JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

Comparison of SMOTE and ADASYN in Optimizing Random Forest Model for Imbalanced Financial Ratio Bankruptcy Prediction

Novanda Rizky Ramadhana[1*], Fuad Muhajirin Farid[2], Yeni Rahkmawati[3]
[1,2,3]Statistika, Universitas Lambung Mangkurat, Indonesia
[1]novandaazzz@gmail.com, [2]fuad.farid@ulm.ac.id , [3]yeni.rahkmawati@ulm.ac.id

Abstract

Classification is a data analysis process that can predict classes based on predefined characteristics. In the era of big data, classification can be performed using machine learning. The problem of machine learning in classification analysis is imbalance data which often affect model performance. SMOTE and ADASYN are oversampling techniques to solve this problem. This study aims to evaluate the effectiveness of SMOTE and ADASYN in improving the performance of the Random Forest model on imbalanced data in the case of company bankruptcy using financial ratios. Models were built using training data with various splitting data and oversampling techniques. Then, the resulting models will be tested using testing data. The results show that the best model was achieved with a combination of splitting data 70:30 using SMOTE technique, which produced the highest f1-score of 40.57%, compared to ADASYN technique with 36.11% (a decrease of 4.46%), and without oversampling techniques with 19.51% (a decrease of 21.06%). The findings indicate SMOTE and ADASYN can identify minority values which are the main problem of imbalance data, with SMOTE showing better performance compared to ADASYN. This study contributes empirical insights on the effectiveness of SMOTE and ADASYN in handling imbalanced data for corporate bankruptcy prediction based on financial ratios.

Keywords: SMOTE; ADASYN; Random Forest, Company Bankruptcy, Financial Ratios

1. INTRODUCING

The increasing number of corporate failures indicates that bankruptcy has become a crucial issue, given its broad impact on employment, social welfare, and overall economic stability [1]. Bankruptcy occurs when a company fails to meet its financial obligations, leading to liquidity problems as an early warning sign. A firm is considered bankrupt when its long-term returns fall below its total costs, and prolonged financial distress threatens its continuity as liabilities exceed its assets [2]. Therefore, early bankruptcy prediction is crucial for stakeholders to prevent losses, with financial information serving as a key indicator of a company's financial health and bankruptcy risk.

To address this issue, the variables used in this study are financial ratios, which are commonly used to assess the risk of corporate bankruptcy [3]. The financial ratios are organized based on their purpose, such as liquidity, solvency, activity, and profitability ratios [4].

This study uses the Company Bankruptcy Prediction dataset from Taiwan Economic Journal (1999–2009), containing financial variables to predict company bankruptcy. The dataset is widely used in financial risk analysis research because it provides comprehensive and structured financial ratio data suitable for developing and evaluating machine learning models.

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

Building upon the financial variables obtained from the dataset, this study applies classification methods to distinguish between bankrupt and non-bankrupt companies. Classification is a method used to determine which category an object belongs to by comparing it with existing data. It involves creating a model or function that assigns data points to specific groups based on their features [5][6]. This technique is widely used in various fields such as agriculture, finance, healthcare, and business [7][8][9][10].

Along with technological advances, classification increasingly uses Artificial Intelligence (AI), particularly Machine Learning (ML), which applies supervised and unsupervised algorithms to recognize patterns, make predictions, and support automated decision-making [11]. In this research, the classification is performed using the Random Forest algorithm. The Random Forest algorithm can lessen overfitting and resilient to outliers and missing data [7][12][13]. However, imbalance data can reduce prediction accuracy, as models tend to favor the majority class over the minority class [14][15].

To fix this issue, resampling methods are employed to balance the classes before training the model. Balanced data has equal numbers of majority and minority classes, while slightly imbalanced data might have ratios like 55:45, 60:40, or 70:30 [16]. There are three main resampling strategies: under-sampling, over-sampling, and a mix of both. Popular over-sampling techniques include SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling Approach) [17]. Research has shown that these methods can significantly improve the performance of machine learning models when dealing with imbalanced data [18][19][20]. Both SMOTE and ADASYN help by generating new samples for the minority class, thus enchancing classification accuracy.

This research aims to evaluate and compare the effectiveness of SMOTE and ADASYN in improving the performance of the Random Forest model for classifying corporate bankruptcy based on financial ratios.

2. METHODOLOGY

2.1 Research Process

The process of this study can be seen in the flowchart presented in Figure 1 below.

