CAUCHY – Jurnal Matematika Murni dan Aplikasi
Volume 11 (1) (2026), Pages 683-695
p-ISSN: 2086-0382; e-ISSN: 2477-3344

Comparison Of Methods For Handling Imbalanced Datasets In Improving Classification Algorithm Performance

Dyah Setyo Rini, Winalia Agwil*, Dian Agustina, and Ahmad Famuji

Statistics Study Program, Bengkulu University, Indonesia

Abstract

Data availability in large observations and dimensions is known as big data. There are several problems in processing big data, such as imbalanced datasets. In classification modeling, an imbalanced dataset is a common challenge. Data class predictions are more likely to be accurate in the majority class data and inaccurate in the minority class, resulting from the problem of imbalanced data. The data-level, the algorithm-level, and the ensemble method approach are the solutions that have been extensively researched. Some methods with a data-level approach are SMOTE, Undersampling, and Oversampling. The algorithm-level method is NWKNN. And then, the ensemble approach is UnderBagging, RUSBoosting, SMOTEBoost, and SMOTEBagging. This study aims to identify the most effective method for handling different levels of class imbalance, namely mild, moderate, and extreme imbalance. A simulation study was conducted for each imbalance level to evaluate classification performance across the compared methods. Based on sensitivity as the primary evaluation metric, the results indicate that SMOTEBagging outperformed other methods in the mild imbalance scenario, UnderBagging showed the best performance under moderate imbalance, and RUSBoosting achieved superior performance in the extreme imbalance condition. These findings demonstrate that ensemble-based resampling approaches provide more robust and reliable performance for imbalanced classification problems across varying imbalance levels.

Keywords: Imbalanced; SMOTE; NWKNN; Ensemble; AUC.

1. Introduction

The rapid development of technology in the 21st century, especially information technology, makes it easy for information technology systems to be digitally connected, so that the process of collecting, storing, and accessing data in large quantities and dimensions (big data) is very easy. Big data has been utilized by some companies in Indonesia and also by government agencies to find insights as needed. Extracting insights can be done using one of them with machine learning methods. According to [1], machine learning is a set of computational methods that automatically extract patterns from data using mathematical models and algorithms to support prediction and decision-making processes. These methods are broadly classified into supervised and unsupervised learning [1].

Unsupervised learning approaches are usually used for the purpose of clustering observations, while supervised learning approaches are used for prediction. Supervised learning approaches

*Corresponding author. E-mail: winaliaagwil@unib.ac.id

Submitted: July 25, 2025 Reviewed: August 07, 2025 Accepted: March 02, 2026
DOI: https://doi.org/10.18860/cauchy.v11i1.35780

include regression models and classification models [2]. The problem of large amounts of data is very diverse, one of which is the condition of data with imbalanced data classes (an imbalanced dataset) [3].

An imbalanced dataset is a condition where one class of data has a significantly smaller proportion of observations (minor class) compared to another class of data (major class). This data condition becomes a problem when the purpose of extracting data is for the classification or prediction of categorical responses [3]. The result of an imbalanced dataset when predicting categorical responses is misclassification in the minor class because the prediction tends towards the major class. The problem of an imbalanced dataset is found in many real cases, such as banking data (credit score), health data (step disease status), poverty data, and others [3].

Research with imbalanced dataset conditions has been carried out by several researchers, but in these studies, no special handling of this condition was carried out, such as [4] conducted poverty classification research in Semarang City with the QUEST algorithm. In this case, there is an imbalanced dataset problem that results in classification errors when predicting minor classes. The same thing is also faced in research [5], in which a household poverty status classification is performed using SVM and CART methods. Both methods show good results for classifying households from the non-poor class (major class), but less good for predicting poor households (minor class).

Several approaches can be taken to deal with imbalanced dataset problems. Le categorizes them into three approaches, namely, algorithm-level, data-level, and ensemble [6]. In addition to these three approaches, ensemble methods can also be used to handle imbalanced dataset problems.

Ensemble approaches such as RUSBoost (Random Under-Sampling and Boosting) and UnderBagging (Under-Sampling and Bagging) have also been reported to outperform many other more complex algorithms in handling imbalanced classification problems [7]. Permatasari has also used the ensemble method in handling imbalanced dataset cases; the results show that the method is sensitive in predicting minority classes (classes with few observations).

This research will examine how the performance of handling imbalanced datasets with the algorithm-level, data-level, and ensemble approaches for each severity of imbalanced dataset conditions. There are 3 groups of imbalanced dataset conditions, namely mild, moderate, and extreme. Mild imbalanced condition, if the percentage of minor classes ranges between 20%-40% of the total data. Moderate imbalanced condition, if the percentage of minor classes ranges between 1-20% of the total data. While an extreme imbalanced condition, if the percentage of minor classes is smaller than 1% of the total data available.

