Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 9, No. 2, June 2023, pp. 396-410
ISSN: 2338-3070, DOI: 10.26555/jiteki.v9i2.26155

Application of SMOTE to Handle Imbalance Class in Deposit Classification Using the Extreme Gradient Boosting Algorithm

Dina Arifah, Triando Hamonangan Saragih, Dwi Kartini, Muliadi, Muhammad Itqan Mazdadi
Computer Science Lambung Mangkurat University, Jalan A.Yani Km 36, Banjarbaru 70714, Indonesia

ARTICLE INFO

Article history:

Received April 02, 2023
Revised May 30, 2023
Published June 03, 2023

Keywords:

Classification;
SMOTE;
Imbalance Class;
Deposit;
Extreme Gradient Boosting;
Bank Marketing;
Customer Identification

ABSTRACT

Deposits became one of the main products and funding sources for banks and increasing deposit marketing is very important. However, telemarketing as a form of deposit marketing is less effective and efficient as it requires calling every customer for deposit offers. Therefore, the identification of potential deposit customers was necessary so that telemarketing became more effective and efficient by targeting the right customers, thus improving bank marketing performance with the ultimate goal of increasing sources of funding for banks. To identify customers, data mining is used with the UCI Bank Marketing Dataset from a Portuguese banking institution. This dataset consists of 45,211 records with 17 attributes. The classification algorithm used is Extreme Gradient Boosting (XGBoost) which is suitable for large data. The data used has a high-class imbalance, with "yes" and "no" percentages of 11.7% and 88.3%, respectively. Therefore, the proposed solution in the research, which focused on addressing the Imbalance Class in the Bank marketing dataset, was to use Synthetic Minority Over-sampling (SMOTE) and the XGBoost method. The result of the XGBoost study was an accuracy of 0.91016, precision of 0.79476, recall of 0.72928, F1-Score of 0.56198, ROC Area of 0.93831, and AUCPR of 0.63886. After SMOTE was applied, the accuracy was 0.91072, the precision was 0.78883, the recall was 0.75588, F1-Score was 0.59153, ROC Area was 0.93723, and AUCPR was 0.63733. The results showed that XGBoost and SMOTE could outperform other algorithms such as K-Nearest Neighbor, Random Forest, Logistic Regression, Artificial Neural Network, Naïve Bayes, and Support Vector Machine in terms of accuracy. This study contributes to the development of effective machine learning models that can be used as a support system for information technology experts in the finance and banking industries to identify potential customers interested in subscribing to deposits and increasing bank funding sources.

Corresponding Author:

Triando Hamonangan Saragih, Computer Science Lambung Mangkurat University, Banjarbaru 70714, Indonesia
Email: triando.saragih@ulm.ac.id

1. INTRODUCTION

The intense competition that occurs in the business world has a major impact on every industry, including the banking industry [1]. The Bank makes marketing an effective means of increasing business. The influence of communication progress has had an impact on marketing changes that were previously carried out by meeting directly with customers, then changing to marketing via telephone communication. Communication using the telephone uses a low cost when compared to marketing which requires meeting in person. This marketing is also known as telemarketing. Product marketing that is commonly offered by banks is time deposits. Deposits are the main source of funds and have the characteristics of strong stability at low costs [2]. Marketing done by telemarketing turned out to be less effective because it required marketing staff to make phone calls to every customer and the lack of response from customers [3] which wasted a lot of time. Based

on the problems, a classification is needed to identify customers with the appropriate criteria for taking deposits. So that telemarketing is carried out more effectively in terms of time which can be useful in improving deposit marketing performance because it is very important to improve deposit marketing which is the main product as well as a source of funding for banks.

The banking industry increases the effectiveness of marketing employees for marketing via telephone calls by applying data mining because currently data mining is used in various industries, including finance and banking [4], [5]. There are various effective methods in data mining that can be used for easier decision-making [1]. In research conducted by Valarmathi *et al.* to compare the accuracy of classification algorithms such as Naïve Bayes, J48, KNN, and Bayesnet using bank marketing data. J48 obtained the best accuracy results of 91.2% after reducing dimensions [6]. In another study by Verma with the same data using SMOTE to overcome data imbalance with several algorithms such as Decision Tree (C 4.5), Naïve Bayes, Multilayer Neural Network, SVM, Logistic Regression and Random Forest. The best results were obtained by Random Forest with SMOTE [7]. Then research using the Uncertain Decision Tree by Yang and Chen with a collection of bank marketing data obtained a test accuracy of 93.5% [8]. In addition, another study was conducted by Zeinulla *et al.* with the UCI Bank Marketing dataset. Based on the research results, the best model for predicting the effectiveness of bank telemarketing is Random Forests with an accuracy of 90.884% [9].

