Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 9, No. 2, June 2023, pp. 223-238
ISSN: 2338-3070, DOI: 10.26555/jiteki.v9i2.25929

Evaluating Sampling Techniques for Healthcare Insurance Fraud Detection in Imbalanced Dataset

Joanito Agili Lopo, Kristoko Dwi Hartomo
Satya Wacana Christian University, Jl. Diponegoro 52-60, Salatiga – Indonesia 50711

ARTICLE INFO

Article history:

Received March 10, 2023
Revised April 13, 2023
Published April 18, 2023

Keywords:

Healthcare Insurance;
Imbalanced Dataset;
Oversampling;
XGBoost;
Fraud Detection;
Undersampling

ABSTRACT

Detecting fraud in the healthcare insurance dataset is challenging due to severe class imbalance, where fraud cases are rare compared to non-fraud cases. Various techniques have been applied to address this problem, such as oversampling and undersampling methods. However, there is a lack of comparison and evaluation of these sampling methods. Therefore, the research contribution of this study is to conduct a comprehensive evaluation of the different sampling methods in different class distributions, utilizing multiple evaluation metrics, including AUC_{ROC}, $G-mean$, $F1_{macro}$, Precision, and Recall. In addition, a model evaluation approach be proposed to address the issue of inconsistent scores in different metrics. This study employs a real-world dataset with the XGBoost algorithm utilized alongside widely used data sampling techniques such as Random Oversampling and Undersampling, SMOTE, and Instance Hardness Threshold. Results indicate that Random Oversampling and Undersampling perform well in the 50% distribution, while SMOTE and Instance Hardness Threshold methods are more effective in the 70% distribution. Instance Hardness Threshold performs best in the 90% distribution. The 70% distribution is more robust with the SMOTE and Instance Hardness Threshold, particularly in the consistent score in different metrics, although they have longer computation times. These models consistently performed well across all evaluation metrics, indicating their ability to generalize to new unseen data in both the minority and majority classes. The study also identifies key features such as costs, diagnosis codes, type of healthcare service, gender, and severity level of diseases, which are important for accurate healthcare insurance fraud detection. These findings could be valuable for healthcare providers to make informed decisions with lower risks. A well-performing fraud detection model ensures the accurate classification of fraud and non-fraud cases. The findings also can be used by healthcare insurance providers to develop more effective fraud detection and prevention strategies.

Corresponding Author:

Joanito Agili Lopo, Satya Wacana Christian University, Jl. Diponegoro 52-60, Salatiga – Indonesia 50711
Email: 682019013@student.uksw.edu

1. INTRODUCTION

Fraud is the illegal use or intentional decision of resources or assets to gain an advantage. In healthcare insurance, fraud involves purposeful deception to receive benefits or advantages through the insurance process. Deception refers to hiding or distorting details concerning the outcomes of medical benefits and disregarding the guidelines of conventional medical practices [1]–[3]. Healthcare insurance fraud can occur at any stage of an insurance transaction by an individual applying for insurance, third-party claimant, or policyholders. The consequences of fraud in healthcare insurance can lead to compromised quality of healthcare services and facilities, financial loss, and risk to patient safety. For instance, in 2012, healthcare insurance fraud resulted in the misappropriation of public funds amounting to an estimated 17 billion to 57 billion [4]. In 2016, a major

German public medical insurance company was found to have €7 million in insurance anomalies. In the same year, the U.S. Department of Justice prosecuted the most significant medical anomaly case in its history, resulting in losses of up to $900 million and involving over 300 individuals, including healthcare professionals [5]. The need to address this issue has driven many researchers to develop a fraud detection model to detect healthcare insurance fraud.

The main idea in fraud detection is to identify the possible model of fraud incorporated with known fraudsters and predict the probability of a new transaction being fraudulent. There are two sets of models in fraud detection: Expert-driven, which require domain knowledge from the investigator to define rules, and Data-driven models, which use some techniques that produce a rule based on data. In standard, Data-driven models tend to represent a more practical approach to addressing the pattern and trends related to fraudulent behaviour, more accurate detection and achieved good performance [4], [6], [7]. Moreover, a data-driven model commonly includes machine learning and statistical methods and has shown promising results [3], [8]. The machine learning method is a technique that learns from the raw data and makes predictions without being explicitly programmed. Moreover, there are still many challenges that arise with these methods. One of the challenges is an imbalanced problem, defined as a condition where the distribution over each class in data is uneven.

Further, the imbalanced dataset will have the ratio between one class much lower than the one's other classes. It causes the underrepresented of classes in the dataset. Since the rate of fraudulent transactions is usually low or rare, machine learning methods prone to discard learning or undetected about fraudulent transaction patterns [7]. The methods could produce good accuracy, but on the other hand, the classification of the rare or the minority class is much lower than the majority. As a result, the classifier may favour the majority and ignore the minority, leading to it being treated as noise and ignored during classification. The problems can negatively affect machine learning techniques, particularly in applications with vast amounts of imbalanced data. The healthcare sector has produced significant data from the patients, provider payment, and claim data. This matter makes the complexity and noises attached to the data, which has fewer cases outside of everyday activities, in this case, less fraud than non-fraud.

Researchers have categorized methods for handling class imbalance into four approaches [9]–[12]. The first approach is algorithm-level, focusing on the classifier learning algorithm to adapt it toward the minority class. It will modify the classifier learning procedure to alleviate majority class bias instead of altering the supplied training set. The second is data-level approaches. This approach will modify a set of imbalanced class distributions using different procedures (i.e., sampling methods) to provide balanced or more adequate data. The third is the Cost-sensitive learning approach, which adds cost to the sample and modifies the learning process to accept costs. The cost aims to minimize the conditional risk. The last is the ensemble-based method, which combines an ensemble algorithm and one of the techniques above, either data-level or cost-sensitive. The data Level approach has become standardized over the years [10]. Therefore this study will mainly focus on the Data level approach, which can be categorized into groups or families [10], [13]. The first group is undersampling methods. The undersampling method creates a subset of the original dataset using random choice or cleaning techniques to decrease the majority class amount and balance the class distributions. The second group is Oversampling methods, which create a superset of the original dataset by replicating some instances or producing new instances based on the space in the data to balance the class. Hybrid methods combine both sampling approaches.

Machine learning has significantly increased fraud detection research in recent years, particularly in using sampling methods to address the imbalanced data issue. Some studies have been applied and resulted in significant performance. A study [12] proposed a novel under-sampling technique using the DBSCAN algorithm to select suitable samples from the majority class in imbalanced datasets. They used classification accuracy and F-measure scores to evaluate the method. The experiment result showed that this method outperformed six other preprocessing methods, including over-sampling, under-sampling, and hybrid approaches. A related study in [6] proposed a method that combined clustering analysis and instance selection to reduce the number of data samples in the majority class while preserving the important information. The experiment showed that the proposed method outperformed the six baseline approaches regardless of the type of combination of techniques used. In similar works [14], the authors introduced an under-sampling method that utilizes the Support Vectors classifier to identify the most informative majority class instance. This approach helps to generate decision boundaries for the model. They used a single evaluation method, the Receiver Operating Characteristic (ROC) curve, to evaluate several classifiers on 13 imbalanced datasets. The result showed that the proposed method produces a high score when classifying minority and majority class instances compared to other existing methods.

SMOTE is one of the popular methods to address the imbalanced dataset. This method has many modifications to increase the performance, such as in [15]. The author proposed a new algorithm that extends

the SMOTE with the Kalman filter to handle imbalanced datasets and reduce the size of the data by filtering out noisy samples. The performance of the proposed method is evaluated using Accuracy, AUC, Precision, Recall, and F1 score metrics on several imbalanced datasets. However, the inconsistent scores of each metric make it difficult to generalize the significant performance of this method. For instance, scores performed in datasets for AUC, Recall, F1 score, and Precision were 0.87, 0.75, 0.6, and 0.5, respectively. A previous study [16] encountered a similar issue. The authors proposed the NUS method, which involves clustering the minority class samples and removing noisy samples from both minority and majority classes. However, inconsistencies in scores for each metric were also observed in this study.

