CAUCHY – Jurnal Matematika Murni dan Aplikasi
Volume 11 (2) (2026), Pages 2584–2599
p-ISSN: 2086-0382; e-ISSN: 2477-3344

The Comparative Analysis of Hybrid Sampling, EasyEnsemble, and SVM-SMOTE Performance for Imbalanced Data Based on Bayesian Optimization

Megawati Zein Waliulu*, Kusman Sadik, and Asep Saefuddin

Department of Statistics and Data Science, School of Data Science, Mathematics, and Informatics, IPB University, Indonesia

Abstract

Imbalanced data represents a common challenge in classification modeling, as it can lead to prediction bias and reduce the model's ability to correctly identify minority classes, which are often more critical. This study aims to compare the performance of three imbalance-handling techniques—Hybrid Sampling, EasyEnsemble, and SVM-SMOTE—applied to the XGBoost classifier with hyperparameter tuning using Bayesian Optimization. A simulation study was conducted under a factorial completely randomized design involving three factors: class proportion (70:30, 80:20, 90:10), predictor variable types (10 categorical variables and a combination of 5 categorical and 5 numeric variables), and imbalance handling methods. Model performance was evaluated using balanced accuracy, sensitivity, and specificity. During the optimization process, 20 hyperparameter combinations were initially selected as prior evaluation points, followed by 20 additional optimization iterations to refine the search for the optimal hyperparameter configuration. The Bayesian Optimization process produced different optimal hyperparameter combinations for each simulation scenario, reflecting the distinct characteristics of the data under different experimental conditions. The analysis of variance revealed that both the main effects and the interactions among all factors significantly influenced classification performance. Among the evaluated methods, EasyEnsemble consistently achieved the best performance across all class imbalance scenarios. For all class proportions (70:30, 80:20, and 90:10), both in datasets consisting of 10 categorical predictor variables and in datasets comprising a combination of 5 categorical and 5 numerical predictor variables, EasyEnsemble achieved balanced accuracy, sensitivity, and specificity values consistently exceeding 0.90. Furthermore, Bayesian Optimization effectively enhanced the classification performance of XGBoost by efficiently identifying optimal hyperparameter configurations. Overall, the findings demonstrate that the combination of EasyEnsemble and Bayesian Optimization provides an effective approach for addressing class imbalance and improving classification performance.

Keywords: Bayesian Optimization; EasyEnsemble; Hybrid Sampling; Imbalanced Data; SVM-SMOTE

Copyright © 2026 by Authors, Published by CAUCHY Group. This is an open access article under the CC BY-SA License (https://creativecommons.org/licenses/by-sa/4.0)

*Corresponding author. E-mail: waliulumegawati@apps.ipb.ac.id

Received: May 26, 2026 Revised: October 2, 2026 Accepted: October 31, 2026
DOI: https://doi.org/10.18860/cauchy.v11i2.42968

1. Introduction

Imbalanced data refers to a condition in which the distribution of samples across classes is disproportionate, where one or more classes contain substantially fewer samples than the others [1]. Such class distribution imbalance within a dataset can introduce bias in the classification process, leading the model to favor the majority class. This bias may increase classification errors and result in the minority class being largely overlooked [2]. When a predictive model is confronted with imbalanced data, commonly used evaluation metrics often fail to provide an accurate representation of its performance. A model may achieve high accuracy simply by assigning most samples to the majority class, even though it fails to correctly identify the minority class, which may be of greater importance. Various approaches have been developed to address imbalanced data to enable models to produce more accurate and balanced predictions.

One approach commonly used to address imbalanced data is Hybrid Sampling, which combines undersampling and oversampling techniques to balance class distribution within a dataset [3]. Oversampling aims to increase the number of minority class samples by generating synthetic data or duplicating existing samples [4], whereas undersampling reduces the number of samples from the majority class to achieve a more balanced distribution relative to the minority class [5]. Hybrid sampling mitigates the limitations of each individual technique by preserving essential information from the majority class while ensuring adequate representation of the minority class. This method has been shown to be effective in improving the performance of classification models by reducing the risk of overfitting commonly associated with oversampling and minimizing the loss of important information that can occur with undersampling.

Another widely used approach for handling imbalanced data is EasyEnsemble, an ensemble method based on bootstrap aggregating (bagging). This technique operates by partitioning the majority class into several smaller subsets and subsequently training multiple learning models using each subset combined with the complete minority class [6]. Through this process, every model within the ensemble retains information from the minority class without excessively discarding majority class samples. The main advantage of this method lies in its ability to enhance model generalization without requiring substantial modification to the original dataset distribution [7].

The SVM-SMOTE method is another technique used to address imbalanced data by integrating Support Vector Machine (SVM) with the Synthetic Minority Oversampling Technique (SMOTE). SMOTE generates synthetic minority class samples by interpolating existing data points, thereby producing a more balanced class distribution. Meanwhile, SVM serves as a classification model capable of effectively separating classes using an optimal margin [8]. This combination enables the model to achieve improved sensitivity toward the minority class without inducing overfitting. Furthermore, the incorporation of SVM in this approach facilitates the construction of a more accurate hyperplane for class separation, ultimately enhancing overall classification performance in imbalanced data scenarios [9].

In modern research, optimization approaches such as Bayesian Optimization have also been employed to enhance model performance in addressing class imbalance. This technique is considered one of the most effective methods for identifying the optimal combination of hyperparameters in machine learning [10]. In its process, Bayesian Optimization utilizes an acquisition function to determine the next set of hyperparameters to evaluate, based on the outcomes of previous evaluations [11]. A key advantage of Bayesian Optimization over other hyperparameter search methods, such as Grid Search or Random Search, is its ability to adaptively explore the parameter space more efficiently, enabling the discovery of optimal configurations with fewer evaluations [12].

On the other hand, several learning algorithms have been developed to directly address class imbalance problems. For example, Extreme Gradient Boosting (XGBoost) is a gradient-boosting-based algorithm capable of adjusting error weights to improve sensitivity to minority classes [13]. The main advantages of XGBoost lie in its ability to handle complex feature interactions,

the incorporation of regularization techniques to reduce overfitting, and support for parallel processing that accelerates model training. Furthermore, XGBoost can optimize error weighting during the training process, enabling it to be more adaptive in learning patterns from imbalanced datasets.

