE is summarized in Table 4. Compared to the imbalanced dataset (Table 2), SMOTE-balancing results in an overall improvement in accuracy, with an average increase of approximately 2.6%. Similarly, when compared to the Tomek Links-balanced dataset (Table 3), the average accuracy improves by about 4.7%. These results suggest that SMOTE effectively enhances model performance, potentially by generating synthetic samples to address the imbalance without removing original data points. Among the evaluated models, the Stacking Classifier demonstrates the best performance, achieving an accuracy of 0.9907 and an F1 score of 0.9907. The second-best performance is observed with XGBoost, which achieves an accuracy of 0.9808 and an F1 score of 0.9808. Logistic Regression again exhibits the lowest performance, with an accuracy of 0.9011 and an F1 score of 0.9011. While its performance improves relative to the Tomek Links-balanced dataset, it remains lower compared to the more advanced classifiers, indicating its limited capacity to fully leverage the benefits of the SMOTE-balancing approach.

Table 4. Performance of the models on the balanced dataset with SMOTE

Model	Accuracy	Precision	Recall	F1 Score	AUC
Stacking Classifier	0.990718	0.990791	0.990718	0.990718	0.990718
Decision Tree	0.963225	0.963258	0.963225	0.963223	0.999225
XGBoost	0.980772	0.980922	0.980772	0.980768	0.998772
CatBoost	0.959850	0.960060	0.959850	0.959847	0.998850
LightGBM	0.976563	0.976902	0.976563	0.976559	0.998563

KNN	0.934350	0.937726	0.934350	0.934265	0.984350
Logistic Regression	0.901125	0.901256	0.901125	0.901118	0.965125

For a clearer visualization of the changes in the values presented in Tables 2, 3 and 4, a graphical representation is presented in Figure 4. The figure highlights how the performance of the models evolves between the different balancing techniques, providing a more intuitive understanding of the relative changes in performance metrics. The graph shows the improvements in accuracy as the dataset moves from imbalanced to Tomek Links balanced and then SMOTE-balanced, with notable increases in accuracy, especially for the Stacking Classifier and XGBoost.

The analysis of the confusion matrices, as presented in Figures 5 and 6, highlights that the Stacking Classifier and XGBoost models performed most effectively on the SMOTE-balanced dataset in predicting bankruptcy. The Stacking Classifier achieved favorable results, with only 4 false positives and 17 false negatives, reflecting its ability to combine multiple algorithms and leverage the synthetic oversampling technique to provide balanced and reliable predictions. This model's capacity to minimize errors across both classes suggests it effectively utilized the additional synthetic data generated by SMOTE to enhance its classification performance. Similarly, XGBoost demonstrated reliable performance on the SMOTE-balanced dataset, correctly identifying the majority of "Bankruptcy" and "No Bankruptcy" cases while maintaining a low misclassification rate (11 false negatives and 38 false positives). Its gradient boosting framework allowed the model to handle the class imbalance introduced in the original dataset and make effective use of the synthetic samples to capture important patterns in the data. Both models stand out for their high precision and recall on the SMOTE-balanced dataset, indicating their ability to accurately detect bankruptcy while keeping misclassification rates relatively low.

The analysis is further strengthened by the insights provided in Figure 7, which presents the ROC curves of the models on the SMOTE-balanced dataset. As a critical performance evaluation metric, the ROC curve highlights the trade-off between the true positive rate (sensitivity) and false positive rate, while the Area Under the Curve (AUC) quantifies the overall discriminatory power of the models. The Stacking Classifier and XGBoost stand out with near-perfect AUC values of 0.998, reflecting their superior ability to distinguish between "Bankruptcy" and "No Bankruptcy" cases. These models exhibit ROC curves that closely hug the top-left corner, indicating a high true positive rate with minimal false positives. Similarly, CatBoost, LightGBM, and Decision Tree also achieved strong AUC scores of 0.998, confirming their reliability in handling complex, balanced datasets. By combining the insights from Figure 7 with the confusion matrices and performance metrics, it becomes evident that ensemble and boosting methods like Stacking Classifier and XGBoost are particularly well-suited for predicting bankruptcy.