The purpose of this study is to evaluate and compare the performance of several classification methods under different levels of class imbalance. The study examines the effectiveness of data-level techniques, algorithm-level approaches, and ensemble-based methods in handling mild, moderate, and extreme imbalance conditions. By using a controlled experimental framework, the analysis focuses on identifying the most suitable method for each imbalance level and providing a clear comparison of how different imbalance-handling strategies perform across varying imbalance scenarios Methods.

2. Methods

Imbalanced data is a data problem that occurs when there are several observations from a class or category that dominate observations from other categories or classes [9]. The data class that has more observations is called the major class, while the data class that has fewer observations is called the minor class [10]. In classification analysis, imbalanced data can lead to errors in classification. This is because when classifying, it is more likely to be classified in the major class, while the minor class is ignored. There are several methods to overcome the problem of imbalanced data, including the data-level approach, algorithm-level approach, and ensemble approach.

2.1. Research Data

The data used in this study are simulated to represent three different levels of class imbalance. The predictor variables, denoted as X_1, X_2, and X_3, were generated from a multivariate normal distribution using RStudio software. The response variable is categorical, with the majority class coded as 0 and the minority class coded as 1. A total of $10,000$ observations were generated, and the class proportions at each imbalance level are presented in Table 1. For classification modeling, the Classification and Regression Tree (CART) algorithm was employed as the base classifier, except for the NWKNN method, which used KNN as its classifier. The dataset was initially divided into 80% training data and 20% testing data. Furthermore, to obtain more reliable performance estimates and reduce variability, a 5-fold cross-validation procedure was applied to the data during model development.

Table 1: Class Distribution Across Different Imbalance Levels

Imbalance Level	Minority Class	Majority Class
Mild	2500	7500
Moderate	800	9200
Extreme	100	9900

2.2. Data-Level Approach

Handling imbalanced datasets with a data-level approach is developed with the idea of resampling the data, either oversampling, undersampling, or a combination of both sampling methods (hybrid method), with the aim of balancing the proportion of observations in each data class. The level data approach uses several methods such as Oversampling, Undersampling, and SMOTE. The SMOTE method was first introduced by [10], which is a method of overcoming imbalanced data by resampling the data. In general, the SMOTE method can be described as the process of adding new data to the minority class so that the number of observations is equal to the majority class. The generation of new data is based on k-nearest neighbor information [11]. There are two stages in this method: the first stage is determining the nearest neighbor for each observation in the minority class using Euclidean distance if the data is numeric and Value Difference Metric (VDM) if the data is categorical. The second stage is the generation or generation of synthetic data. If the data is numeric, then the new data is calculated through the following equation,

$$x^* = x_i + (\text{random number}(0-1) \times \text{difference}\,(x_i, x_{ij})$$

As for categorical data, the new observation is the majority of the values of its k-nearest neighbors.

2.3. Algorithm-Level Approach

Handling imbalanced data through an algorithm-level approach is done by giving greater weight to minor data classes or data classes that have fewer observations. One method that can be used in handling imbalanced data at the algorithm level is the Nearest Weighted K-Nearest Neighbor (NWKNN). Tan Introduced the Neighbor-Weighted K-Nearest Neighbor (NWKNN) method, which is a development method of the K-NN algorithm used to classify imbalanced data [12]. The NWKNN algorithm uses the principle of weighting (E) on data classes, thus distinguishing it from K-NN. The weighting function is used to balance the data; neighbors from the majority class are given a small weight, and neighbors from the minority class are given a large weight.

2.4. Ensemble Approach

Popular ensemble methods used are bagging and boosting [13]. Bagging is an ensemble method introduced by Breiman in 1996, which is an acronym for bootstrap and aggregating. This method builds several datasets from the original dataset using the bootstrap resampling technique, then

from each dataset, the classification process is carried out. The results of the classification are voted on to obtain the final prediction [14]. Boosting uses the same data but has a different weight distribution at each iteration, depending on the previous classification. The use of weights is also carried out during the process of combining the final prediction of the many trees produced [13]. Handling imbalanced data with ensemble methods has the concept of combining data-level handling and one of the ensemble methods. Methods that can be used such as RUSBoost, Underbagging, SMOTEBoost, and SMOTEBagging. The RUSBoost algorithm adds a resampling technique, namely random undersampling, to the boosting algorithm. Mounce et al. describes the RUSBoost algorithm as follows [13]: Suppose our dataset consists of m observations, with y as the response variable that has k classes. Briefly, the steps of the algorithm can be written as follows:

- Initial determination of the weight of each observation, i.e.

$$D_1(i) = \frac{1}{m}$$

  for all $i = 1, 2, \ldots, m$.
- Suppose t is the iteration number, then for $t = 1, 2, \ldots, T$ do the following process:
  - Create S_t dataset using random undersampling.
  - Build a classification model of the dataset S_t with respect to the weights D_t.
  - Calculate the classification error rate

$$\varepsilon_t = \sum_{(i,y_i):\, y_i \neq y} D_t(i)\,(1 - h_t(x_i, y_i) + h_t(x_i, y)) \tag{1}$$