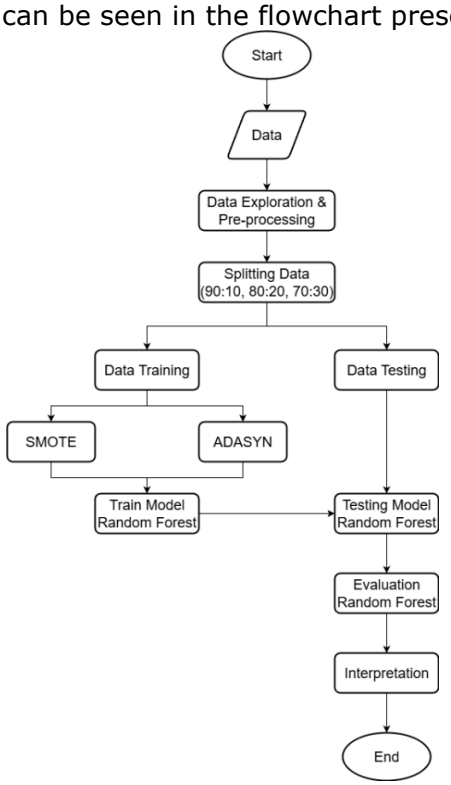

Figure 1. Flowchart of Research Process

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

From Figure 1, the research workflow begin with collecting financial ratio data from Kaggle's Company Bankruptcy Prediction dataset, followed by exploring and preprocessing, splitting into training and testing sets (80:20, 70:30, 90:10), applying SMOTE and ADASYN oversampling for *training* data, trained and tested a Random Forest model, evaluating performance using accuracy, precision, recall, and F1-score, compared SMOTE and ADASYN results, and drew conclusions with recommendations.

2.2 Random Forest

Random Forest (RF) is a machine learning technique that combines multiple decision trees by randomly selecting samples and features to reduce correlation between trees, minimize overfitting, and improve model accuracy [13]. Node splits are determined by testing binary splits for each predictor, with numeric variables divided at midpoint [21]:

$$X \leq c \tag{1}$$

If the data meets the criterion, it goes to the left subnode; otherwise, to the right subnode.

Common split criteria are Entropy and Gini Index. Gini Index measures node purity, with 0 indicating the purest split. For dataset L with classes j, it is defined as [22]:

$$GINI(L) = 1 - \sum_{i=1}^{j} p_i^2 \tag{2}$$

When L is split by feature Ainto $L_1$and L_2, the weighted Gini is:

$$GINI_A(L) = \frac{N_1}{N} GINI(L_1) + \frac{N_2}{N} GINI(L_2) \tag{3}$$

The impurity reduction is:

$$\Delta GINI(A) = GINI(L) - GINI_A(L) \tag{4}$$

A higher impurity reduction indicates a better split.

The following figure 2 is a simplified example of a Random Forest.

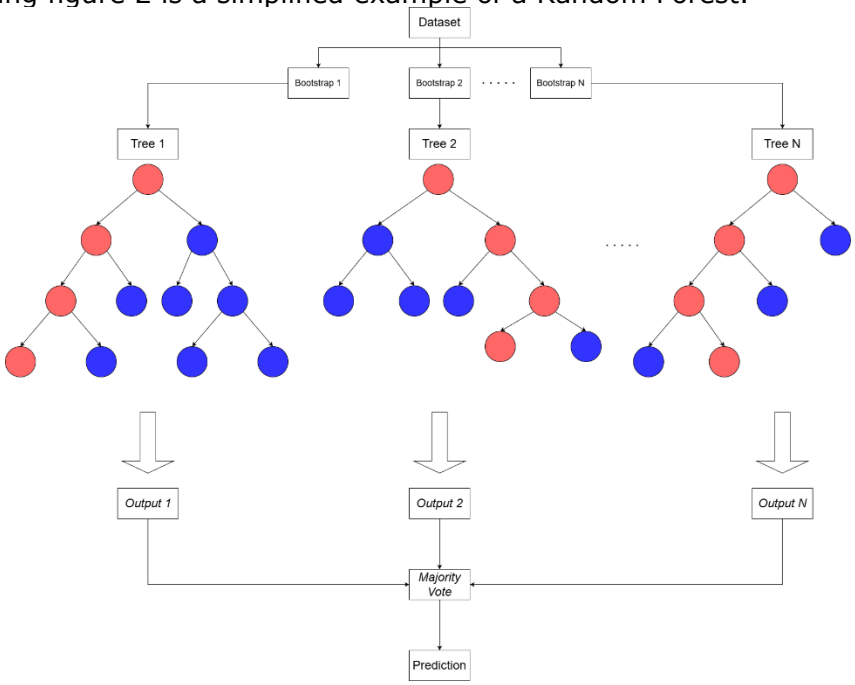

Figure 2. Random Forest Tree Illustration

From Figure 2, the procedure of the Random Forest algorithm involves bootstrap resampling N times with replacement to create datasets, building classification trees by randomly selecting features (commonly $\sqrt{m}$) at each node, making predictions for each

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

tree, repeating this process until N trees are generated, and combining their predictions through majority voting [23].