The most popular technology used for research is Machine Learning [10]. The superior Machine Learning algorithm for classification problems is the XGBoost algorithm [11]. XGBoost is a form of improving all basic classifiers such as Decision Tree, K-Nearest Neighbor, Support Vector Machine, Logistic Regression, and other algorithms [12] that were developed from the Gradient enhancement technique to produce a robust classification [13]. Research on developing a system for automatic diagnosis in classifying tumors conducted by Sinha *et al.* showed that XGBoost produces better accuracy than the Support Vector Machine, K-Nearest Neighbor, Random Forest, and Adaboost Classifier algorithms with an accuracy value of 98% [14]. Another study by Prabha *et al.* using the XGBoost algorithm based on Hybrid FS in the detection of diabetes mellitus obtained the highest accuracy when compared to the K-Nearest Neighbors, Support Vector Machine, and Random Forest algorithms, namely 99.93% [15]. Then in research conducted by Li and Zhang for the diagnosis of orthopedic diseases, it was shown that the XGBoost algorithm has a high level of accuracy, calculation speed, performance, and memory when compared to other algorithms such as the Random Forest algorithm and the Associated Classification algorithm [10], [13]. XGBoost has great potential for widespread use in real-world binary classification tasks which generally involve large amounts of data and unbalanced labels [16]. XGBoost's highly scalable or parallelizable capabilities, fast execution, and superior performance over other algorithms are some of its key benefits [14].

The performance of the Machine Learning algorithm will not get maximum results if there is an imbalance in the data [17]. If using traditional classifiers on unbalanced data, the results obtained may be biased towards the majority class. This can lead to poor classifier performance. The algorithm that is often used to improve classifier performance on unbalanced data by creating new minority samples is SMOTE [18]. The use of SMOTE can significantly improve classification performance, as in a study conducted by Mohammed *et al.* using six algorithms and the algorithm with the highest performance was Decision Tree [19]. In another study conducted by Du, it was shown that improved SMOTE with XGBoost could obtain the best accuracy compared to the other four methods, namely RF, SVM, BP, and KNN in the analysis of student psychological stress [20]. Based on some of the previous explanations, it is evident that SMOTE is an algorithm that is often used to improve classifier performance [18].

The existing problem is the ineffectiveness of telemarketing by banks because they have to contact each customer to offer a deposit. So it is necessary to classify customers who have the potential to take deposits so that marketing can be carried out on target. Based on some of the problems previously described, this study focuses on overcoming class imbalances in the Bank Marketing data set using the XGBoost algorithm to identify customers who have the potential to subscribe to deposits and SMOTE to overcome the problem of data imbalance in the Bank Marketing dataset. This research contributes to the development of an effective machine learning model that can be used as a support system for informatics experts in finance and banking in identifying potential customers who are interested in subscribing to deposits. So as to increase the effectiveness of banking telemarketing with the ultimate goal of increasing sources of funding for banks.

2. RESEARCH METHOD

The main objective of this study is to identify customers who have the potential to take time deposits based on existing data according to predetermined criteria with the ultimate goal of increasing sources of funding for banks. If customer identification is successful, the bank's telemarketing can be more effective and on target. The proposed system for identifying time deposit customers is shown in Fig. 1.

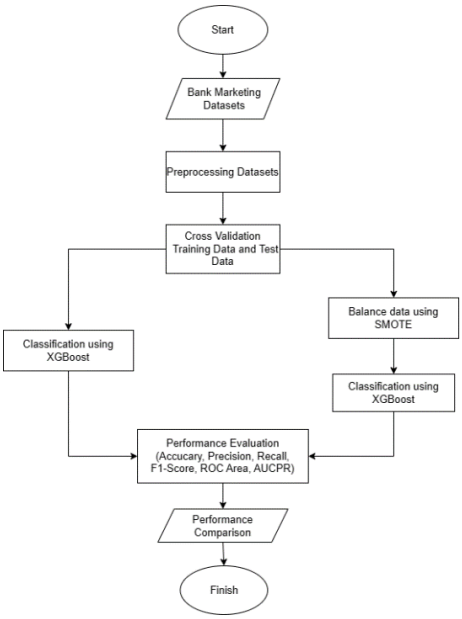

Fig. 1. The Proposed system architecture

2.1. Data Collection

This study used public data on bank marketing taken from the UCI Repository Bank Marketing Dataset obtained from banking institutions in Portugal to promote bank deposit products from May 2008 to November 2010 which was published in 2012 [21]. The data used relates to marketing efforts via direct telephone calls made by a banking institution in Portugal to promote bank deposit products [7]. Data Bank Marketing consists of 45211 records with 16 attributes and one class label y. In addition, Bank Marketing data also consists of input and output variables, with each variable having its attributes [17]. For attribute specifications along with descriptions of each variable can be seen in Table 1. The number of the majority class in the data is 39922 (no), while the number of minority data classes is 5289 (yes) [7], with the percentage comparison between (yes) and (no) classes being 11.7% and 88.3%. This shows that the large data set used has an imbalance of data classes [17].