  - Calculate α as

$$\alpha_t = \frac{\varepsilon_t}{1 - \varepsilon_t} \tag{2}$$

  - Determine the new weight for each observation to be

$$D_{t+1}(i) = D_t(i)\alpha_t^{\frac{1}{2}(1+h_t(x_i,y_i)-h_t(x_i,y):y\neq y_i)} \tag{3}$$

  For misclassified observations, while for correctly estimated observations, the weight is fixed.
- The final estimation is

$$H(x) = \arg\max_{y \in Y} \sum_{t=1}^{T} h_t(x, y) \log \frac{1}{\alpha_t} \tag{4}$$

  The UnderBagging method is a combination of undersampling and bagging techniques, first introduced by [15]. In the UnderBagging algorithm, each subset Sk is created by undersampling the majority class data to construct the k-th classifier. The final classification decision follows the most voted class [16]. SMOTE-Boost combines the SMOTE algorithm and standard boosting procedures, utilizing SMOTE to improve minority class predictions and utilizing boosting so as not to sacrifice accuracy over the entire data set. SMOTE-Boosting originally used the iteration procedure of AdaBoost.M2 boosting [17]. The SMOTE-Boosting algorithm with the AdaBoost.M2 iteration procedure is as follows:
  Input: $(x_i, y_i), \ldots, (x_m, y_m)$ where $x_i \in X$ and $y_i \in Y = \{1, \ldots, k\}$
  Initialization:

$$D_1(i, y) = \frac{1}{|B|}$$

  for all i.
  For $t = 1, \ldots, T$

- Create a new training dataset S'_t using SMOTE.
- Train the weak learner using the D_t distribution.
- Get the weak hypothesis and calculate pseudo loss in the equation:

$$\varepsilon_t = \sum_{(i,y)\in B} D_t(i,y)\left(1 - h_t(x_i,y_i) + h_t(x_i,y)\right) \tag{5}$$

  If the value of $\varepsilon_t > 0.5$ is obtained, then the learning process stops.
- Calculate

$$\beta_t = \frac{\varepsilon_t}{(1-\varepsilon_t)}$$

- Update the weight value:

$$D_{t+1}(i,y) = \frac{D_t(i,y)}{Z_t}\beta_t^{\frac{1}{2}(1+h_t(x_i,y_i)-h_t(x_i,y))} \tag{6}$$

  where Z_t is the normalization constant.
- The output of the final hypothesis, that is:

$$H(x) = \arg\max_{y\in Y} \sum_{t=1}^{T} \left(\log \frac{1}{\beta_t}\right) h_t(x,y) \tag{7}$$

SMOTE-Bagging is a combination of the SMOTE and Bagging algorithms. SMOTE-Bagging involves a synthetic data generation step during subset construction [18]. SMOTE works by regenerating the synthetic data until the number of minor data equals the number of major data. Each subset in SMOTE-Bagging is obtained through a bootstrap process that is balanced by SMOTE before modeling. In summary, the stages of the SMOTE-Bagging algorithm can be written as follows:

- Training data initiation, D.
- (Bagging step): For $t = 1, 2, \ldots, T$ do the following process:
  - Create a dataset D_t using minor class resample (N) with replacement.
  - Generate a new dataset with SMOTE.
  - Train a classifier from SMOTE dataset (H_t).
- The final classification decision follows the most voted class.

2.5. Model goodness evaluation

Classification performance is usually measured by the Confusion Matrix Table 2. A confusion matrix is a tabulation of classification accuracy on predicted and actual data. True Positive is the number of observations correctly classified from the positive class, True Negative is the number of observations correctly classified from the negative class, False Negative is the number of observations misclassified from the positive class, while False Positive is the number of observations misclassified from the negative class.

Table 2: Confusion Matrix

Prediksi	**Aktual**	
	Positif	Negatif
Positif	*True Positive* (TP)	*False Positive* (FP)
Negatif	*False Negative* (FN)	*True Negative* (TN)

Based on the values in the Confusion Matrix, the Accuracy, Sensitivity, and Specificity values can be calculated. Accuracy value is the level of accuracy of the classifier in classifying the observations. The following formula is used to see the classification performance:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP}$$
$$\text{Sensitivity (Recall)} = \frac{TP}{TP + FN}$$
$$\text{Specificity} = \frac{TN}{TN + FP}$$
$$\text{Precision} = \frac{TP}{TP + FP}$$
$$F1\text{-Score} = 2 \cdot \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

The level of classification accuracy can also be seen in the AUC value, which is the area under the ROC curve that ranges from 0 to 1. The area under the ROC curve (AUC) is a standard metric for quantifying and comparing binary classifiers [19]. The curve is a plot of the percentage of *False Positive* ($1 - Specificity$) with the percentage of *True Positive*.