2.3 SMOTE

SMOTE is an oversampling method that generates synthetic minority samples using k-nearest neighbors and Euclidean distance [24]. Although SMOTE reduces excessive overfitting, it may still cause similarity among duplicated samples [25]. The Euclidean distance is calculated as follows [26].

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (5)$$

where:

$d(x,y)$ = Euclidean distance
n = Number of data points
i = Attribute index
x_i = The i-th observation selected from the minority class
y_i = The nearest i-th observation from the minority class

The generation of synthetic data using SMOTE is as follows.

1. Select a minority data instance denoted as x_i.
2. Calculate the nearest minority neighbor of x_i using the Euclidean distance in (5).
3. Generate synthetic SMOTE data using the following formula [27].

$$x_{syn} = x_i + (x_{knn} - x_i) \times \gamma \quad (6)$$

where:

x_{syn} = Synthetic sample generated
x_i = The i-th minority class observation
x_{knn} = The nearest minority class neighbor of x_i
γ = A random number between 0 and 1

2.4 ADASYN

ADASYN is an oversampling technique that adds minority samples based on learning difficulty, using parameters β for balance and d_{th} for imbalance tolerance. The procedure for generating synthetic data using ADASYN is as follows [27].

1. Determine ADASYN parameters β and d_{th}. If $0 < \beta < 1$, the generated synthetic data will be less than the majority class; if $\beta = 1$, both classes will be balanced. d_{th} is the maximum imbalance ratio.
2. Calculate the balance degree:

$$d = \frac{m_s}{m_l} \quad (7)$$

where:

d = Balance degree
m_s = Minority samples
m_l = Majority sample

Continue if $d < d_{th}$

3. Compute the number of synthetic samples:

$$G = (m_l - m_s) \times \beta \quad (8)$$

where:

G = Number of generated synthetic samples
β = Balance level parameter with a value between 0 and 1

4. Calculate the ratio using K-Nearest Neighbor and Euclidean distance:

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

$$r_i = \frac{\triangle_i}{k} \tag{9}$$

where:
r_i = Density distribution ratio of the i-th instance
$\triangle_i$ = Number of nearest majority samples to the i-th minority sample
k = Total number of nearest neighbors selected

5. Normalize r_i:

$$\hat{r_i} = \frac{r_i}{\sum_{i=1}^{m_s} r_i} \tag{10}$$

where:
$\hat{r_i}$ = Normalized density distribution ratio of the i-th instance
r_i = Density distribution ratio of the i-th instance

6. Determine the number of synthetic samples for each minority instance:

$$g_i = \hat{r_i} \times G \tag{11}$$

where:
g_i = Number of synthetic samples to be generated
$\hat{r_i}$ = Normalized density distribution ratio of the i-th instance
G = Total number of synthetic samples generated

7. Generate synthetic data using (6) for each g_i.

2.5 Confusion Matrix

A Confusion Matrix is a tool used to evaluate the performance of a classification model after the data mining process. It provides a comparison between the predicted classifications produced by the model and the actual classifications [28].

The confusion matrix table is typically presented as shown in the following Table 1 [29].

Table 1. The Structure of the Confusion Matrix

		Prediction Class	
Actual Class		1	0
	1	TP *(True Positive)*	FN *(False Negative)*
	0	FP *(False Positive)*	TN *(True Negative)*

where:
1 = Positive class
0 = Negative class
TP (*True Positive*) = Number of positive instances correctly predicted
TN (*True Negative*) = Number of negative instances correctly predicted
FP (*False Positive*) = Number of negative instances incorrectly predicted as positive
FN (*False Negative*) = Number of positive instances incorrectly predicted as negative

The confusion matrix is used to evaluate model performance using accuracy, precision, recall, and F1-score [28]. However, in cases of imbalanced data, the F1-score is more suitable for evaluating model performance [30].

1. Accuracy

Accuracy measures how well the model classifies correctly, as shown in the following formula.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{12}$$

2. Precision

Precision measures how accurately the model predicts positive classes, as shown in the following formula.