Several studies have examined comparisons of machine learning methods for handling class imbalance to improve prediction and classification performance. [7] employed SVM-SMOTE together with EasyEnsemble and demonstrated improvements in model accuracy and stability. Likewise, [14] evaluated several ensemble-based methods, including EasyEnsemble, Balance-Cascade, and Self-Paced Ensemble, for predicting bicycle-related traffic accident severity and reported that EasyEnsemble combined with XGBoost achieved the best predictive performance. In addition, [15] compared Bayesian Optimization, Grid Search, and Random Search, showing that Bayesian Optimization outperformed the other hyperparameter optimization methods. Although these studies have demonstrated the effectiveness of imbalance handling methods and hyperparameter optimization, their evaluations were predominantly conducted using empirical datasets whose characteristics are fixed and cannot be systematically controlled. Consequently, it remains difficult to isolate the influence of class imbalance levels and predictor characteristics on model performance, making objective comparisons among imbalance handling methods under identical experimental conditions challenging.

Simulation studies provide an effective alternative for overcoming these limitations because they enable researchers to generate datasets with predefined characteristics while maintaining control over the data-generating process. In contrast to empirical datasets, simulation allows factors such as class imbalance ratios, predictor types, coefficient structures, and other distributional characteristics to be systematically manipulated while holding other conditions constant. This provides an experimental framework for examining not only which imbalance handling method performs better, but also whether its effectiveness changes across different levels of class imbalance and predictor settings. Repeated simulation under identical experimental conditions further improves the reproducibility and robustness of the comparison.

Based on this research gap, the present study employs a logistic regression-based simulation framework to generate imbalanced datasets under several experimental scenarios. The simulated datasets are constructed with three class imbalance ratios, and two predictor settings, namely datasets containing only categorical variables and datasets containing a combination of categorical and numerical variables. Subsequently, Hybrid Sampling, EasyEnsemble, and SVM-SMOTE are implemented to address the class imbalance problem, while XGBoost is used as the classification algorithm. To ensure a fair comparison among the competing methods, the hyperparameters of XGBoost are optimized using Bayesian Optimization before model evaluation. The performance of each method is then assessed using Balanced Accuracy, Sensitivity, and Specificity across all simulation scenarios.

The methodological novelty of this study lies in the development of a controlled and reproducible simulation framework that enables the effects of class imbalance severity and predictor composition on the effectiveness of imbalance handling methods to be systematically isolated and evaluated. Rather than relying on a single empirical dataset, the factorial simulation design allows Hybrid Sampling, EasyEnsemble, and SVM-SMOTE to be compared under identical data-generation conditions across multiple imbalance ratios and predictor settings. Furthermore, the consistent application of Bayesian Optimization to XGBoost provides a standardized hyperparameter optimization procedure across competing imbalance handling methods. This framework therefore enables evaluation of both the relative performance and robustness of imbalance handling methods across controlled experimental conditions, providing evidence that is less dependent on the specific characteristics of any single empirical dataset.

2. Methods

This section outlines the methodological framework used in this study, which employs a simulation-based experimental design to compare the performance of Hybrid Sampling, EasyEnsemble, and SVM-SMOTE in handling imbalanced data using XGBoost optimized through Bayesian Optimization. The experimental design combines different class proportions (70:30, 80:20, and 90:10), predictor types (10 categorical predictors or a combination of 5 categorical and 5 numerical predictors), and data imbalance handling methods. The methodology includes synthetic data generation, application of imbalance handling techniques, hyperparameter optimization, and model performance evaluation using Balanced Accuracy, Sensitivity, and Specificity. The results are analyzed using three-way ANOVA and Tukey's post hoc test to evaluate the effects of the experimental factors and their interactions.

2.1. Data

The simulation data generation in this study was conducted in accordance with the predefined experimental design scenarios. Data were generated by considering three primary factors, namely the type of predictor variables (10 categorical variables, and a combination of 5 categorical and 5 numeric variables), class proportions (70:30, 80:20, 90:10), and the methods used to address class imbalance (Hybrid Sampling, EasyEnsemble, and SVM-SMOTE), including a scenario without imbalance handling, resulting in a total of 24 combinations. The 24 combinations are presented in Table 1.

Table 1: Simulation Data Design

Factor	**Level**
Class Proportion (A)	70:30
	80:20
	90:10
Variable Types (B)	10 categorical variables
	5 categorical and 5 numeric variables
Imbalance Handling Methods (C)	Without handling
	Hybrid Sampling
	EasyEnsemble
	SVM-SMOTE

2.2. Experimental Procedure

The following are the stages of research related to simulation data:

1. Data Generation
   (a) The predictor variables X were generated according to predetermined proportions. Categorical predictor variables were generated from a multinomial distribution, while numerical predictor variables were generated from a normal distribution. The parameters of both distributions were specified subjectively because this study is a simulation study that does not rely on empirical data. The selected parameter values were intended to generate simulated datasets with diverse characteristics for both categorical and numerical variables, thereby representing a range of conditions that may occur in practice. Such variability enables a more comprehensive evaluation of the performance of class imbalance handling methods under different simulation scenarios. Eq. (1) defines the multinomial distribution used to generate categorical variables with three levels (A, B, and C).

$$\mathbf{P}(\mathbf{X} = k) = \frac{\mathbf{n}!}{\mathbf{x}_1!\,\mathbf{x}_2! \cdots \mathbf{x}_k!}\,\mathbf{p}_1^{\mathbf{x}_1}\,\mathbf{p}_2^{\mathbf{x}_2} \cdots \mathbf{p}_k^{\mathbf{x}_k} \tag{1}$$

where $\mathbf{P}_k$ is the probability of a category, $\mathbf{x}_k$ is the number of observations in a category, and $\mathbf{n}$ is the total number of observations.