3. Results and Discussion

This section reports the results obtained from the simulation study and provides a discussion of the performance of each imbalance-handling method. To facilitate interpretation, the presentation starts with a general description of the generated datasets, after which the classification results for mild, moderate, and extreme imbalance conditions are discussed in sequence.

3.1. Dataset Overview

This study uses data generation with several imbalanced data conditions. Data is generated with as many as 10000 observations with the provisions of three predictor variables and categorical response variables consisting of 2 categories. Datasets with mild imbalance categories are set with response variables consisting of a minority class of 2500 observations and a majority class of 7500 observations.

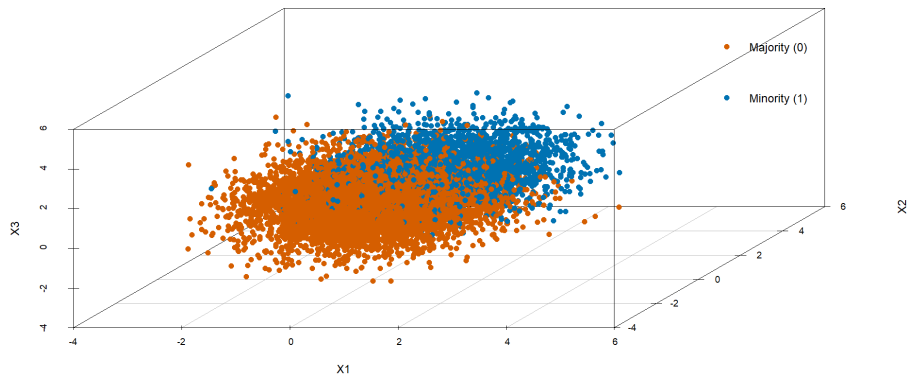

Fig. 1: Overview of the Mild Imbalance Category Dataset

Based on Fig. 1. it can be seen that the dataset condition has a mild imbalance in which the minority class of minority observations is indicated by black points while the majority class of observations is indicated by observations with red points. Next is the condition of the dataset imbalance at a moderate level, with a ratio of the number of minority and majority observation classes of 800:9200.

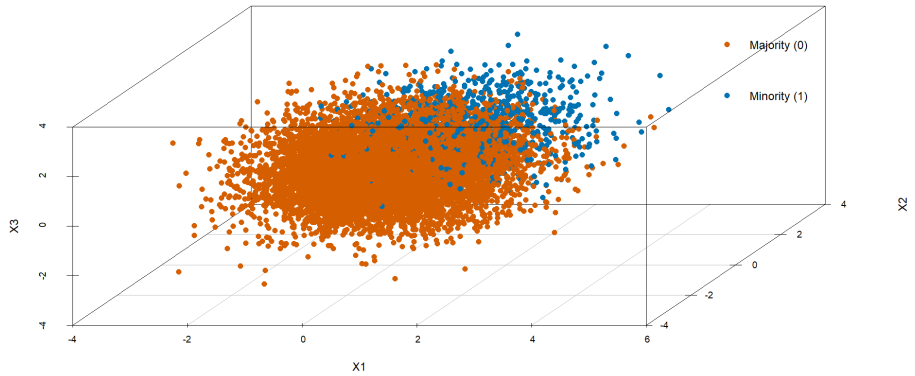

Fig. 2: Moderate Imbalance Category Dataset Overview

Fig. 2 is an illustration of the dataset condition when the observed classes are at a moderate imbalanced level, but it can be seen that the observations with black points coming from the minority class are almost covered by observations with red points. Next is an illustration of a dataset with an extreme imbalanced level, which is a degenerate dataset with respect to the ratio of the number of observed classes of 100:9900. So, it can be seen in Figure 3 that the observations coming from the minority class (black points) are almost invisible or undetectable.

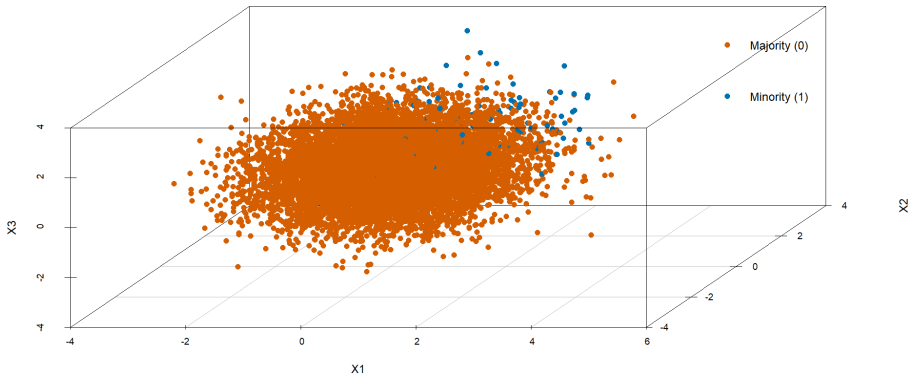

Fig. 3: Moderate Imbalance Category Dataset Overview

The three previously described dataset conditions may pose significant challenges in classification tasks and in predicting the classes of new observations. This difficulty arises because the model-building process is often dominated by the majority class, leading to biased predictions that favor the prevalent class. Consequently, the model tends to misclassify minority class instances, reducing overall predictive performance. To address this issue, the present study employs a combination of strategies, including data-level approaches, algorithm-level modifications, and ensemble methods. The overarching objective is to identify effective handling methods for each degree of class imbalance, thereby improving classification accuracy and robustness across varying imbalance scenarios.