An effective machine learning algorithm and appropriate resampling techniques could enhance the accuracy of the fraud detection model. Therefore, a study in [17] used PCA for dimension reduction and Artificial Neural Networks as their classifier. Precision, Recall, and AUC (Area Under Curve) were used to evaluate the SMOTE oversampling method. The result showed average precision of 85.3%, recall of 73%, and AUC of 0.864 reached after performing SMOTE oversampling and genetic algorithm to optimize and avoid local minima. Meanwhile, in [18], they proposed a new sampling method called NC_Link_MWMOTE. This method modified the MWMOTE hierarchical clustering method by considering the distribution of minority samples and the distance between samples. The six datasets used the F1 score, recall, and precision as the model evaluation. The study concludes that the proposed method can effectively improve the classification algorithm. For instance, in Yeast Dataset, the F1, recall, and precision scores are 0.79, 0.68, and 0.93, respectively, against the previous method's 0.77, 0.66, and 0.93. Furthermore, a study cited as [19] utilized several machine learning algorithms to achieve an accuracy score of 97.43%, a 0.06% of precision score, a perfect recall score of 100%, and 11.82%, 0.98 scores of F1 and AUC score, particularly in the Random Forest classifier with the Random Under-sampling technique.

The literature described above shows that various techniques have been proposed to address the imbalanced data issue in fraud detection. However, there is a lack of comparison and evaluation of the different sampling methods, which makes it challenging to determine the most effective method for addressing the imbalanced data issue. Furthermore, using different evaluation metrics makes it difficult to generalize the results. It provides an unfair comparison of the most efficient sampling method. Therefore, this research addresses this gap by comprehensively evaluating the different sampling methods, mainly using Random Undersampling, Random Oversampling, SMOTE, and Instance Hardness Threshold. The evaluation will be conducted using popular evaluation metrics, including AUC_{ROC}, $G-mean$, $F1_{macro}$, Precision, and Recall, to compare the methods' effectiveness. In addition, this research proposed a model evaluation approach to address the issue of inconsistent scores when using multiple evaluation metrics, which keeps a high score across the metric. The study will be used a real-world insurance claim dataset from the Indonesia Social Security Agency of Health (BPJS). The XGBoost algorithm will be employed to perform analysis, and its hyperparameter will be tuned to optimize its performance. The research contributions of this study can be summarized as follows:

- A comprehensive evaluation of four different sampling methods to address the imbalanced data issue in fraud detection
- The use of multiple evaluation metrics, including ROC-AUC, G-MEAN, PR-AUC, Precision, Recall, and F1-score, to compare the effectiveness of the sampling methods
- A proposed model evaluation approach to address the issue of inconsistent scores when using multiple evaluation metrics, which maintains high scores across the metrics
- The application of the XGBoost algorithm and hyperparameter tuning to optimize model performance, particularly in the highly imbalanced dataset
- The use of a real-world insurance claim dataset from the Indonesia Social Security Agency of Health (BPJS) for analysis

2. METHODS

Fig. 1 shows the process of the entire experiment. It starts with preprocessing the real-world data, performing sampling methods, tuning the XGBoost hyperparameter, training the model, and evaluating it using multiple metrics. This section consists of five parts: the first part is to describe the data source; the second describes data preprocessing; the third sampling method; the fourth model classifiers; and the fifth part describes the proposed model evaluation metrics.

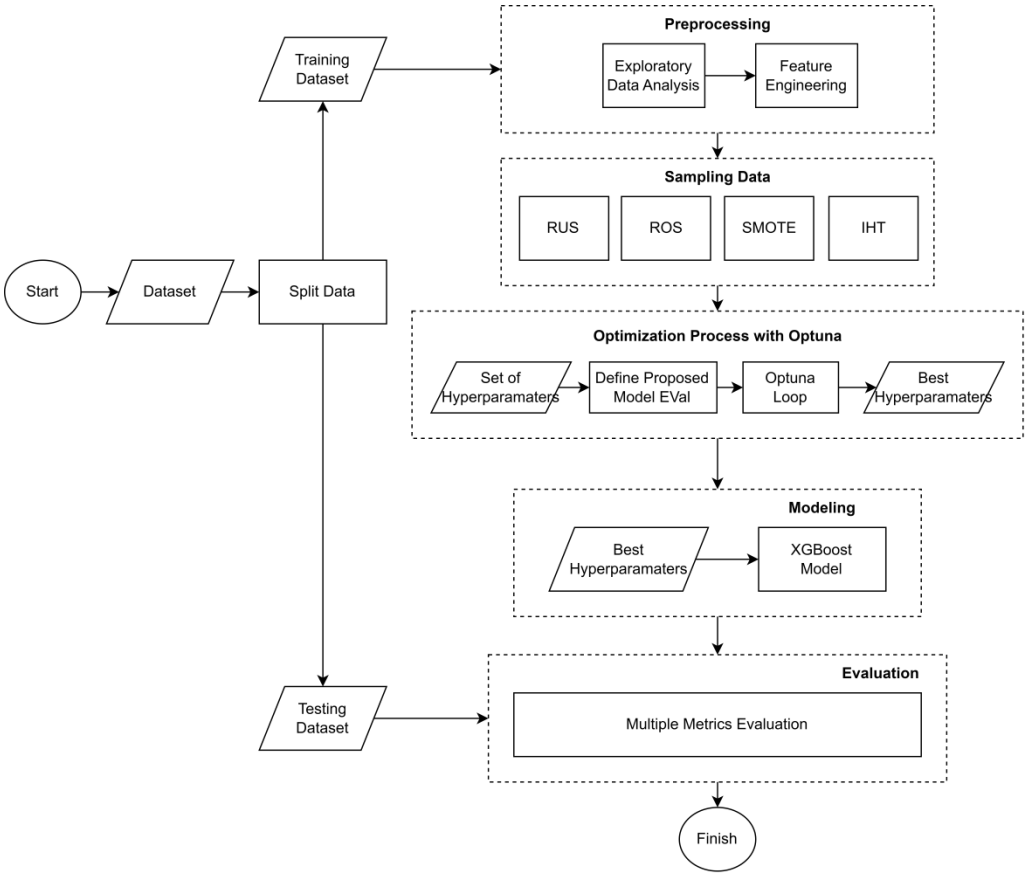

Fig. 1. The Stages of the Research

2.1. Dataset

This study will use The Healthcare Insurance Claim dataset from the Indonesia Social Security Agency of Health (BPJS). The dataset contains insurance claim records that the healthcare service providers claimed to the BPJS. The dataset includes data from multiple years and covers many healthcare service providers across Indonesia. The dataset consists of 2,400,791 instances with 27 features, categorized into two classes: the majority class (class 0) with 2,244,297 instances and the minority class (class 1) with 156,494 instances. The majority class represents non-fraudulent or efficient transactions, while the minority class represents fraudulent or non-efficient transactions. The dataset is highly imbalanced, with the percentage of each class being 94% for class 0 and 6% for class 1. Fig. 2 shows an illustration of the distribution of both classes. The data will be partitioned into training and testing sets with a ratio of 80:20, respectively. Using stratified sampling can ensure that both training and testing sets represent the imbalanced nature of data. Stratified sampling keeps the number of samples in each split proportional to the class area [20]. This approach enables the model to generalize to new unseen data and evaluate its performance on a separate data set while preserving the representation of the minority class.