Table 2: Category Probabilities of the 10 Categorical Predictors Used in the Simulation

Predictor	*Level*		
	A	**B**	**C**
X1	0.2727	0.2727	0.4545
X2	0.1765	0.3529	0.4706
X3	0.2500	0.4167	0.3333
X4	0.3333	0.2500	0.4167
X5	0.0909	0.6364	0.2727
X6	0.2000	0.5000	0.3000
X7	0.1429	0.3333	0.5238
X8	0.1538	0.0769	0.7692
X9	0.1739	0.2609	0.5652
X10	0.1579	0.3158	0.5263

Table 3: Category Probabilities of Categorical Predictors in the Mixed-Predictor Scenario

Predictor	*Level*		
	A	**B**	**C**
X1	0.1818	0.5455	0.2727
X2	0.0909	0.2727	0.6364
X3	0.1111	0.3333	0.5556
X4	0.1250	0.3125	0.5625
X5	0.0769	0.3077	0.6154

The category probabilities were derived by normalizing the prespecified sampling weights for each predictor so that the probabilities within each predictor sum to one. The resulting category probabilities for the 10 categorical predictor variables and the categorical predictors in the mixed-predictor scenario are presented in Tables 2 and 3. Different probability configurations were specified across predictors to introduce variation in the marginal distribution of the categorical variables while maintaining a controlled data-generating process. These predefined probabilities were kept consistent across simulation replications to ensure reproducibility and comparability across experimental conditions.

Eq. (2) defines the normal distribution $N(\mu, \sigma^2)$ used to generate numerical predictors, with the mean and standard deviation values assigned subjectively. In this study, the mean and variance parameters for each numerical predictor variable were specified subjectively. The parameter values used for generating the numerical predictor variables are as follows: $X_6 \sim N(5,1)$, $X_7 \sim N(1,2.25)$, $X_8 \sim N(3,1)$, $X_9 \sim N(1,2.25)$, $X_{10} \sim N(3,2.25)$.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2}$$

(b) The response variable Y was generated with an imbalanced class proportion. The response variable Y was generated according to a logistic regression model to obtain a binary class outcome [16]. A random error term was generated from a normal distribution with mean and standard deviation using the rnorm(n, 0, 0.5) function. The error term was then added to the linear predictor to introduce random variability into the simulated data before transforming the linear predictor into response probabilities

using the logistic function. Thus, the response variable Y can be generated using Eq. (3) and Eq. (4).

$$P\left(Y=1\right)=\frac{e^{z}}{1+e^{z}} \tag{3}$$

$$z=\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_4X_4+\cdots+\beta_{10}X_{10}+\varepsilon \tag{4}$$

In simulation studies, the regression coefficients are not intended to estimate real-world relationships between predictor and response variables. Instead, they are used to define the data-generating mechanism by assigning different levels of influence to each predictor on the linear predictor. The coefficients were therefore selected to produce sufficient variability in the response probabilities, enabling the generation of binary outcomes with the desired class proportions. This approach facilitates a comprehensive evaluation of class imbalance handling methods under various simulation scenarios.

(c) The dataset consisting of variables Y and X is combined into a simulated dataset with a total of 5000 observations for each scenario.

(d) The imbalanced data simulation is performed by comparing the predicted probability values obtained from the logistic model with a specified threshold value. The threshold values were determined based on the quantiles of the simulated probability distribution, specifically the 70th, 80th, and 90th percentiles, to generate imbalanced class proportions of 70:30, 80:20, and 90:10, respectively. Observations with predicted probabilities exceeding the corresponding threshold were assigned to class 1, whereas those with probabilities less than or equal to the threshold were assigned to class 0. The thresholds for each scenario are presented in Table 4.

Table 4: Threshold

Class Proportion	Threshold
10 Categorical Variables	
70:30	• if $P(Y=1) > 0.999611$, then $Y=1$ • if $P(Y=1) \leq 0.999611$, then $Y=0$
80:20	• if $P(Y=1) > 0.999960$, then $Y=1$ • if $P(Y=1) \leq 0.999960$, then $Y=0$
90:10	• if $P(Y=1) > 0.999999$, then $Y=1$ • if $P(Y=1) \leq 0.999999$, then $Y=0$
5 Categorical and 5 Numeric Variables	
70:30	• if $P(Y=1) > 0.999971$, then $Y=1$ • if $P(Y=1) \leq 0.999971$, then $Y=0$
80:20	• if $P(Y=1) > 0.999994$, then $Y=1$ • if $P(Y=1) \leq 0.999994$, then $Y=0$
90:10	• if $P(Y=1) > 0.999999$, then $Y=1$ • if $P(Y=1) \leq 0.999999$, then $Y=0$

2. The data generation procedure was repeated 30 times independently for each simulation scenario to account for random variation and obtain stable estimates of model performance. A new dataset was generated in each repetition using the same data-generating mechanism and simulation settings.
3. The dataset was split into training (70%) and testing (30%) subsets to ensure a fair and robust evaluation of the model performance.
4. Data balancing techniques were applied to the training dataset using three methods:
   (a) Hybrid Sampling: a combination of undersampling and oversampling to obtain a more balanced class distribution.

(b) EasyEnsemble: a method that reuses majority-class samples discarded during undersampling, preserving majority-class information without introducing imbalance [6].

(c) SVM-SMOTE: The Support Vector Machine-Synthetic Minority Oversampling Technique (SVM-SMOTE) is an extension of the SMOTE method, designed to handle imbalanced datasets in classification problems. This method leverages the SVM algorithm to determine the decision boundary between the majority and minority classes [8].

5. Optimizing the model using Bayesian Optimization and training the model with the XGBoost algorithm by utilizing three evaluation metrics (balanced accuracy, sensitivity, and specificity).
6. Applying factorial ANOVA to the classification results by considering combinations of factor levels in variable X, in order to examine the main effects and interaction effects among the factors on model performance.

3. Results and Discussion

Data analysis begins with generating synthetic datasets based on different class proportions and predictor configurations, with 30 independent replications for each experimental scenario. Each dataset consists of 5,000 observations and is divided into training and testing sets using a 70:30 ratio. Hybrid Sampling, EasyEnsemble, and SVM-SMOTE are applied to the training data, followed by XGBoost classification with hyperparameters optimized using Bayesian Optimization. Model performance is evaluated based on Balanced Accuracy, Sensitivity, and Specificity. The results show that EasyEnsemble consistently achieves the best overall performance across all simulated scenarios, with all three evaluation metrics exceeding 0.90. Furthermore, three-way ANOVA indicates significant effects of the experimental factors and their interactions on model performance. These findings demonstrate that EasyEnsemble combined with XGBoost and Bayesian Optimization provides an effective approach for classifying imbalanced data under the simulation conditions considered in this study.