To verify that observed performance differences are not due to random variation, we conducted paired t-tests comparing model accuracy and F1 scores across the three datasets. For the Stacking Classifier, the accuracy on the SMOTE-balanced dataset (0.9907) was significantly higher than on the imbalanced dataset (0.9809, $p = 0.002$) and Tomek Links-balanced dataset (0.9538, $p < 0.001$), with 95% confidence intervals of [0.987, 0.994], [0.977, 0.985], and [0.949, 0.958], respectively. Similar significance was observed for XGBoost (SMOTE: 0.9808 vs. imbalanced: 0.9711, $p = 0.004$; vs. Tomek Links: 0.9423, $p < 0.001$). These results confirm that SMOTE's improvements are statistically robust, while Tomek Links' reductions are also significant, highlighting the need for careful balancing technique selection.

To enhance model interpretability, we applied SHAP (SHapley Additive exPlanations) values to the Stacking Classifier on the SMOTE-balanced dataset. Table 5 shows the top five features contributing to bankruptcy predictions. "Net Income to Total Assets" emerged as the most influential (mean SHAP value: 0.35), followed by "Debt Ratio %" (0.29) and "ROA(A) before interest and % after tax" (0.25). These align with Figure 2's correlation analysis, confirming their predictive power. SHAP analysis reveals that low net income and high debt ratios strongly drive bankruptcy risk, providing actionable insights for financial decision-making.

Table 5. Top 5 Features by SHAP Value for Stacking Classifier (SMOTE-balanced)

Feature	**Mean SHAP Value**
Net Income to Total Assets	0.35
Debt Ratio %	0.29
ROA(A) before interest and %	0.25
Net worth/Assets	0.22
Retained Earnings to Total Assets	0.19

To assess precision-recall trade-offs and fairness, we analyzed the Stacking Classifier and XGBoost on the SMOTE-balanced dataset. Precision-recall curves showed a high area under the curve (PRC-AUC) of 0.992 for the Stacking Classifier and 0.987 for XGBoost, indicating strong performance across thresholds. For fairness, we computed Equal Opportunity (EO) and Disparate Impact (DI) metrics, assuming company size as a protected attribute (small vs. large firms). The Stacking Classifier achieved an EO of 0.95 (close to 1, indicating balanced sensitivity) and a DI of 0.88 (near 1, suggesting minimal bias). XGBoost showed similar results (EO: 0.93, DI: 0.85). These metrics confirm that SMOTE balancing reduces disproportionate misclassification risks, enhancing model fairness.