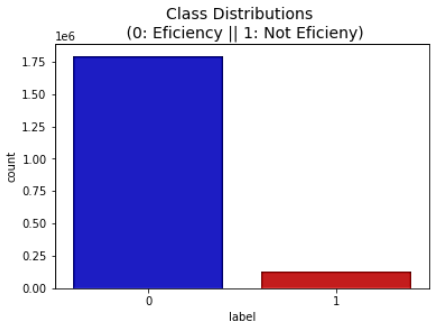

Fig. 2. Class Distributions

2.2. Data Preprocessing

Obtaining relevant information from the data is a crucial step. Factors such as skewed distribution, unequal classes, and multiple overlaps in the imbalanced dataset can impact the model's performance. It is essential to implement data preprocessing to ensure the data is proper and suitable for modeling. Fig. 3 shows the phases of the data preprocessing.

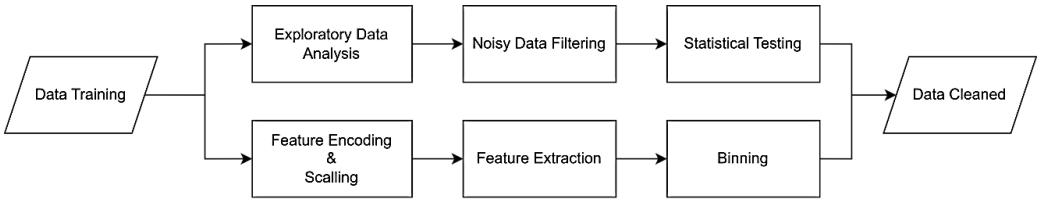

Fig. 3. The Preprocessing Phase

Exploratory Data Analysis (EDA) is an essential step to understanding the data, cleaning and transforming it into a more suitable format for modeling. Performed EDA allows for data understanding, creating hypotheses for the analysis, and making informed decisions [21]. This study will examine the summary of the statistical measures and use various data visualizations, such as histograms, box plots, and heatmaps, to gain the data distribution, identify patterns and correlation matrix, and analyze feature relationships. This step enables the detection of outliers or anomalies, providing insights into the nature of data and the appropriate way to handle each feature.

Noisy Data Filtering. Reducing the noise is crucial to ensure accurate and valid data, particularly in real-world data. It could make the model effective in identifying fraud. The noise in the dataset could be an unuseful column, outlier, and missing values. In this study, columns that do not relate to the task will be dropped. Features with less than 10% missing values are kept in the analysis to maintain sufficient data. Meanwhile, this study keeps the outlier with further processing because each feature's outliers may represent fraudulent data. This study uses the interquartile range (IQR) method to detect each feature's outliers to be analyzed. Let $Q1_i$ be the first quartile of the *i*-th feature, $Q3_i$ be the third quartile of the *i*-th feature, and IQR_i be the interquartile range (i.e, IQR = Q3 – Q1). Any data points below the lower bound or above the upper bound are considered an outlier. The calculation of the lower and upper bounds are using (1) and (2).

$$Lower\ Bound_i = Q1_i - 1.5 * IQR_i \tag{1}$$

$$Upper\ Bound_i = Q3_i - 1.5 * IQR_i \tag{2}$$

Statistical Testing. The Mann-Whitney U test is used to identify any significant difference in fraud case distribution. The Mann-Whitney U test is statistical testing that does not use the actual values of the class, making the results of inference not sensitive to outliers. In addition, chi-squared and t-tests are also used to analyze feature correlation and determine their relationship with the label and model as a comparison with The Mann-Whitney U test. These statistical tests could help to identify important features for analysis and guide the feature selection and engineering steps.

Binning Techniques. The objective of creating bins is to have equal intervals on a continuous measurement scale. Bins also aim to contain identical samples and be selected using more complex unsupervised methods such as clustering [22]. This study will apply binning techniques: manual binning, frequency-based, clustering-based, and encoding-based binning techniques to address the issue of large categorical values. The choice of binning methods will vary depending on the types of features present in the dataset. The Encoding-based binning techniques use for features with small categorical values or changes with additional information on the categoric value of the features. Frequency-based binning methods examine values with a substantial number of occurrences and create features. The analysis will group features with insignificant occurrences as "others" features. Additionally, the k-means algorithm will cluster the categoric values. The steps include selecting the number of clustering based on elbow methods, calculating the percentage of fraudulent claims per feature, and comparing it to the total number of claims. After that, define the threshold for grouping the percentage based on the level of fraudulent risk. Let $\overline{X}_i$ is the mean of the fraud percentage of each feature, and s_i is the standard deviation of each feature, then the threshold formula is shown in (3).

$$threshold_i = \overline{X}_i + 2s_i \tag{3}$$

Feature Extraction. The analysis will extract additional features to gain insight into the relationship between the features and the target variable for those with low statistical test values. For instance, the "age"

feature is a numerical value. Then, it will derive new features from the existing data by aggregating the numerical values into fewer bins representing the features. Furthermore, this study also combines datasets from multiple sources. It adds external information related to the features and the target variable. The extraction process will include extracting the code for each primary care physician's and specialty hospitals' diagnoses, incorporating new features. Finally, one-hot and Ordinal Encoding will be performed before feeding the features into the model. One-hot will convert categorical features into numerical features. Meanwhile, Ordinal Encoding represents the ordinal numbers. It aims to provide each feature in binary vector format that the algorithm can use. To ensure each feature is standardized in a specific format, the Standard Scaller will use to scale numerical features with zero mean and unit variance to help the model converge faster and provide better results.

2.3. Data Sampling Methods

This study uses sampling data to balance the class. The two main types of data sampling used are: oversampling and undersampling. Oversampling balances classes by adding replicating or generating samples to the minority class; meanwhile, undersampling reduces the majority class size by deleting or selecting the majority class sample.

Undersampling Techniques The simple undersampling technique removes the sample randomly or the Random Undersampling (RUS) method. This technique has shown good performance and has few weaknesses when dealing with large datasets. Some modifications or combinations using this approach showed significant performance and could generate excellent sampled instances [10], [23], [24]. This technique aims to obtain the same class samples by reducing the information in the majority class, which can expedite the training process. However, RUS has a downside – randomly chosen samples lead to the loss of important information. Therefore, some methods, such as Near Miss [25], Edited Nearest Neighbors[26], Neighborhood Cleaning Rule [27], and Instance Hardness Threshold, could handle the issue. However, this study considers only using Instance Hardness Threshold because it is a relatively recent undersampling method compared to others.

Instance Hardness Threshold (IHT) is a method that identifies the "hard" minority class instances (i.e., those most difficult to classify) and discards them from the training set. Knowing which instances are frequently misclassified or hard to classify correctly can improve the algorithm's learning process by reducing the noise in the training data [28]. IHT uses Bayes' theorem to calculate the probability of correctly classifying the data given the input features and labels. Given a dataset D, let x_i be the i-th instance in D, and $IH(x_i)$ be the instance hardness value of x_i, to compute the instance hardness value for all instances in D using (4).

$$IH(x_i) = 1 - P(y_i|x_i, D^c) \tag{4}$$

Where y_i is the actual label of the instance x_i and D^c is the complement of the training set D, a set of all other instances in the D. Then, to remove instances with the highest instance hardness values to obtain the new dataset D' calculate using (5).

$$D' = \{x_i|IH(x_i) < T\} \tag{5}$$

Where T is a threshold value for the maximum allowable instance hardness, in addition, an instance's hardness is a real value between 0 and 1. If the value is close to 0, it will probably be classified correctly. Conversely, if the hardness value is close to 1, the instance will likely be misclassified.

Oversampling Techniques The simplest oversampling method is called Random Oversampling (ROS). It duplicates samples from the minority class and adds them to the training dataset. Since there are many more non-fraud cases than fraud, ROS could be oversampled at high rates to balance the class distribution. Some modification has been combined used ROS and given significant performance [29], [30]. It works by generating additional examples of the minority class. It provides the classifier with more training data to learn from, leading to improved performance in detecting fraud [31]. However, it can cause increases in the size of the dataset and the computational cost and make the classifier tend to the problem of overfitting. In detail, it can make the classifier fit the training data too closely and not generalize the model to unseen new samples.