3.1. Data Generation

The response variable Y is generated using a logistic regression model to produce binary classes. The regression coefficients are assigned subjectively to provide weighting for each category of the predictor variables and are then combined with an error term generated from a normal distribution. Eq. (5) specifies the logistic regression model for the scenario involving 10 categorical predictor variables, while Eq. (6) specifies the logistic regression model for the scenario combining 5 categorical and 5 numerical predictor variables. The mathematical form of both models is presented as follows.

$$\begin{aligned} z_1 = & -4.001X_{1B} - 4.935X_{1C} + 4.138X_{2B} + 3.000X_{2C} \\ & - 4.050X_{3B} + 4.007X_{3C} + 2.346X_{4B} - 1.652X_{4C} \\ & + 3.764X_{5B} - 3.529X_{5C} - 2.723X_{6B} + 4.703X_{6C} \\ & + 2.863X_{7B} + 1.901X_{7C} + 4.444X_{8B} + 3.247X_{8C} \\ & - 4.063X_{9B} - 3.081X_{9C} + 1.743X_{10B} + 2.295X_{10C}. \end{aligned} \tag{5}$$

$$\begin{aligned} z_2 = & \, 2.406X_{1B} + 2.046X_{1C} + 2.453X_{2B} + 0.764X_{2C} \\ & - 3.528X_{3B} - 2.723X_{3C} + 1.001X_{4B} + 1.863X_{4C} \\ & + 2.000X_{5B} + 1.444X_{5C} + 1.247X_6 - 1.928X_7 \\ & - 1.810X_8 + 1.743X_9 + 1.295X_{10}. \end{aligned} \tag{6}$$

Their values were selected within a bounded range to provide heterogeneity in both the magnitude and direction of predictor effects. Specifically, the coefficients ranged from approximately -5 to 5, with both positive and negative values assigned across predictor categories. This specification was intended to avoid an overly homogeneous data-generating mechanism in which all predictors would affect the outcome in the same direction or with similar strength.

As a verification step for the conformity of the data generation process with the designed simulation scenarios, the class proportions for each scenario combination were visualized. Figure 1 illustrates the distribution of proportions between class 0 and class 1. This visualization demonstrates that the class imbalance ratios of 70:30, 80:20, and 90:10 were successfully replicated according to the simulation design, indicating that the generated data adequately represent the intended imbalance conditions.

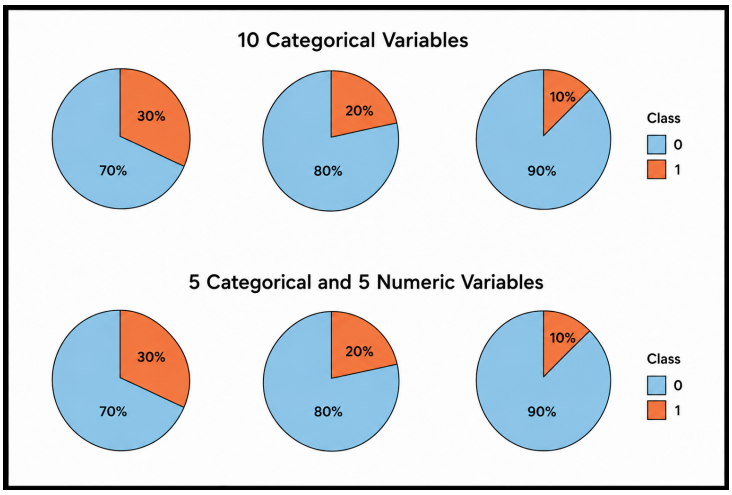


Fig. 1: Distribution of Class Y

3.2. Hyperparameter Optimization

Hyperparameter optimization was performed using Bayesian Optimization to determine the optimal hyperparameter configuration for the XGBoost model. The Bayesian Optimization procedure constructs a surrogate model to approximate the relationship between the XGBoost hyperparameters and the resulting model performance. In this study, a Gaussian Process (GP) was used as the surrogate model. The GP was sequentially updated using the observed performance from previously evaluated hyperparameter configurations, allowing the optimization procedure to identify promising regions of the hyperparameter space while accounting for predictive uncertainty.

The hyperparameters optimized in this study included eta (learning rate), max_depth, min_child_weight, subsample, colsample_bytree, and lambda [17]. To evaluate the contribution of Bayesian Optimization to the improvement of model performance, an initial XGBoost model was constructed using the default hyperparameter configuration adopted from [18]. The search domains were specified to constrain the optimization process to a predefined and computationally feasible range, while the default configuration was used to facilitate a direct comparison between the conventional XGBoost model and the Bayesian-optimized XGBoost model. Table 5 presents the default hyperparameter values and the corresponding search domains used in the Bayesian Optimization process.

The XGBoost model was evaluated using five-fold cross-validation for each candidate hyperparameter configuration. The model was specified with the gbtree booster and the binary:logistic objective, while the AUC was used as the optimization objective. Thus, the Bayesian Optimization procedure aimed to identify the hyperparameter configuration that maximized the mean test-fold AUC obtained from five-fold cross-validation. For each configuration, a maximum of 100 boosting rounds was considered, and the highest mean test AUC across the boosting rounds

Table 5: Hyperparameters of XGBoost

Hyperparameter	Domain	Default
eta	0.01–0.3	0.3
max_depth	3–10	6
min_child_weight	1–10	1
subsample	0.5–1	1
colsample_bytree	0.5–1	1
lambda	0.1–5	1

was returned as the objective score.

The Expected Improvement (EI) acquisition function was used to determine the subsequent hyperparameter configurations to be evaluated. The EI function considers both the predicted performance and the associated uncertainty from the Gaussian Process, thereby balancing exploitation of promising regions with exploration of less-evaluated regions of the hyperparameter space. At each optimization step, the hyperparameter configuration with the highest acquisition value was selected for evaluation, after which the observed AUC was incorporated into the surrogate model for the subsequent iteration.

The Bayesian Optimization procedure was initiated with 20 initial hyperparameter configurations (initPoints = 20) and continued for 20 sequential optimization iterations (iters.n = 20). These settings defined the optimization budget used to search for the optimal hyperparameter configuration. In addition, early stopping was applied within each XGBoost cross-validation run, with training terminated when the validation AUC did not improve for 10 consecutive boosting rounds (early stopping rounds = 10). The best hyperparameter values obtained from the Bayesian Optimization process for each of the 24 simulation scenarios are presented in Table 6.