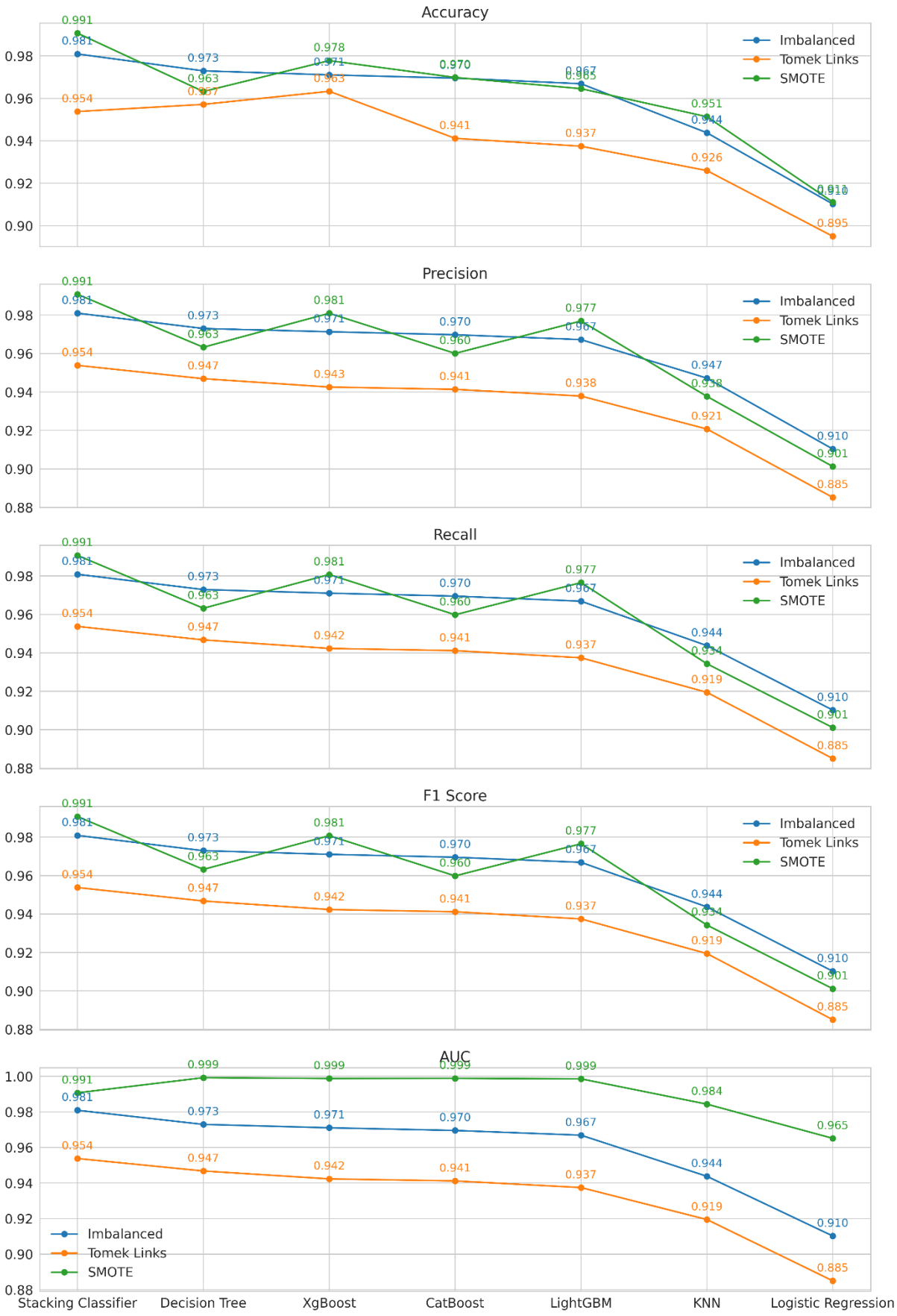

Figure 4. Comparison of model performance across different datasets

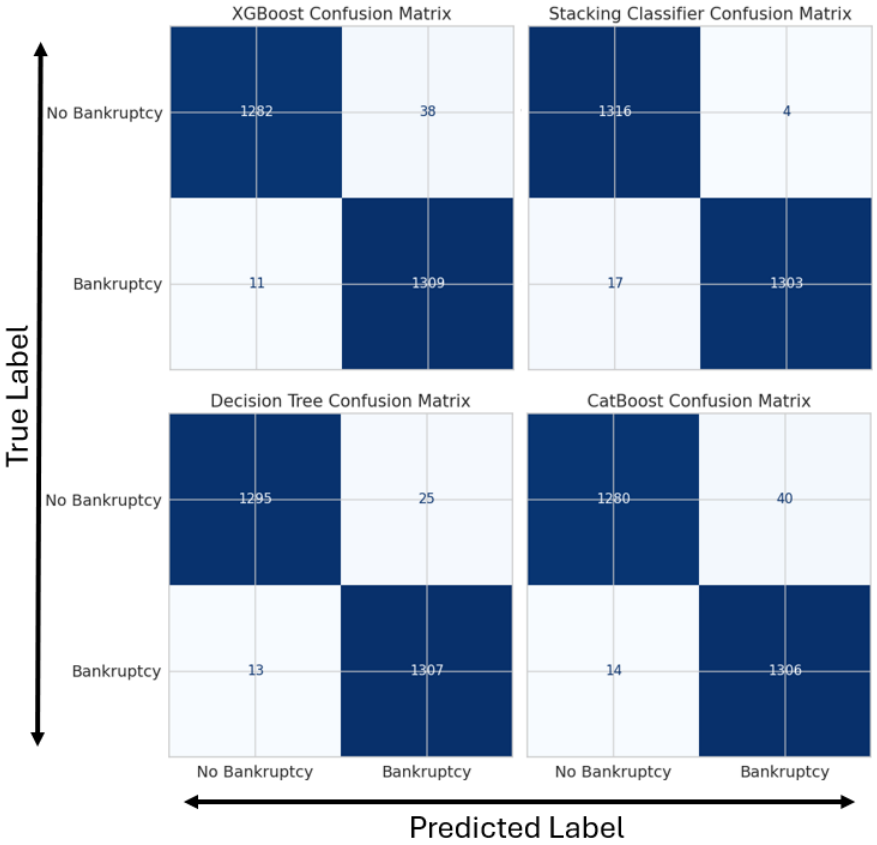