3.2. Relationship between Input Variables

Table 3 presents the correlation matrix among input (predictor) variables in the dataset with a low imbalance level (Mild Imbalance). The results indicate no significant correlation between the predictors, as shown by correlation values close to zero. Similar patterns are observed in the correlation matrices for the moderate imbalance dataset (Table 4) and the extreme imbalanced dataset (Table 5).

Table 3: Correlation Matrix of Datasets with Mild Imbalance

	X_1	X_2	X_3
X_1	1	0.1434	0.1662
X_2	0.1434	1	0.1309
X_3	0.1662	0.1309	1

Table 4: Correlation Matrix of Datasets with Moderate Imbalance

	X_1	X_2	X_3
X_1	1	0.0675	0.0745
X_2	0.0675	1	0.0728
X_3	0.0745	0.0728	1

Table 5: Correlation Matrix of Datasets with Extreme Imbalance

	X_1	X_2	X_3
X_1	1	0.0097	0.0301
X_2	0.0097	1	-0.0220
X_3	0.0301	-0.0220	1

3.3. Handling Class Imbalance at the Mild Imbalance Level

Overcoming imbalance conditions to improve classification performance, namely with a data-level approach, an algorithm-level, and an ensemble. The following is an overview of classification performance when imbalance conditions have not been addressed (original data) and after being addressed by several methods at mild imbalance levels.

Table 6: Classification performance (Mild Imbalance Level)

Methods	Accuracy	Specificity	Sensitivity	F1-Score	AUC value
CART (only)	0.8310	0.9285	0.5384	0.6143	0.7918
SMOTE	0.8333	0.9316	0.5384	0.6175	0.7899
Undersampling	0.7741	0.7768	0.7660	0.6290	0.8156
Oversampling	0.7785	0.7890	0.7468	0.6276	0.8078
MNKNN	0.8334	0.9162	0.5848	0.6370	0.8369
Underbagging	0.7828	0.7909	0.7584	0.6358	0.8512
RUS Boosting	0.8024	0.8069	0.7888	0.6662	0.8820
SMOTEBoost	0.8432	0.8869	0.7120	0.6942	0.8934
SMOTEBagging	0.7600	0.7397	**0.8208**	0.6309	0.8590

The results in Table 6 present the classification performance under the mild imbalance scenario. The baseline CART model achieved an accuracy of 0.8310 with high specificity (0.9285) but relatively low sensitivity (0.5384), indicating that the classifier predominantly favored the majority class while struggling to identify minority instances correctly. A similar pattern was observed for the SMOTE-enhanced CART, where the overall accuracy increased slightly to 0.8333. Still, sensitivity remained unchanged, suggesting that simple synthetic oversampling alone did not substantially improve minority class detection.

Sampling-based strategies demonstrated a different behavior. Undersampling increased sensitivity to 0.7660, showing a marked improvement in detecting minority cases, although this gain was accompanied by a reduction in specificity and overall accuracy. Oversampling exhibited a comparable trade-off, yielding balanced sensitivity (0.7468) and specificity (0.7890), which implies a more equitable treatment of both classes but with moderate classification precision.

The NWKNN method produced competitive performance with balanced metrics, achieving an AUC of 0.8369 and an F1-score of 0.6370, indicating reasonable discrimination capability and balanced precision–recall behavior. However, its sensitivity remained moderate, suggesting limited effectiveness in identifying minority observations compared to ensemble-based approaches.

Ensemble methods showed superior overall discrimination. Underbagging achieved high sensitivity (0.7584) with an AUC of 0.8512, reflecting strong capability in capturing minority patterns while maintaining acceptable specificity. RUS Boosting further improved the balance between sensitivity (0.7888) and specificity (0.8069), resulting in a higher F1-score (0.6662) and AUC (0.8820), which demonstrates robust classification across both classes.

SMOTEBoost delivered the highest AUC value (0.8934) and the best F1-score (0.6942), indicating strong ranking ability and improved balance between precision and recall. Although its sensitivity (0.7120) was slightly lower than that of RUS Boosting and SMOTEBagging, it still showed a substantial improvement over single-model approaches. SMOTEBagging, on the other hand, produced the highest sensitivity (0.8208), highlighting its effectiveness in detecting minority instances, though at the cost of reduced specificity (0.7397).