To handle the problem, one approach that is often better than simply copying existing ones is to create synthetic samples to make the distribution more balanced. It involves interpolating samples that lie together to create new samples [32]. This approach is called SMOTE (Synthetic Minority Over-sampling Technique). SMOTE has been inspiring for most of the new oversampling methods, such as Borderline-SMOTE [33] and ADASYN (*Adaptive Synthetic Sampling Approach*)[34], which does not cover in this study. SMOTE has performed well in addressing imbalanced data with extensive data [15], [19], [29], [35].

SMOTE works by select randomly $x_{i-minority}(i = 1,2,3, \dots, n)$ as the basis for creating new data points. Then, the algorithm selects k nearest neighbors (default 5) by calculating the distance between x_i and other

samples in the same class (points x_{i1} to x_{i4}). Then, the synthetic samples denoted $r_{ij}(i = 1,2,3,\dots,m)$ are generated as an interpolation of the basis points to each kNN using (6).

$$r_{ij} = x_i + \lambda\left(x_{ij} - x_i\right) \tag{6}$$

Where r_{ij} is the synthetic sample generated between x_i and x_{ij}, λ is a random number between 0 and 1, x_{ij} is one of the k nearest neighbors to x_i. The interpolation factor λ controls the amount of interpolation x_i and x_{ij}, and is randomly generated for each synthetic sample.

2.4. Sampling Ratios

When dealing with highly imbalanced data, simply creating a fully balanced class distribution through sampling is not always the best approach. It is because it often results in discarding a significant portion of the original dataset, which can lead to the loss of valuable information and potentially worsen the model's performance. This study will generate three class distributions (majority: and minority), with 50:70:90%. A 90% distribution was selected to analyze the effect of the sampling method when approaching the fully balanced class distribution. Meanwhile, a 50% distribution was chosen to evaluate the model performance, particularly in balanced data. Similarly, a 70% distribution was considered a reasonable representation of imbalanced data. Examining the model's performance on these different distributions gains a better understanding of how the sampling method impacts the model's performance and ability to detect fraud. Table 1 shows the number of instances for each distribution.

Table 1. Total number of instances

Methods	50		70		90	
	0	1	0	1	0	1
RUS	250390	125195	178850	125195	139105	125195
IHT	925830	925830	925830	925830	925830	125195
ROS	1795437	897718	1795437	1256805	1795437	1615893
SMOTE	1795437	897718	1795437	1256805	1795437	1615893

2.5. eXtreme Gradient Boosting (XGBoost)

XGBoost is a machine-learning model based on the boosting concept that integrates multiple weak learners to achieve a strong learner. It is one of the popular models used in the domains such as fraud detection, which addresses the class imbalance that prevents overfitting in training data if not handled properly [36]–[38]. This algorithm also can efficiently handle large-scale datasets, strong predictive performance, and fast training speed [39]. Specifically, XGBoost is an iterative calculation of decision tree classification. At step n, each learner is calculated as (7), where f_k is the basic model of trees, and x_i is an input feature. After that, to measure the performance of each learner L, XGBoost uses a loss function α and the regularization γ term to calculate it. To calculate the perfomance using (8).

$$\hat{y}_i = \sum_{k=1}^{n} f_k(x_i) \tag{7}$$

$$L = \sum_{i} \alpha(\hat{y}_i, y_i) + \sum_{k} \gamma(f_k) \tag{8}$$

The regularization γ calculate using (9), which aims to prevent overfitting, where T is the number of leaves in each learner, σ is the minimal loss, and w is a weight or vector score in leaves.

$$\gamma(f) = \sigma T + \frac{1}{2}\lambda\|w\|^2 \tag{9}$$

XGBoost adds the regularization to reduce the model variance and control model complexity, preserving the fastest possible processing speed with multithreaded parallel computing to speed up the running speed [39]. Furthermore, to optimize the XGBoost algorithm, this study applies hyperparameter tuning for each model with different distributions using Optuna Framework. Here is a brief description of each parameter tuning:

- **Lambda and Alpha**: Higher lambda and alpha values will result in a more regularized model, which avoids or reduces overfitting.
- **Gamma**: Higher gamma values will make the algorithm more conservative, resulting in fewer splits.
- **Min_child_weight**: Higher values will make the model more conservative.

- **Reg_alpha and reg_lambda**: Higher values will result in a more regularized model, which reduces overfitting.
- **Scale_pos_weight**: Higher values will produce a more biased model towards the positive class.
- **Max features**: It is essential to find the right balance between features for the model to learn from and excluding irrelevant or noisy features.
- **Colsample_bytree and Subsample**: Used to reduce overfitting by introducing randomness in the model training. A smaller subsampling will make the model more conservative.
- **Max_depth**: higher values will result in a more complex model which captures more complex interactions between features.

Table 2 shows the hyperparameters, the descriptions, and the range of each value to be tuned.

Table 2. Hyperparameter Testing Value

Hyperparameter	Value	Description
lambda	[1e-3, 10]	L2 regularization term on weights.
alpha	[1e-3, 10]	L1 regulaization term on weights.
gamma	[0, 1, 5]	The minimum loss reduction to make a further partition
min_child_weight	[1, 10]	The minimum sum of instance weight required in a child
reg_alpha	[0, 1]	L1 regularization term on the bias
reg_lambda	[0, 1]	L2 regularization term on the bias
scale_pos_weight	[1, 10]	Control the balance of positive and negative weights
max_features	[auto, sqrt, log2]	Maximum number of features
colsample_bytree	[0.6, 1.0]	Subsample ratio of columns
subsample	[0.6, 1.0]	Subsampleratio of the training instance
max_depth	[9, 11, 13]	Maximum depth of a tree

2.6. The Proposed Model for Evaluation

The chosen metric or model evaluation is critical in an imbalanced task. It is because it measures how well the learning algorithm performs on the test data. In imbalanced data, using just one metric evaluation cannot guarantee the model's ability to generalize the model performance [40]. It could tend to be good in one class rather than another class. Meanwhile, this study aims to balance performance in both classes. Therefore, using multiple metrics is needed to strengthen the analysis. This study also proposed an approach to balance the score of each metric using helped of Optuna, a hyperparameter optimization framework. The following procedure describes how to evaluate the model performance in multiple metrics using Optuna objective function (Table 3).

Table 3. Hyperparameter Testing Value

Proposed Model Evaluation Approach

Input:

- Maximum number of trials (T)
- Hyperparameters θ_i and values v_i, a dictionary containing the hyperparameters to be tuned and the values to be tried, where $i = 1, \dots, n$
- Model and its parameters $M(\theta)$
- Training (X_{train}, y_{train}) and validation (X_{val}, y_{val}) data

Steps for each trial:

1. Define the model: where $M(\theta^{(t)})$ are the hyperparameters for trial t
2. Train the model using training data: $M(\theta^{(t)}) \leftarrow train(M(\theta^{(t)}),\ X_{train}, y_{train})$
3. Evaluate the model using validation data: $\hat{y}val = predict(M(\theta^{(t)}), X_{val})$. Calculate the model's performance on multiple metrics: Precision, recall, F1 score, AUC_{roc} , and $G - mean$.
4. Calculate a weighted score using a formula that combines the metrics: $s^{(t)} = \max(AUC_{roc}, G - mean) \times F1_{macro}$
5. Store the score and hyperparameter for this trial: $(\theta^{(t)},\ s^{(t)})$

Output: the set of hyperparameters that have the highest score by using this formula $\theta^* = argmax_{\theta^{(t)}}\ s^{(t)}$

By taking the maximum value between AUC_{ROC} and $G - mean$ and calculate the product value with $F1_{macro}$ as shown in step 4, the best hyperparameters are the parameters with one metric increasing while the others increase. The Optuna will select the hyperparameters based on the high score $s^{(t)}$. The calculation for each metric has been provided.