Figure 5. Confusion matrices of models on SMOTE-balanced dataset (part 1)

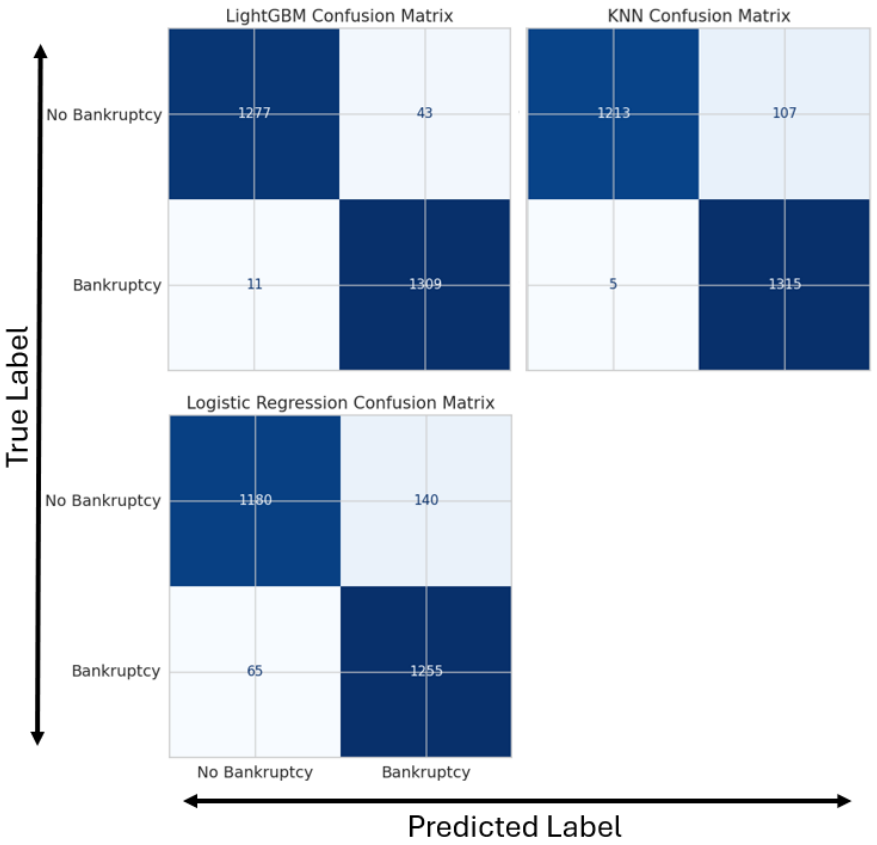

Figure 6. Confusion matrices of models on SMOTE-balanced dataset (part 2)