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

$$Precision = \frac{TP}{(TP + FP)} \quad (13)$$

3. Recall

Recall measures the model's ability to correctly identify all actual positive cases, as shown in the following formula.

$$Recall = \frac{TP}{(TP + FN)} \quad (14)$$

4. F1-Score

The F1 Score measures the balance between precision and recall and serves as the main evaluation metric in this study. The formula is shown below.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (15)$$

3. RESULT AND DISCUSSIONS

The data were obtained from the Taiwan Economic Journal for the period 1999–2009. The dataset contains 6,819 company samples with 96 attribute variables. The variables used in this study are Bankrupt, Current Ratio, Quick Ratio, Debt Ratio, Net Income to Total Assets, Net Income to Stockholder's Equity, and Total Asset Turnover.

3.1 Data Exploration and Preprocessing

First, the frequency of each class is checked using a bar chart to determine whether the data are balanced or not. This step uses the Seaborn and Matplotlib libraries for visualization. The bar chart shows extreme class imbalance 6599 non-bankrupt (96.7%) and 220 bankrupt (3.3%) companies. The data were checked for missing values, duplicates, and outliers. Outliers in Current and Quick Ratios (values >1) were removed, and the distribution was reviewed again using the bar chart.

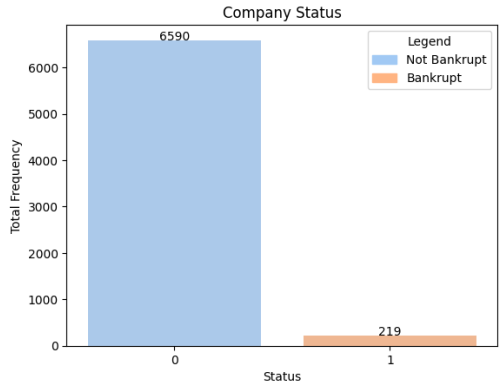

Figure 3. Company Status After Outlier Handling

Based on Figure 3, total of 10 observations are removed from the dataset, consiting of 9 non-bankrupt (0) and 1 bankrupt (1) observations. These observations were identified as abnormal values because they did not meet the characteristics of financial ratio data. For this reason, retaining those values could affect the performance and reliability of the model. Therefore, those values were excluded from further analysis.

3.2 Splitting Data

For modeling, the data were split into training and testing sets with ratios of 90:10, 80:20, and 70:30. The following table 2 shows the data frequencies for each split.

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

Table 2. Data Frequency by Split Ratios

Splitting Data	Total Training Data			Total Testing Data		
	0	1	Total	0	1	Total
90:10	5931	197	6128	659	22	681
80:20	5272	175	5477	1318	44	1362
70:30	4613	153	4766	1977	66	2043

Table 2 shows the data distribution for each split. The dataset was split into training and testing sets using 70:30, 80:20, and 90:10 ratios, with the training data used to build the Random Forest model and testing data used for evaluation

3.3 SMOTE and ADASYN

SMOTE and ADASYN are applied to the training data to balance minority classes, improving Random Forest performance. SMOTE generates samples evenly, while ADASYN focuses on minorities near the majority class.

1. SMOTE

SMOTE is an oversampling technique used to increase minority class samples to match or approach the majority class. It generates synthetic data based on randomly selected minority instances, their nearest neighbors, and a random gap γ (0–1). The nearest neighbors are calculated using Euclidean distance.

Table 3. Ilustration of Selected Data, Neighbors, Synthetic SMOTE, and Gap = 0.2848 Based on Python

Index	Data	Current Ratio	Quick Ratio	Debt Ratio	Net Income to Total Assets	Net Income to Stockholder's Equity	Total Aset Turnover
3452	Selected	0.0071	0.0061	0.194	0.7851	0.8386	0.3133
2017	Neighbor	0.0066	0.0049	0.2176	0.735	0.8295	0.2654
8765	Synthetic	0.007	0.0058	0.2007	0.7708	0.836	0.2997

As an example, a minority instance is selected randomly and synthetic samples are generated using its nearest neighbors, illustrated for the 90:10 data split (see Table 3).The following illustrates the manual procedure for generating synthetic data using SMOTE.

1) Selecting a minority instance at index 3452, denoted as x_i.
2) Next, the nearest minority neighbor of x_i (index 2017) was identified using Euclidean distance as defined in (5).

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d(x,y) = \sqrt{(0.0071-0.0066)^2 + (0.0061-0.0049)^2 + (0.194-0.2176)^2}$$
$$\sqrt{+(0.7851-0.735)^2 + (0.8386-0.8295)^2 + (0.3133-0.2654)^2}$$

$$d(x,y) = \sqrt{0.0000002 + 0.000001 + 0.0006 + 0.0025 + 0.00008 + 0.0023} = \sqrt{0.0054} = 0.0738$$

3) Then, generate the synthetic data using (6).

$$x_{syn} = x_i + (x_{knn} - x_i) \times \gamma$$

$$x_{syn} = (0.0071,0.0061,0.194,0.7851,0.8386,0.3133) +$$
$$((0.0066,0.0049,0.2176,0.735,0.8295,0.2654) -$$
$$(0.0071,0.0061,0.194,0.7851,0.8386,0.3133)) \times 0.2848$$

$$x_{syn} = (0.007,0.0058,0.2007,0.7708,0.836,0.2997)$$

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

The manually generated synthetic data will be used for Random Forest-SMOTE modeling. The process was repeated for other instances to balance the training data for all splits (90:10, 80:20, 70:30).