Table 6: Optimal Hyperparameters Obtained from Bayesian Optimization

Handling	Class Proportion	eta	Max Depth	Min Child Weight	Subsample	Colsample Bytree	lambda
10 Categorical Predictor Variables							
Hybrid Sampling	70:30	0.283	3	4.246	0.907	0.725	0.881
	80:20	0.281	3	1	0.575	0.945	0.1
	90:10	0.2	3	1.680	0.689	1	0.1
EasyEnsemble	70:30	0.3	3	3.483	0.766	0.913	0.1
	80:20	0.282	3	3.656	0.697	0.898	2.308
	90:10	0.299	3	1.925	0.5	1	0.1
SVM-SMOTE	70:30	0.232	3	6.202	0.685	1	0.1
	80:20	0.271	7	3.569	0.682	0.907	0.1
	90:10	0.274	9	2.148	0.566	0.506	0.756
5 Categorical & 5 Numerical Variables							
Hybrid Sampling	70:30	0.3	6	6.949	1	0.589	5
	80:20	0.3	3	1	1	0.735	0.1
	90:10	0.230	10	1	0.686	1	4.630
EasyEnsemble	70:30	0.269	5	5.033	0.847	1	3.930
	80:20	0.281	3	1.142	0.686	0.750	2.799
	90:10	0.219	3	2.697	0.884	0.716	0.447
SVM-SMOTE	70:30	0.242	5	7.336	1	1	0.1
	80:20	0.3	3	3.585	1	1	0.1
	90:10	0.284	3	1	0.883	1	2.976

3.3. Comparison of the Performance of Data Imbalance Handling Methods

An important stage in this research that must be considered prior to the modeling process is the handling of imbalanced data, as imbalance may cause the model to become biased toward

the majority class and reduce its ability to recognize minority class instances. Therefore, three imbalance handling methods, Hybrid Sampling, EasyEnsemble, and SVM-SMOTE, were applied across three class proportion scenarios (70:30, 80:20, 90:10). Table 7 presents the resulting class proportions after applying the three methods, both for datasets consisting of 10 categorical independent variables and for a combined dataset of 5 categorical and 5 numerical independent variables. The imbalance handling procedures were performed after splitting the dataset into 70% training and 30% testing sets. This stage serves as a critical foundation before model development to ensure that the resulting model can provide more accurate and representative predictions for both classes.

Table 7: Distribution of Class Y

Class Proportion (%)	**Hybrid Sampling (%)**		**EasyEnsemble (%)**		**SVM-SMOTE (%)**	
	0	**1**	**0**	**1**	**0**	**1**
70:30	50,2	49,8	50	50	56,2	43,8
80:20	50,8	49,2	50	50	57,1	42,9
90:10	50,5	49,5	50	50	75	25

Table 7 demonstrates that the Hybrid Sampling method produces a relatively balanced class distribution between class 0 and class 1 across all three proportions. The EasyEnsemble method shows a perfectly balanced distribution, with 50% representation for each class in all class proportions. This indicates that the ensemble technique is capable of maintaining class balance effectively by generating proportional data subsets between the majority and minority classes. Meanwhile, the SVM-SMOTE method yields a reasonably balanced distribution in the 70:30 and 80:20 proportions; however, in the 90:10 proportion, although an improvement is observed after the imbalance handling procedure, the distribution cannot yet be considered balanced. Overall, EasyEnsemble provides the optimal class balance compared to Hybrid Sampling and SVM-SMOTE.

The XGBoost model was applied to evaluate the performance of three class imbalance handling methods on the simulation dataset. Model performance was assessed based on three primary evaluation metrics: balanced accuracy, sensitivity, and specificity. The evaluation process was repeated 30 times to obtain more stable and consistent results.

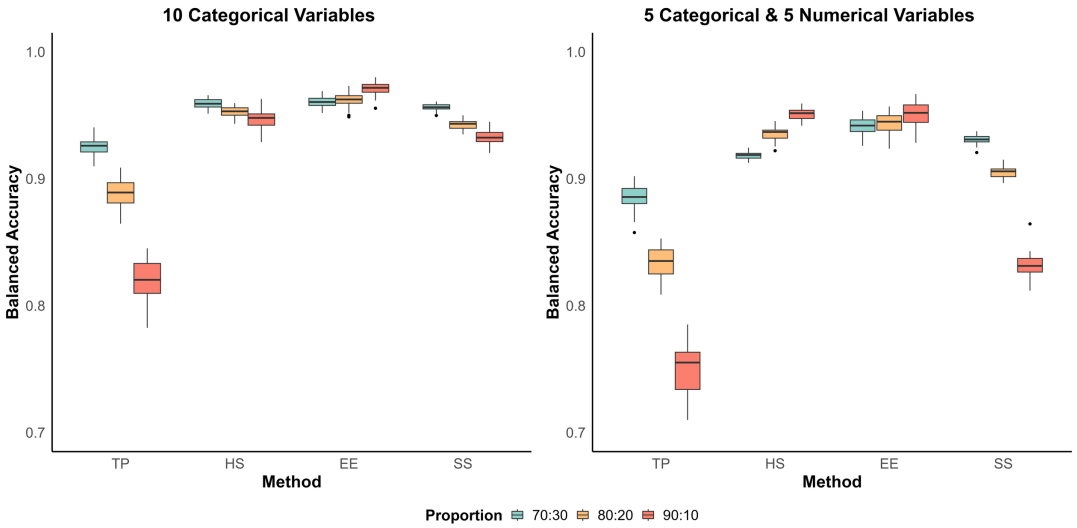


Fig. 2: Boxplot of Balanced Accuracy

Balanced accuracy is employed to evaluate a model's ability to classify majority and minority classes in a balanced manner, particularly under imbalanced data conditions [19]. As illustrated

in Figure 2, the XGBoost model without class imbalance handling (TP) exhibits a pronounced decline in performance as the degree of class imbalance increases. This pattern indicates that the model tends to be biased toward the majority class. In contrast, the application of class imbalance handling methods generally improves model performance. EasyEnsemble (EE) demonstrates the best and most stable performance across all class proportions, both for datasets consisting solely of categorical variables and for those combining categorical and numerical variables. Hybrid Sampling (HS) also yields a substantial improvement, although its performance slightly deteriorates under more extreme class imbalance. SVM-SMOTE (SS) likewise enhances balanced accuracy; however, its consistency remains inferior to that of EasyEnsemble (EE).