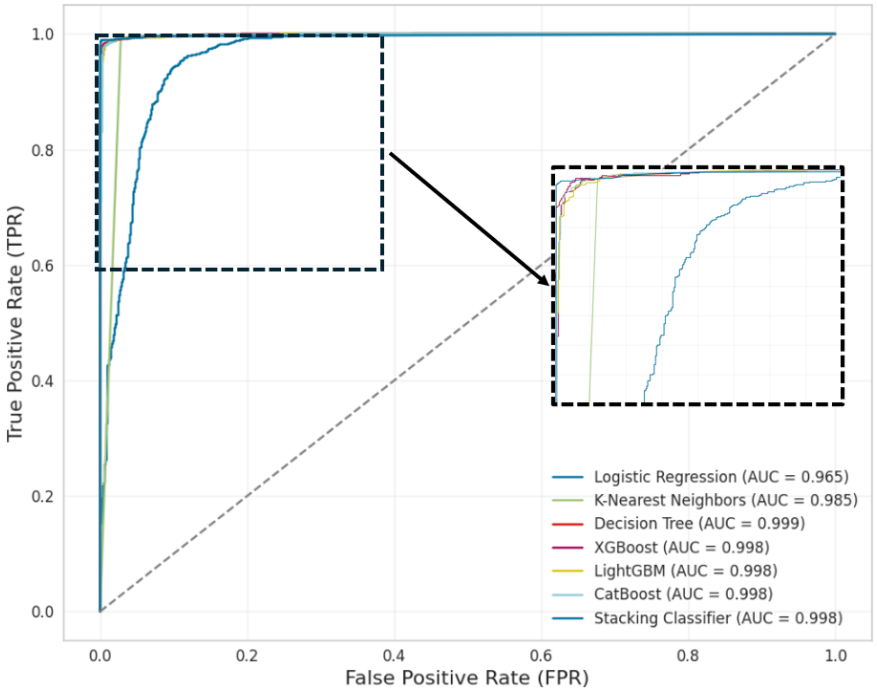

Figure 7. ROC curves for models on SMOTE-balanced dataset with AUC scores

Discussion

This study provides a comparative analysis of various machine learning models in predicting bankruptcy using datasets processed with different balancing techniques, including no balancing, Tomek Links, and SMOTE. The findings emphasize the influence of these techniques on model performance, measured through a range of metrics such as accuracy, F1 score, and AUC. The results contribute to understanding how data balancing impacts classification tasks, particularly in imbalanced datasets where predictive reliability is paramount.

The analysis revealed that SMOTE yielded the most favorable outcomes among the three approaches. Specifically, the application of SMOTE enhanced the performance of all models, as evidenced by an average accuracy increase of 2.6% compared to the imbalanced dataset and 4.7% compared to the Tomek Links-balanced dataset. These results align with existing literature, which suggests that synthetic oversampling can effectively mitigate class imbalance by generating representative data points without sacrificing original information. For instance, Zhao and Aumeboonsuke [37] demonstrated that SMOTE improves the predictive performance of classifiers by creating synthetic samples along the feature space between existing minority instances, thus reducing the bias towards majority classes. Conversely, balancing with Tomek Links resulted in a slight decline in accuracy, with an average reduction of 2.7% compared to the imbalanced dataset. Although Tomek Links can effectively remove overlapping or noisy samples to improve class separation, this method also reduces the dataset size, which may explain the observed decline in performance. These findings are not consistent with earlier studies, such as those by Swana et al. [38] and Pereira et al. [39], which reported that under-sampling techniques like Tomek Links can enhance model performance by removing noisy or overlapping samples, thereby improving class separability. However, in our study, balancing with Tomek Links resulted in a general decline in performance metrics, including an average accuracy reduction of 2.7% compared to the imbalanced dataset. One potential explanation for this discrepancy could be the characteristics of our dataset. Unlike the datasets used in prior studies, which may have contained a higher proportion of noisy or misclassified samples, our dataset might have had a relatively clean separation between classes, making the removal of data points through Tomek Links less beneficial and even detrimental. Additionally, the dimensionality and feature distributions of the dataset could influence the effectiveness of Tomek Links. For instance, if the features exhibit significant overlap or non-linearity, removing samples near class boundaries might disrupt critical decision boundaries for some classifiers. Another contributing factor could be the difference in the choice of classifiers. Previous studies often utilized models less sensitive to reductions in training data size, such as K-Nearest Neighbors or simpler decision trees. In contrast, the models used in our study, particularly ensemble and boosting-based classifiers, rely on diverse and extensive training data to construct solid predictions. The removal of samples through Tomek Links might therefore have hindered their ability to fully leverage the available data, leading to a decline in performance. Future studies could explore these factors more systematically to clarify the conditions under which Tomek Links proves advantageous.