First, the F1 measure provides single measures that capture precision and recall properties. F1-measure might be the popular metric for imbalanced classification. Precision is a metric that measures the model's reliability in predicting positive class; used to minimize false positives. Recall measures the model's ability to predict the positive class coverage by minimizing the false negative. This study will use Macro Average performed in Precision, Recall, and F1 measures. Macro Average is a type of averaging that evaluates the classifier's overall performance by treating all classes equally. Equation (10) shows how to calculate the F1 measure with macro average, where n is the amount of class, i is a label, and F1 is the F1 measure that calculates using (11).

$$F1_{macro} = \frac{1}{n}\sum_{i=1}^{n} F1_i \tag{10}$$

$$F1 = 2 \times \frac{Precision_{macro} \times Recall_{macro}}{Precision_{macro} + Recall_{macro}} \tag{11}$$

This study uses a variation of the F-measure called $G - mean$. The metric use geometric means of measurement. It includes information from both classes and aims to balance the classification performances of the majority and minority classes. It calculates TPR or True Positive Rate that summarizes how well the positive class was predicted, and TNR or True Negative Rate summarizes how well the negative class was predicted. $G - mean$ equation is shown (12).

$$TPR = \frac{TP}{TP + FN}$$
$$TNR = \frac{TN}{TN + FP}$$

$$G - mean = \sqrt{TPR \times TNR} \tag{12}$$

As a comparison with $G - mean$, AUC_{ROC} will add to the model evaluation. It summarizes performance over the range of true positive rates (TPRs) and false positive rates (FPRs). The true positive rate is the recall or sensitivity. After that, the metric will calculate the AUC to provide a single score to summarize the curve plot, which is helpful for model comparison. To calculate the AUC_{ROC} using (13).

$$TPRate = \frac{TP}{TP + FN}$$

:

$$FPRate = \frac{FP}{FP + TN}$$

$$AUC_{ROC} = \int_0^1 TPRate(FPRate^{-1}(x))\, dx \tag{13}$$

3. RESULTS AND DISCUSSION

This section provides the result and the analysis of the experiment. The cleaned data were trained without performing the sampling method and used the default XGBoost parameters. It is used as a baseline model and noted as `Baseline` in the Table 4, Table 5, and Table 6. The experiment shows that the model's performance significantly improved when using sampling methods and hyperparameter tuning compared to the baseline model. It highlights the impact of the sampling method on the model's ability to classify the minority class correctly. However, upon examining each metric's score, the scores of both classes were not significantly different. It suggests the model can perform reasonably well in detecting both classes despite the low score and imbalanced class distribution. In addition, a difference in the precision and recall score indicates that the model misses some positive instances, a common issue in imbalanced class distributions.

Based on the scores in Table 4, the model with RUS and ROS techniques performed better across all metrics. The AUC_{ROC} score of (0.79 and 0.81) and the $G - mean$ score of 0.76 indicates that these models can effectively distinguish between the positive (fraud) and negative (non-fraud) classes while maintaining performance in both classes. The high $F1_{macro}$ also suggests that these models balance precision and recall,

correctly identifying and classifying positive instances while minimizing false positives and negatives. On the other hand, the model using SMOTE had higher precision and recall scores than the others but had lower scores in AUC_{ROC} and $G-mean$. While a high precision and recall score indicates good performance in correctly identifying positive instances, it may not accurately distinguish between positive and negative instances, as apparent in decreased AUC_{ROC} and $G-mean$ scores. It means that the model prioritizes the performance of the positive class over the negative class, leading to misclassifying negative instances. Furthermore, the model with IHT could not identify positive instances in the dataset. The $F1_{macro}$ of 0.73 suggests the model underperformed in balance precision and recall compared to others. It misclassified many instances as false positives or negatives. It indicated that the model was biased toward one class's performance, leading to suboptimal results in detecting fraud.

Table 4. 50% class distribution

Method	AUC	GM	F1	PR	RC
Baseline	0.64	0.67	0.67	0.71	0.65
RUS	**0.79**	**0.76**	**0.75**	**0.73**	**0.78**
IHT	0.77	0.74	0.73	0.71	0.76
ROS	**0.81**	**0.76**	**0.76**	**0.73**	**0.78**
SMOTE	0.75	0.73	0.76	0.77	0.76

Significant differences appear in the results when the distribution is 70%. According to Table 5, the models with RUS and ROS performed worse than SMOTE and IHT in terms of $F1_{macro}$, precision, and recall scores. It indicates that the models did not effectively balance precision and recall and were biased towards the majority class, which misclassified minority-class instances as false negatives.

Table 5. 70% class distribution

Method	AUC	GM	F1	PR	RC
Baseline	0.64	0.67	0.67	0.71	0.65
RUS	0.80	0.78	0.71	0.67	0.79
IHT	**0.76**	**0.72**	**0.75**	**0.74**	**0.75**
ROS	0.82	0.78	0.72	0.68	0.79
SMOTE	**0.76**	**0.74**	**0.74**	**0.73**	**0.76**

Although they have high AUC_{ROC} and $G-mean$ scores, they may not be optimal as they are not effectively identifying positive instances and may have high false rates (FNR). On the other hand, it appears that the models with SMOTE and IHT performed well across all evaluation metrics with no significant difference between them. It suggests that the models consistently perform across different measures and generalize well to new unseen data.

Meanwhile, when the distribution was increased to 90%, as shown in Table 6, most methods underperformed, especially in the model with RUS and ROS. While the AUC_{ROC} score of 0.81 and the $G-mean$ score of 0.79 may seem high, but it is essential to note that these scores do not guarantee a good classifier. The RUS and ROS techniques prioritized distinguishing between positive and negative instances. However, they underperformed in correctly identifying positive instances, as shown by $F1_{macro}$, and precision scores were less than 0.7. The significant difference in recall score indicates that the models can avoid false positives (FP) but fail in avoid false negatives, which may lead to a high rate of missed fraud cases.

Table 6. 90% class distribution

Method	AUC	GM	F1	PR	RC
None	0.64	0.67	0.67	0.71	0.65
RUS	0.81	0.79	0.67	0.64	0.8
IHT	**0.76**	**0.72**	**0.74**	**0.75**	**0.75**
ROS	0.81	0.79	0.69	0.65	0.8
SMOTE	0.77	0.75	0.72	0.69	0.77

In addition, the Precision-Recall Area Under the Curve (AUC_{PR}) is presented in Fig. 4. AUC_{PR} evaluates the model's ability to identify positive classes by balancing graphical precision against recall at various threshold values. It provides a holistic measure of the model ability to correctly identify positive instances, particularly in binary classification tasks where the positive class is rare or imbalanced. The area under this precision-recall curve aggregates the model's performance across different trade-offs between precision and recall. A higher AUC_{PR} value indicates better performance, with a perfect classifier having a AUC_{PR} 1.0. Based

on the figure, all methods performed better than random chance, with scores above 0.5 demonstrating the effectiveness of the sampling method in addressing the imbalanced data. The model with SMOTE in 50% and 70% class distribution achieved higher scores, indicating that these methods effectively balanced precision and recall in improve the model's ability to detect positive classes. Further, the model with ROS at 50% distribution achieved the highest score among all methods.