Overall, the findings suggest that ensemble-based imbalance handling methods outperform single classifiers and basic resampling techniques. In particular, RUS Boosting and SMOTEBoost provide the most favorable trade-off between discrimination ability and minority class detection, while SMOTEBagging excels in maximizing sensitivity. These results confirm that combining resampling strategies with ensemble learning is more effective for mildly imbalanced classification problems than relying on standalone models.

3.4. Handling Class Imbalance at the Moderate Imbalance Level

Datasets with moderate imbalance levels are generated using R software with 800 minority class observations and 9200 majority class observations. The "moderate" imbalance condition is addressed by several methods, with the results of the evaluation metrics in Table 7.

Table 7: Classification performance (Moderate Imbalance Level)

Methods	Accuracy	Specificity	Sensitivity	F1-Score	AUC value
CART (only)	0.9297	0.9856	0.2862	0.3944	0.7404
SMOTE	0.9280	0.9854	0.2675	0.3728	0.7404
Undersampling	0.7670	0.7651	0.7887	0.3513	0.8223
Oversampling	0.7674	0.7641	0.8050	0.3563	0.8172
NWKNN	0.9294	0.9802	0.3450	0.4387	0.7898
Underbagging	0.7500	0.7408	**0.8550**	0.3536	0.8768
RUS Boosting	0.7864	0.7873	0.7750	0.3672	0.8701
SMOTEBoost	0.9260	0.9747	0.3650	0.4410	0.8915
SMOTEBagging	0.7788	0.7769	0.8000	0.3665	0.8567

Table 7 presents the classification performance under the moderate imbalance setting. The baseline CART model achieved high accuracy (0.9297) and very high specificity (0.9856), but extremely low sensitivity (0.2862), indicating that the classifier strongly favored the majority class and struggled to detect minority instances. A similar pattern was observed for the SMOTE-enhanced model, where accuracy and specificity remained high while sensitivity showed only a slight improvement, suggesting SMOTE was insufficient to substantially enhance minority class recognition.

Sampling-based approaches produced a different behavior. Undersampling and oversampling significantly increased sensitivity to around 0.79–0.81, demonstrating improved ability to identify minority observations. However, these gains were accompanied by reduced specificity and overall accuracy, reflecting the typical trade-off in imbalanced classification where better recall often comes at the cost of higher false positive rates.

The NWKNN method showed balanced performance with moderate sensitivity (0.3450) and competitive AUC (0.7898), indicating reasonable discrimination but limited effectiveness in capturing minority cases compared to ensemble methods. In contrast, ensemble-based techniques consistently delivered superior results. Underbagging achieved the highest sensitivity (0.8550) along with a strong AUC (0.8768), highlighting its effectiveness in detecting minority patterns. RUS Boosting and SMOTEBagging also demonstrated robust performance, offering high sensitivity and balanced specificity, which resulted in improved overall discrimination.

SMOTEBoost obtained the highest AUC value (0.8915) and the best F1-score (0.4410), indicating strong ranking capability and balanced precision–recall performance. Although its sensitivity remained moderate, the high AUC suggests that the model effectively separated minority and majority instances probabilistically but remained conservative in classification decisions. Overall, the results indicate that ensemble methods combined with resampling strategies provide the most effective approach for moderate imbalance, achieving a better balance between minority detection and overall discrimination compared to single classifiers and basic sampling techniques.

3.5. Handling Class Imbalance at the Extreme Imbalance Level

The dataset representing an extreme imbalance level was generated by setting the number of minority class observations to 100 and the majority class observations to 9900. Several imbalance-handling methods were then applied to this dataset, with the results summarized in Table 8.

Table 8: Classification performance (Extreme Imbalance Level)

Methods	Accuracy	Specificity	Sensitivity	F1-Score	AUC value
CART (only)	0.9896	0.9993	0.0200	0.0370	0.5818
SMOTE	0.9895	0.9987	0.0700	0.1176	0.5556
Undersampling	0.7225	0.7218	0.7900	0.0538	0.8206
Oversampling	0.8287	0.8304	0.6600	0.0715	0.7593
NWKNN	0.9897	0.9990	0.0600	0.1043	0.6842
Underbagging	0.8148	0.8161	0.6800	0.0684	0.8444
RUS Boosting	0.8220	0.8222	**0.8000**	0.0824	0.8923
SMOTEBoost	0.9884	0.9967	0.1600	0.2162	0.8690
SMOTEBagging	0.8412	0.8428	0.6800	0.0788	0.8384

Table 8 reports the classification performance under the extreme imbalance scenario. The baseline CART model achieved very high accuracy (0.9896) and specificity (0.9993), but extremely low sensitivity (0.0200), indicating that the classifier almost always predicted the majority class and failed to detect minority instances. A similar pattern was observed for the SMOTE-enhanced model, where sensitivity slightly increased but remained very low, showing that simple synthetic oversampling was insufficient to address severe class imbalance.