The superior performance of ensemble-based models, particularly the Stacking Classifier and XGBoost, highlights their strength in handling imbalanced datasets and leveraging the benefits of synthetic data. The Stacking Classifier consistently achieved the highest accuracy and F1 score across all datasets, while XGBoost excelled under SMOTE balancing with near-perfect AUC values. This finding aligns with prior research, such as that by Chen and Guestrin [40], which demonstrated that gradient boosting frameworks like XGBoost are particularly effective in identifying complex patterns in imbalanced data due to their iterative learning process and ability to minimize classification errors. Furthermore, the ROC analysis underscores the reliability of ensemble and boosting models in distinguishing between bankruptcy and non-bankruptcy cases. High AUC values observed in Stacking Classifier and XGBoost models reflect their ability to balance sensitivity and specificity effectively. This observation supports earlier findings by Ribeiro and Reynoso-Meza [41], who identified ensemble methods as highly adaptable to imbalanced data scenarios, particularly when combined with data preprocessing techniques like SMOTE.

The statistical significance tests ($p < 0.001$ for SMOTE vs. Tomek Links) reinforce SMOTE's superiority, aligning with Aslan and Özüpak [42]. SHAP analysis further elucidates that features like "Net Income to Total Assets" are critical predictors, offering a deeper understanding of financial distress drivers. Fairness metrics indicate that SMOTE not only boosts accuracy but also mitigates bias, a finding not emphasized in prior studies like Swana et al. [38].

Conclusion

This study evaluated the influence of data balancing techniques on machine learning models for bankruptcy prediction. Various classifiers, including the Stacking Classifier and XGBoost, were applied to imbalanced data and datasets balanced using Tomek Links and SMOTE, with their performance assessed using a range of evaluation metrics. Dimensionality reduction using PCA was implemented to enhance computational efficiency and mitigate overfitting, ensuring that the models effectively captured the underlying data patterns. Additionally, hyperparameter optimization was conducted using Optuna, which systematically identified the best parameter configurations to maximize model performance. The results revealed distinct impacts of the balancing techniques, with SMOTE significantly enhancing model performance through synthetic sample generation, particularly when paired with ensemble and boosting methods like the Stacking Classifier and XGBoost. In contrast, Tomek Links often led to reduced performance due to the removal of potentially valuable data points. The findings highlight the importance of carefully selecting a balancing method to address class imbalance effectively. The Stacking Classifier and XGBoost demonstrated superior performance on the SMOTE-balanced dataset, capturing complex patterns in the data and achieving high predictive accuracy.

Future research could focus on developing hybrid approaches that combine the strengths of multiple balancing techniques, such as integrating SMOTE with advanced under-sampling methods to preserve data integrity while addressing class imbalance. Additionally, adaptive balancing strategies that dynamically adjust to the characteristics of the dataset, such as using reinforcement learning or meta-learning frameworks, could be explored. These methods may enable models to balance datasets more effectively by learning from the underlying data distribution and tailoring the balancing process accordingly. Furthermore, examining the interplay between balancing techniques and emerging machine learning architectures, such as transformers or deep ensembles, could provide valuable insights into optimizing predictions in highly imbalanced financial datasets.

From a Management Information Systems (MIS) perspective, this study contributes practically to bankruptcy prediction by providing a robust, data-driven framework that enhances decision-making in financial risk management. By leveraging advanced machine learning techniques like the Stacking Classifier and SMOTE, integrated with tools such as Optuna and PCA, this research offers MIS professionals actionable tools to develop predictive systems that can be embedded within enterprise resource planning (ERP) or financial management systems. These systems can proactively identify at-risk companies, enabling managers to implement timely interventions such as debt restructuring or operational improvements. Furthermore, the emphasis on interpretability (e.g., through SHAP values, as suggested earlier) ensures that these models provide transparent insights, aligning with MIS goals of supporting strategic planning and stakeholder communication. Ultimately, this work bridges the gap between technical prediction models and their practical deployment in business environments, enhancing organizational resilience and economic stability.

Ethics committee approval and conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared.

There is no conflict of interest with any person / institution in the article prepared.

Acknowledgment

An initial version of this research was showcased as a short abstract at the 3rd BİLSEL International Korykos Scientific Researches and Innovation Congress in 2024. This manuscript is an expanded version of that study,

providing a thorough analysis and extended discussions to offer a deeper investigation of the subject.

References

[1] T. J. Zywicki, "An economic analysis of the consumer bankruptcy crisis," *Nw. UL Rev.*, vol. 99, pp. 1463, 2004.