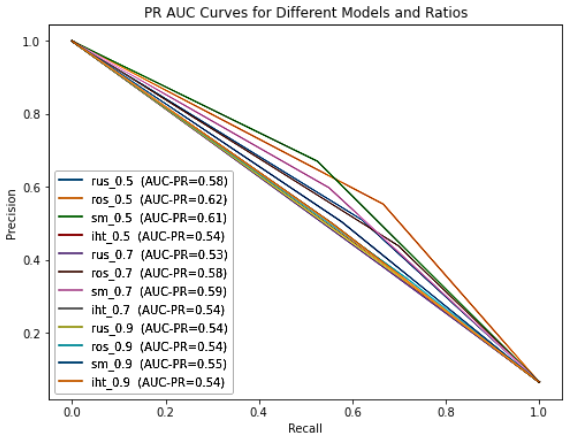

Fig. 4. Precision-Recall Area Under the Curve

The difference score in each class distribution provides insight into how much the sampling method impacts the model's performance. When the class distribution is more balanced (i.e., 50%), non-heuristic methods like RUS and ROS are more effective at improving model performance. It is because they did not introduce any additional noise or overfitting into the dataset. They can effectively balance the class distribution by removing or adding samples without changing the data. In contrast, heuristic methods such as SMOTE and IHT underperformed as they introduced noise or overfit in the model. If the synthetic samples are too similar to the existing minority class samples, it could lead to the over-representation of specific regions in the feature space. IHT methods also focus on removing hard-to-classify samples, which can remove helpful information and guide to underfitting.

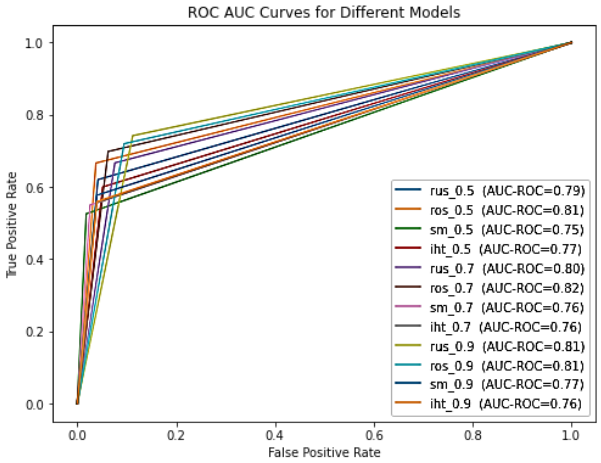

Fig. 5. Receiver Operating Characteristic Area Under the Curve

Fig. 5 demonstrates the effectiveness of RUS and ROS methods, particularly in AUC_{ROC} metrics. It showed scores close to one, which is the perfect value in AUC_{ROC} standard score. RUS and ROS in 70% seem to perform better than other methods and give a confidence score. Moreover, this situation can be wrong or make the score too optimistic, resulting in a bad performance in real-world cases. In fact, the RUS and ROS underperformed because the minority class was even more underrepresented, as shown in inconsistent scores across metrics in Table 5. It is more challenging for RUS and ROS to generate representative samples that reflect the actual distribution of the minority class. It was because there was limited data to select or remove samples randomly. Meanwhile, SMOTE and IHT were more effective and achieved the desired score. Those

methods work by generating synthetic samples, which improves the representation of the minority class instead of duplication the data. It creates more diverse and balanced data by interpolating between existing minority samples. Similarly, IHT can identify the minority sample that is difficult to classify and remove from the dataset. It can help reduce the noise and improve the model's overall performance, even if it does not directly balance the class distribution.

When the distribution increased to 90%, most methods performed poorly, likely due to the challenge of generating representative samples of the minority class. Fig. 6 shows the $F1_{macro}$ scores for the models with RUS, ROS, and SMOTE underperformed compared to others. The scores decreased and stacked to around 0.6 scores. Each sampling methods that reduced more data from the majority class or added more data to the minority class were less effective. They have not provided enough data for the model to learn effectively or introduced too much noise or overfitting. Interestingly, IHT performed relatively well, as it focused on removing samples that were hard to classify. The model can learn more effectively from the remaining data and tend to be a more representative sample set for the model to learn.

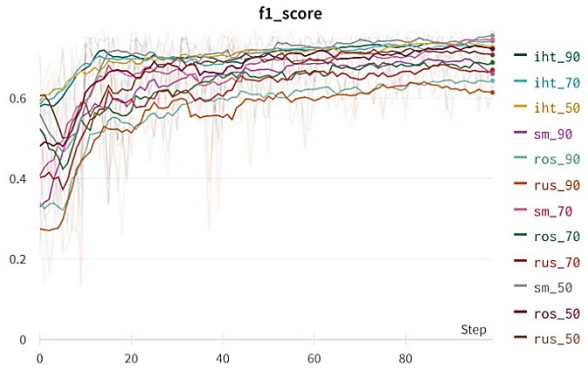

Fig. 6. F1 Score Trend

The model with SMOTE in 70% has demonstrated superior performance, particularly in scoring consistency across each metric. It indicates that the model can effectively classify both the positive (majority) and negative (minority classes), resulting in balanced and high scores across different evaluation metrics. This consistency in performance is noteworthy, as it ensures that the model is not overly optimistic by relying on a single metric but instead provides a comprehensive analysis of its ability to detect fraud in the imbalanced dataset. This approach of evaluating multiple metrics and observing consistent scores is different from previous studies, such as in [15], [17]–[19], which have shown inconsistent scores across different metrics. Considering multiple metrics strengthens the analysis of the model's performance. It provides a more comprehensive understanding of its ability to detect fraud accurately, particularly in the imbalanced datasets that are common in healthcare fraud detection. It is crucial to accurately predict fraud and non-fraud cases in real-world healthcare fraud detection scenarios. However, it should be noted that using a single classifier may not always result in the highest scores. The hyperparameter range values must also be tuned based on the specific dataset characteristics. Future research could explore using a multiclassifier, which could improve the model's performance even further.

In addition, the analysis of computation time and features that are significantly impacted will also be analyzed. As shown in Fig. 7, the ROS and RUS are much faster than IHT and SMOTE. Meanwhile, in Figure 8, the feature importance analysis suggests that the *num_biaya* and the *num_total_diag_sekunder* features were the most informative and influenced in detecting fraud claims in the dataset. The *num_biaya,* or the cost features, represent the cost claimed by the healthcare service and are the most informative in detecting fraudulent claims in our dataset. Fraudulent claims involve overcharging for medical services or claiming costs for services not provided, as found in the EDA process.

Meanwhile, the *num_total_diag_sekunder* feature represents the number of patients' specialty hospital diagnoses, which is also important because fraudulent claims involve exaggerating the severity or complexity of a patient's condition to justify unnecessary treatments or procedures. In addition, other features such as the total number of days each patient has, code-based group, gender, location, primary care physician's diagnosis code, and severity level also influence the model's performance. Based on the feature importance (Fig. 8), it suggests that the preprocessing data improves model performance.