2. ADASYN

Unlike SMOTE, ADASYN generates synthetic data near the majority class by selecting minority instances closest to it and their nearest neighbors.

Table 4. Ilustration of Selected Data, Neighbors, Synthetic ADASYN, and Gap = 0.4972 Based on Python

Index	Data	Current Ratio	Quick Ratio	Debt Ratio	Net Income to Total Assets	Net Income to Stockholder's Equity	Total Aset Turnover
424	Selected	0.0071	0.0061	0.194	0.7851	0.8386	0.3133
2017	Neighbor	0.0065	0.0029	0.1905	0.7963	0.8401	0.3133
8013	Synthetic	0.0068	0.0045	0.1923	0.7906	0.8393	0.3133

As an example, a minority instance is selected randomly and synthetic samples are generated using its nearest neighbors, illustrated for the 90:10 data split (see Table 4). The following illustrates the manual procedure for generating synthetic data using ADASYN.

1) Set the balance level parameter (β = 1) and maximum imbalance ratio (d_{th} = 0.4)
2) Calculate the balance degree using (7).

$$d = \frac{m_s}{m_l} = \frac{197}{5931} = 0.0332$$

3) Calculate the number of synthetic samples to generate using (8).

$$G = (m_l - m_s) \times \beta = (5931 - 197) \times 1 = 5734$$

4) Calculate the ratio using K-Nearest Neighbors with euclidean distance, as defined in (9).

$$r_i = \frac{\triangle_i}{k} = \frac{4}{5} = 0.8$$

5) Normalize r_i using (10).

$$\hat{r_i} = \frac{r_i}{\sum_{i=1}^{m_s} r_i} = \frac{0.8}{194} = 0.0041$$

6) Calculate the number of synthetic samples to generate for each minority instance using (11).

$$g_i = \hat{r_i} \times G = 0.0041 \times 5734 = 23.5094 \approx 24$$

7) Generate g_i synthetic samples using (6).

$$x_{syn} = x_i + (x_{knn} - x_i) \times \gamma$$

$$x_{syn} = (0.0071,0.0061,0.194,0.7851,0.8386,0.3133) + ((0.0065,0.0029,0.1905,0.7963,0.8401,0.3133) - (0.0071,0.0061,0.194,0.7851,0.8386,0.3133)) \times 0.4972$$

$$x_{syn} = (0.0068,0.0045,0.1923,0.7907,0.8393,0.3133)$$

The manually generated synthetic data will be used for Random Forest-ADASYN modeling. The process was repeated for other minority instances to balance the training data for all splits (90:10, 80:20, 70:30). Table 5 shows the total minority and majority samples before and after oversampling.

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

Table 5. Training Data: Original vs SMOTE vs ADASYN

Splitting Data	Original			SMOTE			ADASYN		
	0	1	Total	0	1	Total	0	1	Total
90:10	5931	197	6128	5931	5931	11862	5931	5917	11848
80:20	5272	175	5447	5272	5272	10544	5272	5281	10553
70:30	4613	153	4766	4613	4613	9266	4613	4622	9235

3.4 Building the Random Forest Model

To build the Random Forest model, 100 bootstrap samples were first generated with replacement, allowing duplicates. Decision trees were then created for each bootstrap, with $\sqrt{6} \approx 2$ features randomly selected per split. Node purity was calculated using the Gini index (2–4). Figure 3 shows an example tree for the 90:10 split without oversampling.