Sampling-based methods such as undersampling and oversampling substantially improved sensitivity to around 0.66˘0.79, demonstrating better capability in identifying minority observations. However, this improvement came at the expense of lower specificity and accuracy, reflecting the common trade-off in highly imbalanced data where increased recall often leads to more false positives. The NWKNN method still showed very low sensitivity (0.0600) despite high specificity, suggesting limited effectiveness in capturing rare minority patterns under extreme imbalance.

Ensemble-based approaches provided more robust performance. Underbagging and SMOTEBagging achieved high sensitivity (0.6800) with acceptable specificity, indicating a more balanced classification between majority and minority classes. RUS Boosting produced the highest sensitivity (0.8000) and the best AUC value (0.8923), demonstrating strong ability to detect minority instances while maintaining good overall discrimination. Although SMOTEBoost achieved high accuracy and specificity, its sensitivity remained relatively low (0.1600), suggesting that the

model was still conservative in classifying minority cases despite having good ranking ability as reflected by a high AUC (0.8690).

Overall, the results show that single classifiers and basic resampling techniques are inadequate for extreme imbalance, as they tend to bias predictions toward the majority class. In contrast, ensemble methods that integrate resampling strategies, particularly RUS Boosting and bagging-based approaches, provide more effective and stable performance by significantly improving minority class detection while preserving reasonable overall discrimination.

3.6. Evaluation Metric Confidence Interval

Based on the results presented in Table 6, Table 7, Table 8, the best-performing method at each imbalance level was selected solely based on sensitivity, as this metric directly reflects the model's ability to correctly identify minority class instances, which is the primary objective in imbalanced classification problems.

For the mild imbalance condition (Table 9), SMOTEBagging shows relatively narrow confidence intervals for accuracy, sensitivity, and specificity, indicating consistent predictive performance and reliable detection of minority instances. Under the moderate imbalance scenario (Table 10), Underbagging maintains stable performance, with a sensitivity interval that remains consistently high, suggesting robustness in identifying minority observations despite increased imbalance. For the extreme imbalance level (Table 11), RUSBoost exhibits wider sensitivity intervals compared to the other levels, reflecting greater variability under severe class imbalance. Nevertheless, the confidence intervals still confirm that the method maintains acceptable and stable classification performance while effectively detecting minority instances.

Table 9: Classification performance of Testing Data (Mild Imbalance)

Methods	Accuracy Confidence Interval		Sensitivity Confidence Interval		Specificity Confidence Interval	
	Lower Limit	Upper Limit	Lower Limit	Upper Limit	Lower Limit	Upper Limit
SMOTE-Bagging	0.7448	0.7780	0.7917	0.8539	0.7216	0.7618

Table 10: Classification performance of Testing Data (Moderate Imbalance)

Methods	Accuracy Confidence Interval		Sensitivity Confidence Interval		Specificity Confidence Interval	
	Lower Limit	Upper Limit	Lower Limit	Upper Limit	Lower Limit	Upper Limit
Under-bagging	0.7363	0.7664	0.8067	0.9011	0.7262	0.7577

Table 11: Classification performance of Testing Data (Extreme Imbalance)

Methods	Accuracy Confidence Interval		Sensitivity Confidence Interval		Specificity Confidence Interval	
	Lower Limit	Upper Limit	Lower Limit	Upper Limit	Lower Limit	Upper Limit
Rus-Boost	0.8048	0.8361	0.6362	0.9230	0.8053	0.8375

4. Conclusion

This study evaluated several resampling and ensemble-based methods for handling class imbalance under mild, moderate, and extreme imbalance conditions. The results consistently showed that single classifiers and basic resampling techniques tend to bias predictions toward the majority

class, leading to high specificity but poor sensitivity, particularly as the imbalance level increases. Ensemble-based approaches demonstrated more robust and balanced performance across all imbalance scenarios. Based on sensitivity as the primary evaluation criterion, SMOTEBagging outperformed other methods under the mild imbalance condition, UnderBagging showed the best performance for the moderate imbalance level, and RUSBoost achieved the highest sensitivity under the extreme imbalance condition, indicating their superior capability in detecting minority class instances across different imbalance settings.

CRediT Authorship Contribution Statement

Dyah Setyo Rini: Conceptualization and Methodology. **Winalia Agwil:** Methodology, Formal Analysis, and Writing Review. **Dian Agustina:** Software, Validation. **Ahmad Famuji:** Visualization, Data Generation.

Declaration of Generative AI and AI-assisted technologies

AI-assisted technologies were used for refining language and structuring initial drafts, while all scientific content and interpretations were independently developed and verified by the authors.

Declaration of Competing Interest

The authors declare no competing interests.

Funding and Acknowledgments

We would like to thank all those who have helped provide valuable suggestions for improving the quality of this paper. Our gratitude to the Faculty of Mathematics and Natural Sciences for research funding "Unggulan FMIPA Tahun 2023".