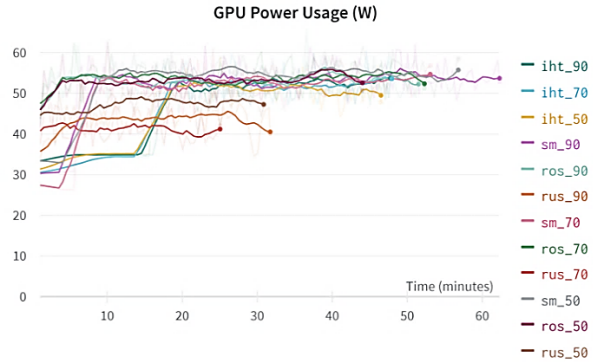

Fig. 7. Computation Time

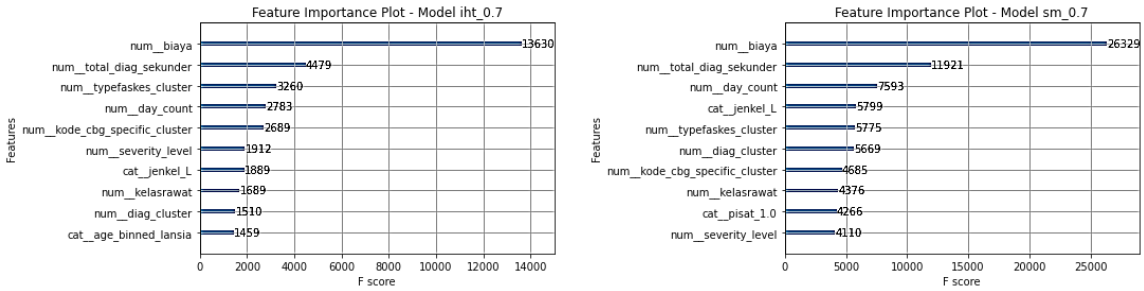

Fig. 8. Feature importance Plot

4. CONCLUSION

This study evaluates the effectiveness of four data sampling approaches in addressing the impact of severe class imbalance. This study also identifies the key features, proposes a model evaluation approach, and provides insight into how much multiple metrics affect the model performance analysis. The results revealed that in the 50% class distribution, the ROS and RUS methods outperformed other methods' overall scores for different metrics, with ROS having the highest scores for AUC_{ROC} score of 0.81, $G-mean$ score of 0.76, $F1_{macro}$ score of 0.76, Precision score of 0.73, and Recall score of 0.78. Meanwhile, in the 70% distribution, the IHT and SMOTE methods outperformed other methods, with SMOTE showing the highest score for AUC_{ROC} score of 0.76, $G-mean$ score of 0.74, $F1_{macro}$ score of 0.74, Precision score of 0.73, and Recall score of 0.76. Additionally, in the 90% class distribution, the IHT method outperformed other methods' overall scores for AUC_{ROC} score of 0.76, $G-mean$ score of 0.72, $F1_{macro}$ score of 0.74, Precision score of 0.75, and Recall score of 0.75.

Based on the overall scores in each distribution, the 70% distribution was more robust, particularly with the SMOTE method. However, they had longer computation times due to the data sampling techniques used. These models consistently performed well across all evaluation metrics, indicating their ability to generalize to new unseen data in both the minority and majority classes. It showed that costs, diagnosis codes, type of healthcare service, gender, and severity level of diseases, were necessary for accurate fraud predictions. These findings could be valuable for healthcare providers, such as BPJS, to make informed decisions with lower risks. A well-performing fraud detection model ensures the accurate classification of fraud and non-fraud cases. The findings also can be used by healthcare insurance providers to develop more effective fraud detection and prevention strategies.

However, some limitations offer opportunities for future research. The reliance on a single real-world dataset may have biases and limitations. Additionally, the study only focuses on single classifiers and traditional machine learning. Therefore, future research could explore multiple datasets from different healthcare insurance providers to validate the findings and enhance the generalizability of the results. Future research can also explore various classifiers and advanced techniques, such as deep learning or ensemble methods for healthcare fraud detection. Lastly, incorporating domain-specific knowledge or external data sources could be promising future research to enhance fraud detection accuracy . Furthermore, the proposed model evaluation approach and insights into the effectiveness of different data sampling can guide future research in developing more robust and accurate fraud detection models for imbalanced healthcare insurance datasets.

Acknowledgments

The authors thank the Indonesia Social Security Agency of Health (BPJS) for providing the data sources for this research project.

REFERENCES

[1] A. Y. B. R. Thaifur, M. A. Maidin, A. I. Sidin, and A. Razak, "How to detect healthcare fraud? 'A systematic review,'" *Gac. Sanit.*, vol. 35, pp. S441–S449, 2021, https://doi.org/10.1016/j.gaceta.2021.07.022.
[2] X. Zhu *et al.*, "Intelligent financial fraud detection practices in post-pandemic era," *Innovation*, vol. 2, no. 4, p. 100176, 2021, https://doi.org/10.1016/j.xinn.2021.100176,
[3] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances," *Expert Syst. Appl.*, vol. 193, p. 116429, 2022, https://doi.org/10.1016/j.eswa.2021.116429.
[4] C. Sun, Q. Li, H. Li, Y. Shi, S. Zhang, and W. Guo, "Patient Cluster Divergence Based Healthcare Insurance Fraudster Detection," *IEEE Access*, vol. 7, pp. 14162–14170, 2019, https://doi.org/10.1109/ACCESS.2018.2886680..
[5] S. Zhou, J. He, H. Yang, D. Chen, and R. Zhang, "Big Data-Driven Abnormal Behavior Detection in Healthcare Based on Association Rules," *IEEE Access*, vol. 8, pp. 129002–129011, 2020, https://doi.org/10.1109/ACCESS.2020.3009006.
[6] C. F. Tsai, W. C. Lin, Y. H. Hu, and G. T. Yao, "Under-sampling class imbalanced datasets by combining clustering analysis and instance selection," *Inf. Sci. (Ny).*, vol. 477, pp. 47–54, 2019, https://doi.org/10.1016/j.ins.2018.10.029.
[7] L. Wang, Z. Zhang, X. Zhang, X. Zhou, P. Wang, and Y. Zheng, "A Deep-forest based approach for detecting fraudulent online transaction," *In Advances in computers,* vol. 120, pp. 1-38, 2021, https://doi.org/10.1016/bs.adcom.2020.10.001.
[8] K. G. Al-Hashedi and P. Magalingam, "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019," *Comput. Sci. Rev.*, vol. 40, 2021, https://doi.org/10.1016/j.cosrev.2021.100402.
[9] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Inf. Sci. (Ny).*, vol. 513, pp. 429–441, 2020, https://doi.org/10.1016/j.ins.2019.11.004.
[10] D. Devi, S. K. Biswas, and B. Purkayastha, "A Review on Solution to Class Imbalance Problem: Undersampling Approaches," in *2020 International Conference on Computational Performance Evaluation, ComPE 2020*, pp. 626–631, 2020, https://doi.org/10.1109/ComPE49325.2020.9200087.
[11] S. Zhou, Y. Gu, H. Yu, X. Yang, and S. Gao, "RUE: A Robust Personalized Cost Assignment Strategy for Class Imbalance Cost-sensitive Learning," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 4, pp. 36–49, 2023, https://doi.org/10.1016/j.jksuci.2023.03.001.
[12] B. Mirzaei, B. Nikpour, and H. Nezamabadi-Pour, "An under-sampling technique for imbalanced data classification based on DBSCAN algorithm," *8th Iran. Jt. Congr. Fuzzy Intell. Syst. CFIS 2020*, pp. 21–26, 2020, https://doi.org/10.1109/CFIS49607.2020.9238718.
[13] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results," in *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, pp. 243–248, 2020, https://doi.org/10.1109/ICICS49469.2020.239556.
[14] M. Y. Arafat, S. Hoque, S. Xu, and D. M. Farid, "An under-sampling method with support vectors in multi-class imbalanced data classification," *2019 13th Int. Conf. Software, Knowledge, Inf. Manag. Appl. Ski. 2019*, pp. 1–6, 2019, https://doi.org/10.1109/SKIMA47702.2019.8982391.
[15] T. G.S., Y. Hariprasad, S. S. Iyengar, N. R. Sunitha, P. Badrinath, and S. Chennupati, "An extension of Synthetic Minority Oversampling Technique based on Kalman filter for imbalanced datasets," *Mach. Learn. with Appl.*, vol. 8, p. 100267, 2022, https://doi.org/10.1016/j.mlwa.2022.100267.
[16] H. Zhu, G. Liu, M. Zhou, Y. Xie, and Q. Kang, "A Noisy-sample-removed Under-sampling Scheme for Imbalanced Classification of Public Datasets," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 624–629, 2020, https://doi.org/10.1016/j.mlwa.2022.100267.
[17] S. K. Shamitha and V. Ilango, "A time-efficient model for detecting fraudulent health insurance claims using Artificial neural networks," *2020 Int. Conf. Syst. Comput. Autom. Networking, ICSCAN 2020*, pp. 1-6, 2020, https://doi.org/10.1109/ICSCAN49426.2020.9262298.
[18] C. Tian, L. Zhou, S. Zhang, and Y. Zhao, "A New Majority Weighted Minority Oversampling Technique for Classification of Imbalanced Datasets," *Proc. - 2020 Int. Conf. Big Data, Artif. Intell. Internet Things Eng. ICBAIE 2020*, pp. 154–157, 2020, https://doi.org/10.1109/ICBAIE49996.2020.00039.
[19] P. Mrozek, J. Panneerselvam, and O. Bagdasar, "Efficient resampling for fraud detection during anonymised credit card transactions with unbalanced datasets," *Proc. - 2020 IEEE/ACM 13th Int. Conf. Util. Cloud Comput. UCC 2020*, pp. 426–433, 2020, https://doi.org/10.1109/UCC48980.2020.00067.
[20] J. Tian, Y. Ren, and X. Cheng, "Stratified feature sampling for semi-supervised ensemble clustering," *IEEE Access*, vol. 7, pp. 128669–128675, 2019, https://doi.org/10.1109/ACCESS.2019.2939581.
[21] S. A. Khan and S. S. Velan, "Application of Exploratory Data Analysis to Generate Inferences on the Occurrence of Breast Cancer using a Sample Dataset," *Proc. Int. Conf. Intell. Eng. Manag. ICIEM 2020*, pp. 449–454, 2020, https://doi.org/10.1109/ICIEM48762.2020.9160290.
[22] A. C. Deckert and E. Kummerfeld, "Investigating the effect of binning on causal discovery," in *Proceedings - 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019*, pp. 2574–2581, 2019, https://doi.org/10.1109/BIBM47256.2019.8983336.
[23] B. Liu and G. Tsoumakas, "Dealing with class imbalance in classifier chains via random undersampling,"