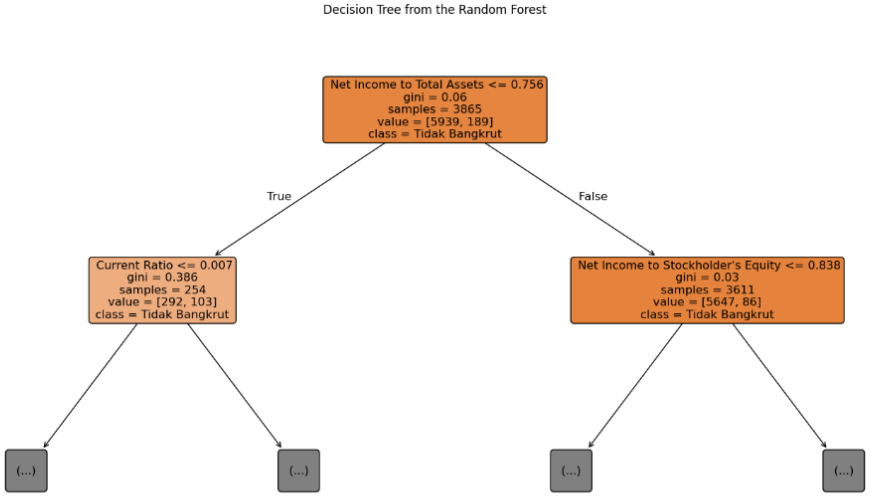

Figure 4. Random Forest Decision Tree Visualization Illustration

Figure 4 shows one decision tree, consisting of a root node (node 0) and child nodes (left node 1, right node 2) used for manual Gini index calculation. The root node feature is Net Income to Total Assets with a split at 0.756. The Gini index is 0.0597, with 5939 majority (0) and 189 minority (1) samples.

Calculated Gini for Root Node:

$$p_0 = \frac{5939}{6128} = 0.9692, p_1 = \frac{189}{6128} = 0.0308$$

$$GINI(L) = 1 - \sum_{i=1}^{j} p_i^2 = 1 - (0.9692)^2 - (0.0308)^2 = 0.0597$$

Calculated Gini for Node 1 and Node 2:

Gini for Node 1:

$$p_0 = \frac{292}{395} = 0.7392, p_1 = \frac{103}{395} = 0.2608$$

$$GINI(L_1) = 1 - \sum_{i=1}^{j} p_i^2 = 1 - (0.7392)^2 - (0.2608)^2 = 0.3856$$

Gini for Node 2:

$$p_0 = \frac{5647}{5733} = 0.985, p_1 = \frac{86}{5733} = 0.015$$

$$GINI(L_2) = 1 - \sum_{i=1}^{j} p_i^2 = 1 - (0.985)^2 - (0.015)^2 = 0.0296$$

Weighted Gini:

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

$$GINI_A(L) = \frac{N_1}{N} GINI(L_1) + \frac{N_2}{N} GINI(L_2) = \left(\frac{395}{6128} \times 0.3856\right) + \left(\frac{5733}{6128} \times 0.0296\right) = 0.0525$$

Gini Reduction:

$$\Delta GINI(A) = GINI(L) - GINI_A(L) = 0.0597 - 0.0525 = 0.0072$$

These steps are repeated recursively until leaf nodes reach a Gini index of 0. This process is used to create 100 decision trees for the Random Forest with default tree depth.

3.5 Model Evaluation

The trained model is tested on the testing data, with predictions aggregated using majority vote. Performance is then evaluated using a confusion matrix to calculate accuracy, recall, precision, and F1-score.

Table 6. Evaluation Metrics Based on Data Splitting and Oversampling Treatment

Splitting Data	Oversampling Technique	Accuracy	Precision	Recall	F1-Score
90:10	SMOTE	93.83%	28.26%	59.09%	38.24%
	ADASYN	92.95%	25 %	59.09%	35.14%
	Without Oversampling	96.33%	33.33%	13.64%	19.36%
80:20	SMOTE	94.35%	30.59%	59.09%	40.31%
	ADASYN	93.76%	27.47%	56.82%	37.04%
	Without Oversampling	96.92%	56.25%	20.45%	30%
70:30	SMOTE	93.83%	29.45%	65.15%	40.57%
	ADASYN	93.25%	26%	59.09%	36.11%
	Without Oversampling	96.77%	50%	12.12%	19.51%

Based on Table 6, SMOTE and ADASYN improved the model's recall and F1-score, enhancing the classification of the minority class, but slightly reduced accuracy and precision. The best model was obtained with a 70:30 split using SMOTE, achieving the highest F1-score and recall at 40.57% and 65.15%, while accuracy and precision decreased by 2.94% and 20.55%.

4. CONCLUSION

The evaluation shows that SMOTE and ADASYN improved recall and F1-score, meaning the model better identifies and predicts bankrupt companies compared to no oversampling. The highest F1-score was 40.57% with a 70:30 split using SMOTE. However, precision and accuracy decreased, causing a slight drop in overall prediction accuracy and a tendency to misclassify non-bankrupt companies as bankrupt. Overall, SMOTE and ADASYN effectively address data imbalance, with SMOTE performing better than ADASYN for predicting bankruptcy using financial ratios. These findings provide empirical insights into the effectiveness of SMOTE and ADASYN in handling imbalanced data for corporate bankruptcy prediction based on financial ratios, providing useful information for future research of machine learning models.