[2] E. I. Altman, "Predicting financial distress of companies: revisiting the Z-score and ZETA® models," in *Handbook of Research Methods and Applications in Empirical Finance*, Edward Elgar Publishing, 2013, pp. 428–456.

[3] M. K. Brunnermeier and Y. Sannikov, "A macroeconomic model with a financial sector," *American Economic Review*, vol. 104, no. 2, pp. 379–421, 2014.

[4] A. W. Lo and D. V. Repin, "The psychophysiology of real-time financial risk processing," *Journal of Cognitive Neuroscience*, vol. 14, no. 3, pp. 323–339, 2002.

[5] J. E. Stiglitz, "Reforming the global economic architecture: lessons from recent crises," *The Journal of Finance*, vol. 54, no. 4, pp. 1508–1521, 1999.

[6] V. Sinap, "Comparative performance analysis of machine learning algorithms in the retail industry: Black Friday sales forecasting," *Journal of Selçuk University Social Sciences Vocational School*, vol. 27, no. 1, pp. 65–90, 2024.

[7] J. Furman, J. E. Stiglitz, B. P. Bosworth, and S. Radelet, "Economic crises: evidence and insights from East Asia," *Brookings Papers on Economic Activity*, vol. 1998, no. 2, pp. 1–135, 1998.

[8] G. Allayannis and E. Ofek, "Exchange rate exposure, hedging, and the use of foreign currency derivatives," *Journal of International Money and Finance*, vol. 20, no. 2, pp. 273–296, 2001.

[9] A. A. Al-Mana, W. Nawaz, A. Kamal, and M. Koç, "Financial and operational efficiencies of national and international oil companies: An empirical investigation," *Resources Policy*, vol. 68, Art. no. 101701, 2020.

[10] W. S. Randall and M. T. Farris, "Supply chain financing: using cash-to-cash variables to strengthen the supply chain," *International Journal of Physical Distribution & Logistics Management*, vol. 39, no. 8, pp. 669–689, 2009.

[11] V. Sinap, "Comparative study of loan approval prediction using machine learning methods," *Gazi University Journal of Science Part C: Design and Technology*, vol. 12, no. 2, pp. 644–663, 2024.

[12] X. Ying, "An overview of overfitting and its solutions," in *Journal of Physics: Conference Series*, vol. 1168, Art. no. 022022, Feb. 2019.

[13] H. Ali, M. M. Salleh, R. Saedudin, K. Hussain, and M. F. Mushtaq, "Imbalance class problems in data mining: A review," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1560–1571, 2019.

[14] I. M. El-Hasnony, S. I. Barakat, M. Elhoseny, and R. R. Mostafa, "Improved feature selection model for big data analytics," *IEEE Access*, vol. 8, pp. 66989–67004, 2020.

[15] M. Singla, K. S. Gill, M. Kumar, R. Rawat, and S. Aluvala, "Incorporating the Catboost classification method in machine learning applications for SMOTE analysis and bankruptcy data equalisation," in *2024 International Conference on E-Mobility, Power Control and Smart Systems (ICEMPS)*, Apr. 2024, pp. 1–5.

[16] T. K. Chen, H. H. Liao, G. D. Chen, W. H. Kang, and Y. C. Lin, "Bankruptcy prediction using machine learning models with the text-based communicative value of annual reports," *Expert Systems with Applications*, vol. 233, Art. no. 120714, 2023.

[17] L. Papíková and M. Papík, "Effects of classification, feature selection, and resampling methods on bankruptcy prediction of small and medium-sized enterprises," *Intelligent Systems in Accounting, Finance and Management*, vol. 29, no. 4, pp. 254–281, 2022.

[18] S. Shetty, M. Musa, and X. Brédart, "Bankruptcy prediction using machine learning techniques," *Journal of Risk and Financial Management*, vol. 15, no. 1, Art. no. 35, 2022.

[19] N. Radwan *et al.*, "An intelligent approach for predicting bankruptcy empowered with machine learning technique," in *2022 International Conference on Cyber Resilience (ICCR)*, Oct. 2022, pp. 1–5.