Knowledge-Based Syst., vol. 192, p. 105292, 2020, https://doi.org/10.1016/j.knosys.2019.105292.
[24] M. Bach, "New Undersampling Method Based on the kNN Approach," *Procedia Comput. Sci.*, vol. 207, pp. 3397–3406, 2022, https://doi.org/10.1016/j.procs.2022.09.399.
[25] A. Tanimoto, S. Yamada, T. Takenouchi, M. Sugiyama, and H. Kashima, "Improving imbalanced classification using near-miss instances," *Expert Syst. Appl.*, vol. 201, p. 117130, 2022, https://doi.org/10.1016/j.eswa.2022.117130.
[26] Y. Zhu, C. Jia, F. Li, and J. Song, "Inspector: a lysine succinylation predictor based on edited nearest-neighbor undersampling and adaptive synthetic oversampling," *Anal. Biochem.*, vol. 593, p. 113592, 2020, https://doi.org/10.1109/ICOMITEE.2019.8921159.
[27] K. Agustianto and P. Destarianto, "Imbalance Data Handling using Neighborhood Cleaning Rule (NCL) Sampling Method for Precision Student Modeling," *Proc. - 2019 Int. Conf. Comput. Sci. Inf. Technol. Electr. Eng. ICOMITEE 2019*, vol. 1, pp. 86–89, 2019, https://doi.org/10.1109/ICOMITEE.2019.8921159.
[28] C. A. Dantas, R. D. O. Nunes, A. M. P. Canuto, and J. C. Xavier, "Instance hardness as a decision criterion on dynamic ensemble structure," *Proc. - 2019 Brazilian Conf. Intell. Syst. BRACIS 2019*, pp. 108–113, 2019, https://doi.org/10.1109/BRACIS.2019.00028.
[29] L. K. Xin and N. binti A. Rashid, "Prediction of depression among women using random oversampling and random forest," in *2021 International Conference of Women in Data Science at Taif University, WiDSTaif 2021*, pp. 1-5, 2021, https://doi.org/10.1109/WiDSTaif52235.2021.9430215.
[30] Y. Pang, Z. Chen, L. Peng, K. Ma, C. Zhao, and K. Ji, "A signature-based assistant random oversampling method for malware detection," *Proc. - 2019 18th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. IEEE Int. Conf. Big Data Sci. Eng. Trust. 2019*, pp. 256–263, 2019, https://doi.org/10.1109/TrustCom/BigDataSE.2019.00042.
[31] X. W. Ding, Z. T. Liu, D. Y. Li, Y. He, and M. Wu, "Electroencephalogram Emotion Recognition Based on Dispersion Entropy Feature Extraction Using Random Oversampling Imbalanced Data Processing," *IEEE Trans. Cogn. Dev. Syst.*, vol. 14, no. 3, pp. 882–891, 2022, https://doi.org/10.1109/TCDS.2021.3074811.
[32] Asniar, N. U. Maulidevi, and K. Surendro, "SMOTE-LOF for noise identification in imbalanced data classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3413–3423, 2022, https://doi.org/10.1016/j.jksuci.2021.01.014.
[33] Y. Chen, R. Chang, and J. Guo, "Effects of Data Augmentation Method Borderline-SMOTE on Emotion Recognition of EEG Signals Based on Convolutional Neural Network," *IEEE Access*, vol. 9, pp. 47491–47502, 2021, https://doi.org/10.1109/ACCESS.2021.3068316.
[34] S. Hasmita, F. Nhita, D. Saepudin, and A. Aditsania, "Chili commodity price forecasting in bandung regency using the adaptive synthetic sampling (ADASYN) and K-Nearest neighbor (KNN) algorithms," *2019 Int. Conf. Inf. Commun. Technol. ICOIACT 2019*, pp. 434–438, 2019, https://doi.org/10.1109/ICOIACT46704.2019.8938525.
[35] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," *Appl. Soft Comput. J.*, vol. 83, p. 105662, 2019, https://doi.org/10.1016/j.asoc.2019.105662.
[36] C. V. Priscilla and D. P. Prabha, "Influence of optimizing xgboost to handle class imbalance in credit card fraud detection," in *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, pp. 1309–1315, 2020, https://doi.org/10.1109/ICSSIT48917.2020.9214206.
[37] S. He, B. Li, H. Peng, J. Xin, and E. Zhang, "An Effective Cost-Sensitive XGBoost Method for Malicious URLs Detection in Imbalanced Dataset," *IEEE Access*, vol. 9, pp. 93089–93096, 2021, https://doi.org/10.1109/ACCESS.2021.3093094.
[38] P. Gupta, A. Varshney, M. R. Khan, R. Ahmed, M. Shuaib, and S. Alam, "Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques," *Procedia Comput. Sci.*, vol. 218, pp. 2575–2584, 2023, https://doi.org/10.1016/j.procs.2023.01.231.
[39] Y. Zhang, J. Tong, Z. Wang, and F. Gao, "Customer Transaction Fraud Detection Using Xgboost Model," *Proc. - 2020 Int. Conf. Comput. Eng. Appl. ICCEA 2020*, pp. 554–558, 2020, https://doi.org/10.1109/ICCEA50009.2020.00122.
[40] 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," *Pattern Recognit.*, vol. 91, pp. 216–231, 2019, https://doi.org/10.1016/j.patcog.2019.02.023.

BIOGRAPHY OF AUTHORS

Joanito Agili Lopo is a Satya Wacana Christian University student pursuing a bachelor's degree in Information System. His research interest is in Data Science and Natural Language Processing, and he is currently working on Multilingual Machine Translation for Low Resource Languages. Email: 682019013@student.uksw.edu. Orcid: 0009-0001-3183-7132.

Kristoko Dwi Hartomo earned his study in the Doctorate Program of Computer Science in Science Faculty of Gadjah Mada University Yogyakarta in 2019. He has been active in doing research since 2008 until now on Geography Information System. He published his papers in international journals. Moreover, he has had five copyrights and has written some reference books on computer science. Email: kristoko@uksw.edu. Orcid: 0000-0003-0237-851X.