5. REFERENCES

[1] V. E. Syukrina Janrosl, A. Putra Prima, P. Studi Akuntansi, F. Ilmu Sosial dan Humaniora, U. Putera Batam, and S. Galileo, "Potensi Kebangkrutan Menggunakan

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

Model Zavgren Dan Altman Pada Perusahaan Di Indonesia," *Measurement: Jurnal Akuntansi*, vol. 16, no. 2, pp. 159–165, 2022.
[2] Reskianty, "Implementasi Metode Support Vector Machine Dan Random Forest Untuk Dataset Tidak Seimbang (Studi Kasus: Klasifikasi Kebangkrutan Perusahaan)," Universitas Hasanuddin, Makassar, 2022.
[3] A. Kurniadi, "Analisis Rasio Keuangan Untuk Memprediksi Financial Distress Perusahaan Manufaktur Di BEI," *Jurnal Ilmiah Manajemen Kesatuan*, vol. 9, no. 3, pp. 495–508, Dec. 2021, doi: 10.37641/jimkes.v9i3.511.
[4] B. G. Putri and S. Munfaqiroh, "Analisis Rasio Keuangan Untuk Mengukur Kinerja Keuangan," *INSPIRASI: Jurnal Ilmu-Ilmu Sosial*, vol. 17, no. 1, pp. 214–226, 2020.
[5] N. P. Aldy, "Pendekatan Algoritma Cost Sensitive Decision Tree Pada Klasifikasi Film Berdasarkan Perolehan Kompilasi Dari Internet Movie Database (IMDB)," Universitas Lambung Mangkurat, Banjarbaru, 2024.
[6] I. Hayati, "Klasifikasi Mahasiswa Berpotensi Drop Out Menggunakan Algoritma Decision Tree C4.5 Dan Naive Bayes Di Universitas Jambi," Universitas Jambi, Jambi, 2021.
[7] Ary Prandika Siregar, Dwi Priyadi Purba, Jojor Putri Pasaribu, and Khairul Reza Bakara, "Implementasi Algoritma Random Forest Dalam Klasifikasi Diagnosis Penyakit Stroke," *Jurnal Penelitian Rumpun Ilmu Teknik*, vol. 2, no. 4, pp. 155–164, Nov. 2023, doi: 10.55606/juprit.v2i4.3039.
[8] R. Hariyanto and A. A. Widodo, "Klasifikasi Hasil Prediksi Panen Padi Berdasarkan Fisiologis Menggunakan Metode Nãive Bayes Classification" *Conference on Innovation and Application of Science and Technology (CIASTECH 2019)*, pp. 237–244, 2019.
[9] O.- Pahlevi, A.- Amrin, and Y.- Handrianto, "Implementasi Algoritma Klasifikasi Random Forest Untuk Penilaian Kelayakan Kredit," *Jurnal Infortech*, vol. 5, no. 1, pp. 71–76, Jun. 2023, doi: 10.31294/infortech.v5i1.15829.
[10] I. Sulistiani, E. Mufida, P. M. Yasser, and L. Alamsyah, "Systematic Literature Review: Bankruptcy Prediction Menggunakan Teknik Machine Learning dan Deep Learning," *INTECH*, vol. 2, no. 1, pp. 13–18, Jun. 2021, doi: 10.54895/intech.v2i1.824.
[11] R. G. Wardhana, G. Wang, and F. Sibuea, "Penerapan Machine Learning Dalam Prediksi Tingkat Kasus Penyakit Di Indonesia," *Journal of Information System Management (JOISM)*, vol. 5, no. 1, pp. 40–45, Jul. 2023, doi: 10.24076/joism.2023v5i1.1136.
[12] H. Marlina, Elmayati, A. Zulius, and H. O. L. Wijaya, "Penerapan Algoritma Random Forest Dalam Klasifikasi Penjurusan di SMA Negeri Tugumulyo," *Brahmana: Jurnal Penerapan Kecerdasan Buatan*, vol. 4, no. 2, pp. 138–143, 2023.
[13] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," *Babylonian Journal of Machine Learning*, vol. 2024, pp. 69–79, Jun. 2024, doi: 10.58496/BJML/2024/007.
[14] R. D. Fitriani, H. Yasin, and T. Tarno, "Penanganan Klasifikasi Kelas Data Tidak Seimbang Dengan Random Oversampling Pada Naive Bayes (Studi Kasus: Status Peserta KB IUD di Kabupaten Kendal)," *Jurnal Gaussian*, vol. 10, no. 1, pp. 11–20, 2021.
[15] R. Siringoringo, "Klasifikasi Data Tidak Seimbang Menggunakan Algoritma Smote Dan k-Nearest Neighbor," *Journal Information System Development (ISD)*, vol. 3, no. 1, pp. 44–49, 2018.
[16] M. H. A. Hamid, M. Yusoff, and A. Mohamed, "Survey on Highly Imbalanced Multi-class Data," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, 2022, doi: 10.14569/IJACSA.2022.0130627.
[17] A. Indrawati, H. Subagyo, A. Sihombing, W. Wagiyah, and S. Afandi, "Analyzing The Impact Of Resampling Method For Imbalanced Data Text In Indonesian Scientific