[20] H. Kim, H. Cho, and D. Ryu, "Corporate bankruptcy prediction using machine learning methodologies with a focus on sequential data," *Computational Economics*, vol. 59, no. 3, pp. 1231–1249, 2022.

[21] M. S. Keya *et al.*, "Comparison of different machine learning algorithms for detecting bankruptcy," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Jan. 2021, pp. 705–712.

[22] T. Le, "A comprehensive survey of imbalanced learning methods for bankruptcy prediction," *IET Communications*, vol. 16, no. 5, pp. 433–441, 2022.

[23] University of California, Irvine, "Taiwanese bankruptcy prediction," *UCI Machine Learning Repository*, 2020. [Online]. Available: https://archive.ics.uci.edu/dataset/572/taiwanese+bankruptcy+prediction

[24] S. Kappal, "Data normalization using median median absolute deviation MMAD based Z-score for robust predictions vs. min–max normalization," *London Journal of Research in Science: Natural and Formal*, vol. 19, no. 4, pp. 39–44, 2019.

[25] 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.

[26] Q. Leng *et al.*, "OBMI: Oversampling borderline minority instances by a two-stage Tomek link-finding procedure for class imbalance problem," *Complex & Intelligent Systems*, vol. 10, pp. 4775–4792, 2024.

[27] I. D. Mienye and Y. Sun, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022.

[28] Y. Jung, "Multiple predicting K-fold cross-validation for model selection," *Journal of Nonparametric Statistics*, vol. 30, no. 1, pp. 197–215, 2018.

[29] W. Jia, M. Sun, J. Lian, and S. Hou, "Feature dimensionality reduction: a review," *Complex & Intelligent Systems*, vol. 8, no. 3, pp. 2663–2693, 2022.

[30] S. Tangirala, "Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, pp. 612–619, 2020.

[31] S. S. Dhaliwal, A. A. Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Information*, vol. 9, no. 7, Art. no. 149, 2018.

[32] N. Javaid *et al.*, "Employing a machine learning boosting classifiers based stacking ensemble model for detecting non-technical losses in smart grids," *IEEE Access*, vol. 10, pp. 121886–121899, 2022.

[33] D. D. Rufo, T. G. Debelee, A. Ibenthal, and W. G. Negera, "Diagnosis of diabetes mellitus using gradient boosting machine (LightGBM)," *Diagnostics*, vol. 11, no. 9, Art. no. 1714, 2021.

[34] S. Uddin *et al.*, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Scientific Reports*, vol. 12, no. 1, Art. no. 6256, 2022.

[35] K. Kirasich, T. Smith, and B. Sadler, "Random forest vs logistic regression: binary classification for heterogeneous datasets," *SMU Data Science Review*, vol. 1, no. 3, Art. no. 9, 2018.

[36] S. Hanifi, A. Cammarono, and H. Zare-Behtash, "Advanced hyperparameter optimization of deep learning models for wind power prediction," *Renewable Energy*, vol. 221, Art. no. 119700, 2024.

[37] Z. Zhao and V. Aumeboonsuke, "Imbalanced credit risk prediction in ensemble learning classifiers: A comparative analysis of SMOTE, ADASYN, SMOTETomek, and cluster centroids," *Journal of Arts Management*, vol. 7, no. 3, pp. 959–984, 2023.

[38] E. F. Swana, W. Doorsamy, and P. Bokoro, "Tomek link and SMOTE approaches for machine fault classification with an imbalanced dataset," *Sensors*, vol. 22, no. 9, Art. no. 3246, 2022.

[39] R. M. Pereira, Y. M. Costa, and C. N. Silla Jr, "MLTL: A multi-label approach for the Tomek Link undersampling algorithm," *Neurocomputing*, vol. 383, pp. 95–105, 2020.

[40] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[41] V. H. A. Ribeiro and G. Reynoso-Meza, "Ensemble learning by means of a multi-objective optimization design approach for dealing with imbalanced data sets," *Expert Systems with Applications*, vol. 147, Art. no. 113232, 2020.

[42] E. Aslan and Y. Özüpak, "Comparison of machine learning algorithms for automatic prediction of Alzheimer's disease," *Journal of the Chinese Medical Association*, vol. 88, no. 2, pp. 98–107, 2025.