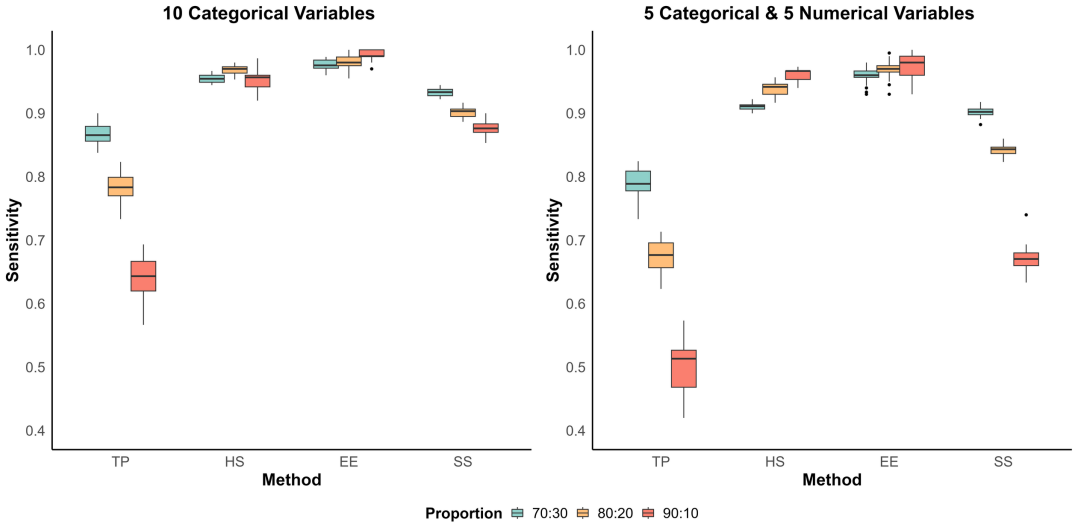


Fig. 3: Boxplot of Sensitivity

Figure 3 shows that the sensitivity of the model without class imbalance treatment (TP) exhibits very poor performance, particularly as the proportion of the minority class decreases, indicating the model's failure to adequately detect the minority class [20]. After applying class imbalance handling methods, sensitivity improves substantially. EasyEnsemble (EE) consistently achieves the highest sensitivity across all scenarios, demonstrating its strong capability in identifying the minority. Hybrid Sampling (HS) and SVM-SMOTE (SS) also lead to significant improvements in sensitivity; however, their performance tends to decline at the 90:10 class proportion, both for datasets with 10 categorical independent variables and for those with a combination of independent variables.

Specificity represents the ability of a model to correctly classify observations belonging to the majority class [21]. Figure 4 shows that the specificity of the XGBoost model without class imbalance treatment (TP) tends to be very high across all scenarios. However, this condition does not reflect good overall model performance; rather, it indicates a strong tendency of the model to predominantly predict the majority class, thereby neglecting the minority class. After applying class imbalance handling methods, specificity slightly decreases but remains at a high level. This decrease reflects a reduced bias toward majority class proportions, leading to a more balanced classification performance between the majority and minority classes.

Overall, based on the boxplots of the three evaluation metrics presented, EasyEnsemble consistently outperforms the other imbalance-handling methods. This finding reinforces that EasyEnsemble is the most effective method for addressing data imbalance, both when the dataset comprises categorical independent variables and when it includes a combination of categorical and numerical independent variables, as it is able to produce more stable and consistent performance outcomes. The detailed evaluation results for each imbalance-handling method are summarized in Table 8 and Table 9, providing a comprehensive comparison of their performance.

In this study, a three-way factorial Completely Randomized Design (CRD) was employed with

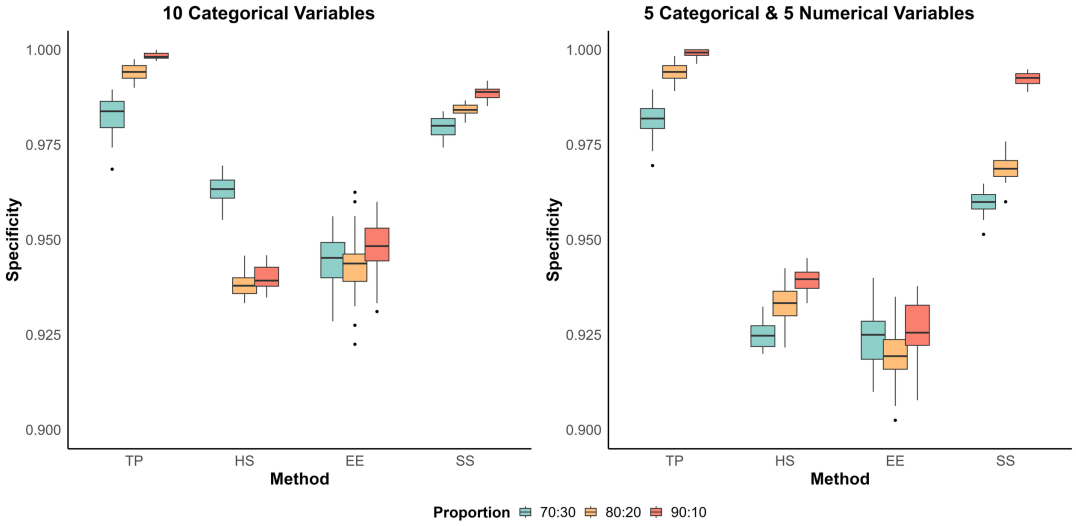


Fig. 4: Boxplot of Specificity

Table 8: Average Performance of XGBoost with 10 Categorical Independent Variables

Handling Method	Class Proportion	*Balanced Accuracy*	Sensitivity	Specificity
Without handling	70:30	0.9255	0.8681	0.9828
	80:20	0.8892	0.7844	0.9939
	90:10	0.8201	0.6420	0.9982
Hybrid Sampling	70:30	0.9567	0.9467	0.9667
	80:20	0.9530	0.9670	0.9380
	90:10	0.9460	0.9520	0.9400
EasyEnsemble	70:30	0.9604	0.9766	0.9443
	80:20	0.9624	0.9818	0.9429
	90:10	0.9707	0.9930	0.9483
SVM-SMOTE	70:30	0.9562	0.9329	0.9796
	80:20	0.9371	0.8900	0.9842
	90:10	0.9324	0.8762	0.9886