Novanda Rizky Ramadhana: *Corresponding Author

JURNAL TEKNOINFO
Volume 20, Number 1, January 2026, Page 81-92
E-ISSN 2615-224X P-ISSN 1693-0010
https://publikasi.teknokrat.ac.id/index.php/teknoinfo/index
DOI: https://doi.org/10.33365/teknoinfo.v20i1.1056

Articles Categorization," *BACA: JURNAL DOKUMENTASI DAN INFORMASI*, vol. 41, no. 2, p. 133, Dec. 2020, doi: 10.14203/j.baca.v41i2.702.
[18] C. Agustina and E. Rahmawati, "Optimalisasi Algoritma Random Forest Menggunakan SMOTE untuk Prediksi Pembatalan Tamu Hotel," *EVOLUSI : Jurnal Sains dan Manajemen*, vol. 12, no. 2, Sep. 2024, doi: 10.31294/evolusi.v12i2.23149.
[19] M. I. Anugrah, J. Zeniarja, and D. S. Setiawan, "Peningkatan Performa Model Hard Voting Classifier dengan Teknik Oversampling ADASYN pada Penyakit Diabetes," *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 1, pp. 290–299, Jun. 2024, doi: 10.29408/edumatic.v8i1.25838.
[20] E. Erlin, Y. Desnelita, N. Nasution, L. Suryati, and F. Zoromi, "Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak seimbang," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 3, pp. 677–690, Jul. 2022, doi: 10.30812/matrik.v21i3.1726.
[21] W.-Y. Loh and Y. Shih, "Split Selection Methods for Classification Trees," *Stat Sin*, pp. 815–840, 1999.
[22] 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, 2020, doi: 10.14569/IJACSA.2020.0110277.
[23] S. Mahmuda, "Implementasi Metode Random Forest pada Kategori Konten Kanal Youtube," *JURNAL JENDELA MATEMATIKA*, vol. 2, no. 01, pp. 21–31, Jan. 2024, doi: 10.57008/jjm.v2i01.633.
[24] E. Sutoyo and M. A. Fadlurrahman, "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 6, no. 3, p. 379, Dec. 2020, doi: 10.26418/jp.v6i3.42896.
[25] A. N. Kasanah, M. Muladi, and U. Pujianto, "Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 2, pp. 196–201, Aug. 2019, doi: 10.29207/resti.v3i2.945.
[26] M. S. Pangestu and M. A. Fitriani, "Perbandingan Perhitungan Jarak Euclidean Distance, Manhattan Distance, dan Cosine Similarity dalam Pengelompokan Data Bibit Padi Menggunakan Algoritma K-Means," *Sainteks*, vol. 19, no. 2, p. 141, Oct. 2022, doi: 10.30595/sainteks.v19i2.14495.
[27] D. V. Ramadhanti, R. Santoso, and T. Widiharih, "Perbandingan Smote Dan Adasyn Pada Data Imbalance Untuk Klasifikasi Rumah Tangga Miskin Di Kabupaten Temanggung Dengan Algoritma k-Nearest Neighbor," *Jurnal Gaussian*, vol. 11, no. 4, pp. 499–505, Feb. 2023, doi: 10.14710/j.gauss.11.4.499-505.
[28] P. Romadloni, B. Adhi Kusuma, and W. Maulana Baihaqi, "Komparasi Metode Pembelajaran Mesin Untuk Implementasi Pengambilan Keputusan Dalam Menentukan Promosi Jabatan Karyawan," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 2, pp. 622–628, Sep. 2022, doi: 10.36040/jati.v6i2.5238.
[29] D. Normawati, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, pp. 697–711, 2021.
[30] S. Riyanto, I. S. Sitanggang, T. Djatna, and T. D. Atikah, "Comparative Analysis using Various Performance Metrics in Imbalanced Data for Multi-class Text Classification," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, 2023, doi: 10.14569/IJACSA.2023.01406116.

Novanda Rizky Ramadhana: *Corresponding Author