Data and Code Availability

The dataset used in this study was synthetically generated through simulation. Data generation was conducted by defining specific parameters designed to reflect the characteristics of real-world data, ensuring that the resulting dataset adequately represents the conditions under investigation. Utilizing simulated data allows for controlled evaluation and testing of the methods, as well as enabling sensitivity analyses across various desired scenarios.

References

[1] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, p. 160, 2021. DOI: 10.1007/s42979-021-00592-x.

[2] C. E. Varma and P. S. Prasad, "Supervised and unsupervised machine learning approaches—a survey," in *Proceedings of the 3rd International Conference on Data Science, Machine Learning and Applications (ICDSMLA 2021)*, Singapore: Springer Nature Singapore, 2023, pp. 73–81. DOI: 10.1007/978-981-19-5936-3_7.

[3] A. Ugarković and D. Oreški, "Supervised and unsupervised machine learning approaches on class imbalanced data," in *2022 International Conference on Smart Systems and Technologies (SST)*, IEEE, 2022, pp. 159–162. DOI: 10.1109/SST55530.2022.9954646.

[4] D. Ispriyanti, A. Prahutama, and M. Mustafid, "Analisis klasifikasi kemiskinan di kota semarang menggunakan algoritma quest," *Statistika*, vol. 7, no. 1, pp. 47–54, 2019.

[5] L. Nuzula, A. Prahutama, and A. R. Hakim, "Klasifikasi status kemiskinan rumah tangga dengan metode support vector machines (svm) dan classification and regression trees (cart) menggunakan gui r (studi kasus di kabupaten wonosobo tahun 2018)," *J. Gaussian*, vol. 9, no. 4, pp. 525–534, 2020. DOI: 10.14710/j.gauss.v9i4.29449.

[6] T. Le, "A comprehensive survey of imbalanced learning methods for bankruptcy prediction," *IET Communications*, vol. 16, no. 5, pp. 433–441, 2021. DOI: 10.1049/cmu2.12268.

[7] A. O. Adegbenjo and M. O. Ngadi, "Handling the imbalanced problem in agri-food data analysis," *Foods*, vol. 13, no. 20, p. 3300, 2024. DOI: 10.3390/foods13203300.

[8] Y. Permatasari, "Penanganan masalah kelas tidak seimbang dengan rusboost dan underbagging (studi kasus: Mahasiswa drop out sps ipb program magister)," http://repository.ipb.ac.id/handle/123456789/80118, M.S. thesis, IPB University, 2016.

[9] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017. DOI: 10.1016/j.eswa.2016.12.035.

[10] 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. DOI: 10.1613/jair.953.

[11] F. R. Torres, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "Smote-d: A deterministic version of smote," in *Mexican Conference on Pattern Recognition*, Cham: Springer International Publishing, 2016, pp. 177–188. DOI: 10.1007/978-3-319-39393-3_18.

[12] S. Tan, "For unbalanced text corpus," *Expert Systems with Applications*, vol. 28, no. 4, pp. 667–671, 2005. DOI: 10.1016/j.eswa.2004.12.023.

[13] S. R. Mounce, K. Ellis, J. M. Edwards, V. L. Speight, N. Jakomis, and J. B. Boxall, "Ensemble decision tree models using rusboost for estimating risk of iron failure in drinking water distribution systems," *Water Resources Management*, vol. 31, no. 5, pp. 1575–1589, 2017. DOI: 10.1007/s11269-017-1595-8.

[14] G. Tüysüzoğlu and D. Birant, "Enhanced bagging (ebagging): A novel approach for ensemble learning," *International Arab Journal of Information Technology*, vol. 17, no. 4, pp. 515–528, 2020. DOI: 10.34028/iajit/17/4/10.

[15] R. Barandela, R. M. Valdovinos, and J. S. Sánchez, "New applications of ensembles of classifiers," *Pattern Anal. Appl.*, vol. 6, pp. 245–256, 2003. DOI: 10.1007/s10044-003-0192-z.

[16] B. Sun, H. Chen, J. Wang, and H. Xie, "Evolutionary under-sampling based bagging ensemble method for imbalanced data classification," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 331–350, 2018. DOI: 10.1007/s11704-016-5306-z.

[17] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *Proceedings of the Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Cavtat-Dubrovnik, Croatia, 2003, pp. 107–119. DOI: 10.1007/978-3-540-39804-2_12.

[18] S. Wang and X. Yao, "Diversity analysis on imbalances data sets by using ensemble models," in *IEEE Symposium on Computational Intelligence and Data Mining*, 2009, pp. 324–331. DOI: 10.1109/CIDM.2009.4938667.

[19] J. Xu, H. Wang, and Z. Li, "Comparing multi-class classifier performance by multi-class roc analysis: A nonparametric approach," *Neurocomputing*, vol. 583, p. 127 520, 2024. DOI: 10.1016/j.neucom.2024.127520.