Table 9: Average Performance of XGBoost with a Combination of 5 Categorical and 5 Numerical Independent Variables

Handling Method	Class Proportion	*Balanced Accuracy*	Sensitivity	Specificity
Without handling	70:30	0.8856	0.7896	0.9815
	80:20	0.8351	0.6762	0.9939
	90:10	0.7493	0.4996	0.9989
Hybrid Sampling	70:30	0.9180	0.9110	0.9250
	80:20	0.9350	0.9380	0.9330
	90:10	0.9510	0.9620	0.9400
EasyEnsemble	70:30	0.9419	0.9592	0.9246
	80:20	0.9436	0.9695	0.9177
	90:10	0.9503	0.9750	0.9257
SVM-SMOTE	70:30	0.9310	0.9020	0.9599
	80:20	0.9051	0.8414	0.9687
	90:10	0.8313	0.6704	0.9922

three primary factors: class proportion (A), independent variable type (B), and class imbalance handling method (C). The analysis was performed for each treatment combination using balanced

accuracy, sensitivity, and specificity obtained from the XGBoost model. Subsequently, a three-way factorial analysis of variance (ANOVA) was applied to evaluate the main effects of each factor as well as the interaction effects among the three factors on model performance.

Table 10: ANOVA Results for Balanced Accuracy

Factor	df	Sum of Squares	Mean Square	F-statistic	p-value
A	2	0.2068	0.1034	1375.88	$< 2e{-}16$
B	1	0.2422	0.2422	3222.93	$< 2e{-}16$
C	3	1.1753	0.3918	5213,19	$< 2e{-}16$
A × B	2	0.0095	0.0047	63.08	$< 2e{-}16$
A × C	6	0.3678	0.0613	815.80	$< 2e{-}16$
B × C	3	0.0591	0.0197	262.28	$< 2e{-}16$
A × B × C	6	0.0628	0.0105	139.36	$< 2e{-}16$
Error	696	0.0523	0.0001	—	—

Table 11: ANOVA Results for Sensitivity

Factor	df	Sum of Squares	Mean Square	F-statistic	p-value
A	2	1.024	0.5122	1774.0	$< 2e{-}16$
B	1	0.679	0.6793	2352.9	$< 2e{-}16$
C	3	7.749	2.5829	8946.4	$< 2e{-}16$
A × B	2	0.072	0.0362	125.4	$< 2e{-}16$
A × C	6	1.689	0.2814	974.8	$< 2e{-}16$
B × C	3	0.333	0.1111	384.7	$< 2e{-}16$
A × B × C	6	0.245	0.0409	141.5	$< 2e{-}16$
Error	696	0.201	0.0003	—	—

Table 12: ANOVA Results for Specificity

Factor	df	Sum of Squares	Mean Square	F-statistic	p-value
A	2	0.0109	0.0054	238.35	$< 2e{-}16$
B	1	0.0256	0.0256	1123.87	$< 2e{-}16$
C	3	0.4386	0.1462	6412.39	$< 2e{-}16$
A × B	2	0.0068	0.0034	149.21	$< 2e{-}16$
A × C	6	0.0147	0.0024	107.14	$< 2e{-}16$
B × C	3	0.0116	0.0039	170.19	$< 2e{-}16$
A × B × C	6	0.0108	0.0018	78.99	$< 2e{-}16$
Error	696	0.0159	0.00002	—	—

The ANOVA conducted on the three evaluation metrics presented in Tables 10, 11, and 12 indicates that all metrics exhibit a statistically significant effect on the interaction among class proportion, independent variable types, and imbalance-handling methods. This finding implies that variations in class proportion, types of independent variables, and handling methods, as well as the condition without any handling, lead to different impacts on the performance of the XGBoost model. Therefore, based on the ANOVA results, the presence of significant differences necessitated a Tukey test to identify which group pairs demonstrate statistically significant differences across each evaluation metric.

Based on Tukey's post hoc test, the best performance in the balanced accuracy metric was achieved by the combination of 10 categorical independent variables with a class proportion of 90:10 using the EasyEnsemble method, which demonstrated the highest balanced accuracy among all evaluated combinations. In terms of sensitivity, the best performance was also observed for the combination with 10 categorical independent variables at a class proportion of 90:10 using the EasyEnsemble method. Meanwhile, for the specificity metric, the best performance was observed for the combination of five categorical and five numerical independent variables with a class proportion of 90:10, under the scenario without any class imbalance handling method.

Although the specificity obtained when applying EasyEnsemble was lower than that achieved by the combinations without imbalance handling, this method provided an optimal balance among balanced accuracy, sensitivity, and specificity, thereby improving the recognition of both minority and majority classes. Overall, these results demonstrate that EasyEnsemble provides consistently superior performance over other approaches and the no-handling baseline.

4. Conclusion

Across all evaluated simulation scenarios, the EasyEnsemble method consistently demonstrated the best performance in handling class imbalance. For all imbalanced class proportions (70:30, 80:20, and 90:10), under both predictor variable settings (10 categorical predictor variables and a combination of 5 categorical and 5 numerical predictor variables), EasyEnsemble achieved balanced accuracy, sensitivity, and specificity values exceeding 0.90. These results indicate that EasyEnsemble provides an optimal balance in classifying both majority and minority classes, making it the most reliable class imbalance handling method evaluated in this study. Furthermore, Bayesian Optimization efficiently identified the optimal XGBoost hyperparameter combinations for each simulation scenario, thereby contributing to improved model performance.

However, the finding that EasyEnsemble achieved the best performance should be interpreted within the scope of the simulation settings considered in this study and should not be generalized to all types of datasets or data-generating conditions. Specifically, under the predictor structures, class proportions, and simulation mechanisms evaluated in this study, EasyEnsemble demonstrated better performance than the other class imbalance handling methods considered. Therefore, the results suggest that EasyEnsemble is a suitable method for data with characteristics similar to those represented in the simulation scenarios of this study, rather than indicating that it is universally superior for all imbalanced datasets.

CRediT Authorship Contribution Statement

Megawati Zein Waliulu contributed to methodology, software development, formal analysis, data curation, original draft writing, and visualization. **Kusman Sadik**, as the main supervisor, contributed to conceptualization, supervision, validation, and review & editing of the manuscript. **Asep Saefuddin** contributed to conceptualization, supervision, and manuscript review.

Declaration of Generative AI and AI-assisted technologies

The author used the generative AI tool ChatGPT (version 4, OpenAI) to support idea development and exploration in code writing and LaTeX formatting. In addition, Grammarly was used to assist with grammar checking and refinement of the English text. All outputs generated by these tools were manually reviewed and adjusted to ensure accuracy, originality, and academic integrity.

Declaration of Competing Interest

The authors declare no competing interests.

Funding and Acknowledgments

This research was supported by the Department of Statistics and Data Science, School of Data Science, Mathematics, and Informatics (SSMI), IPB University. The author also acknowledges financial support from the Indonesia Endowment Fund for Education (LPDP), Ministry of Finance of the Republic of Indonesia. The author thanks Kusman Sadik and Asep Saefuddin for their valuable guidance and supervision.

Data Availability

The data and R scripts used in this study were generated and executed using R version 4.3.3. The data supporting the findings of this study are available from the corresponding author upon reasonable request.

References

[1] B. Krawczyk. "Learning from imbalanced data: open challenges and future directions". In: *Progress in Artificial Intelligence* 5.4 (2016), pp. 221–232. DOI: https://doi.org/10.1007/s13748-016-0094-0.

[2] D. Dablain, B. Krawczyk, and N. V. Chawla. "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data". In: *IEEE Trans Neural Netw Learn Syst* 34.9 (2023), pp. 6390–6404. DOI: https://doi.org/10.1109/TNNLS.2021.3136503.

[3] M. Piccininni, M. Wechsung, B. Van Calster, J. L. Rohmann, S. Konigorski, and M. van Smeden. "Understanding random resampling techniques for class imbalance correction and their consequences on calibration and discrimination of clinical risk prediction models". In: *J Biomed Inform* 155.May (2024), p. 104666. DOI: https://doi.org/10.1016/j.jbi.2024.104666.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357. DOI: https://doi.org/10.1613/jair.953.

[5] J. L. Leevy, J. M. Johnson, J. Hancock, and T. M. Khoshgoftaar. "Threshold optimization and random undersampling for imbalanced credit card data". In: *J. Big Data* 10.1 (2023). DOI: https://doi.org/10.1186/s40537-023-00738-z.

[6] X. Y. Liu, J. Wu, and Z. H. Zhou. "Exploratory under-sampling for class-imbalance learning". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.2 (2009), pp. 539–550. DOI: https://doi.org/10.1109/TSMCB.2008.2007853.

[7] P. Zhang, Y. Jia, and Y. Shang. "Research and application of XGBoost in imbalanced data". In: *Int J Distrib Sens Netw* 18.6 (2022). DOI: https://doi.org/10.1177/15501329221106935.

[8] S. S. Bagui, D. Mink, S. C. Bagui, and S. Subramaniam. "Determining Resampling Ratios Using BSMOTE and SVM-SMOTE for Identifying Rare Attacks in Imbalanced Cybersecurity Data". In: *Computers* 12.10 (2023). DOI: https://doi.org/10.3390/computers12100204.

[9] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla. "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary". In: *Journal of artificial intelligence research* 61 (2018), pp. 863–905. DOI: https://doi.org/10.1613/jair.1.11192.

[10] M. A. K. Raiaan, S. Sakib, N. M. Fahad, A. Al Mamun, M. A. Rahman, S. Shatabda, and M. S. H. Mukta. "A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks". In: *Decision analytics journal* 11 (2024), p. 100470. DOI: https://doi.org/10.1016/j.dajour.2024.100470.

[11] M. I. Sameen, B. Pradhan, and S. Lee. "Application of convolutional neural networks featuring Bayesian optimization for landslide susceptibility assessment". In: *Catena* 186 (2020), p. 104249. DOI: https://doi.org/10.1016/j.catena.2019.104249.

[12] M. ZLOBIN and V. BAZYLEVYCH. "Bayesian Optimizatio for Tuning Hyperparametrs of Machine Learning Models: a Performance Analysis in Xgboost". In: *Computer systems and information technologies* 1 (2025), pp. 141–146. DOI: https://doi.org/10.31891/csit-2025-1-16.

[13] T. Chen and C. Guestrin. "XGBoost: A scalable tree boosting system". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13 (2016), pp. 785–794. DOI: https://doi.org/10.1145/2939672.2939785.

[14] J. Zhou, F. Chen, A. Khattak, and S. Dong. "Interpretable ensemble-imbalance learning strategy on dealing with imbalanced vehicle-bicycle crash data: A case study of Ningbo, China". In: *International Journal of Crashworthiness* 29.5 (2024), pp. 884–897. DOI: https://doi.org/10.1080/13588265.2024.2316924.

[15] Q. A. Hidayaturrohman and E. Hanada. "A Comparative Analysis of Hyper-Parameter Optimization Methods for Predicting Heart Failure Outcomes". In: *Applied Sciences (Switzerland)* 15.6 (2025), pp. 1–17. DOI: https://doi.org/10.3390/app15063393.

[16] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant. *Applied Logistic Regression*. John Wiley and Sons, 2013.

[17] A. Mehdary, A. Chehri, A. Jakimi, and R Saadane. "Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection". In: *Sensors* 24.4 (2024). DOI: https://doi.org/10.3390/s24041230.

[18] F. Inceoglu and P. T. M. Loto'aniu. "Utilizing XGBoosts to correct arcjet contamination in magnetic field measurements from GOES missions". In: *Frontiers in Artificial Intelligence* 8 (2025). DOI: https://doi.org/10.3389/frai.2025.1628029.

[19] A. Luque, A. Carrasco, A Martín, and A de Las Heras. "The impact of class imbalance in classification performance metrics based on the binary confusion matrix". In: *Pattern Recognition* 91 (2019), pp. 216–231. DOI: https://doi.org/10.1016/j.patcog.2019.02.023.

[20] R. Trevethan. "Sensitivity, Specificity, and Predictive Values: Foundations, Pliabilities, and Pitfalls in Research and Practice". In: *Frontiers Public Health* 5 (2017), pp. 1–7. DOI: https://doi.org/10.3389/fpubh.2017.00307.

[21] G. Grunau and S. Linn. "Commentary: Sensitivity, Specificity, and Predictive Values: Foundations, Pliabilities, and Pitfalls in Research and Practice". In: *Frontiers Public Health* 6 (2018), pp. 1–4. DOI: https://doi.org/10.3389/fpubh.2018.00256.