Table 1. Attributes and Description of the Bank Marketing Dataset

Variable	Attributes	Type	Description
Input	age	Numeric	Age at the contact date (≥18)
	job	Categorical	Type of job ("admin", "unknown", "unemployed", "management", "housemaid", "entrepreneur", "student", "blue-collar", "self-employed", "retired", "technician", "services")
	marital	Categorical	Marital status ("married", "divorced", "single"; note: "divorced" means divorced or widowed)
	education	Categorical	Education (“unknown”, “secondary”, “primary”, “tertiary”)
	default	Binary	Has credit in default? (“yes”, “no”)
	balance	Numeric	Average yearly balance, in euros
	housing	Binary	Has a housing loan? (“yes”, “no”)
	loan	Binary	Has a personal loan? (“yes”, “no”)
	contact	Categorical	Contact communication type ("unknown", "telephone", "cellular")
	day	Numeric	Last contact day of the month
	month	Categorical	Last contact month of the year ("jan", "feb", "mar", ..., "nov", "dec")
	duration	Numeric	Last contact duration, in seconds
	campaign	Numeric	Number of contacts performed during this campaign and for this client (includes the last contact)
	pdays	Numeric	Number of days that passed by after the client was last contacted from a previous campaign (-1 means the client was not previously contacted)
	previous	Numeric	Number of contacts performed before this campaign and for this client
	poutcome	Categorical	The outcome of the previous marketing campaign ("unknown", "other", "failure", "success")
Output	y(accepted)	Binary	Has the client subscribed to a term deposit? (“yes”, “no”)

2.2. Data Preprocessing

Preprocessing is a method used to clean raw data and prepare it for input into the algorithm. The goal is to get a clean data set ready to be processed by Machine Learning algorithms [14]. In general, the collected data cannot be directly processed using Machine Learning because it contains several problems such as missing values, class imbalance, data inconsistency, and so on [22]. The data used in this study is quite good because after checking, no missing values or duplicate data were found. The preprocessing carried out in this study is changing categorical variables into dummy variables using the label encoder method for the number of categories of two namely "yes" and "no" and the one-hot encoding method for features that have more than two categories. One-hot encoding is a popular method used to describe vectors of categorical variables needed in statistical models [23].

2.3. Cross Validation

Cross Validation is a technique used to evaluate work processes to improve prediction accuracy. This is the result of turning a large data set into a small one. One part is used to validate the model, while the other part is used to train the classifier. This procedure is repeated K-times with different validation subsets [24]. Other benefits of Cross-Validation include turning the original data into data training and data testing. Tenfold is the definition for K, when K = 10. In this study the number of K used is 10. The data will be reduced to a set if the K value is set to 10, then the data will be 10 pieces of the dataset. One set of data will be used as test data, while the other set will be used as training data, and the process will be repeated for each set [25]. The use of cross-validation techniques can help in estimating the value of unknown tuning parameters and can also be used to estimate the rate of prediction error in the final model [26].

2.4. SMOTE

In this study, unbalanced data were used, therefore a method was needed that could overcome the imbalanced data. One popular technique used to deal with class imbalance problems in data sets is the Synthetic Minority Over-Sampling Technique (SMOTE) [27]. SMOTE is used to improve classifier performance on unbalanced data by creating a new minority sample [18] . However, the poor performance of the classifier cannot be solely attributed to data imbalance [28]. In the SMOTE method, samples are generated for a certain class by connecting data points on the K-Nearest Neighbor. Synthetic data points are generated from the SMOTE method so they are not a direct copy of the minority class examples, the aim is to avoid overfitting [29]. In this study, the K value used ranged from 6 to 10 to know the effect of the K value on the best performance results based on accuracy, precision, recall, F1-Score, ROC Area, and AUCPR in deposit classification. The simple steps of the SMOTE oversampling procedure include the following [27]:

For every X_0 that belongs to the minority class, then do the following action:

1. Choose one of the K nearest neighbors X, which is part of the minority class.
2. Create a new pattern Z by placing it at a random point on the line segment connecting the original pattern and the selected neighbors, according to the following steps:

$$Z = X_0 + w\,(X - X_0) \tag{1}$$

   w in (1) here is interpreted as a uniform random variable with a range [0,1].
3. Repeat the first and second steps as much as N/100, and in each iteration process, a minority sample is taken. The samples will be combined to produce the final result.

2.5. Boosting

One ensemble technique that can improve the performance of some weak classification results into a strong classification process is Boosting which was invented by Robert E. Schapire in 1998. Boosting itself is an average model technique originally designed for classification methods, but can also be used in regression methods [30]. This method utilizes weak learners to improve model performance, by sequentially building models and training weak learners using residual data or prediction errors from previous models. Boosting uses weighting instead of bootstrapping in repetition to produce several weak classifiers which are then combined through voting to get classifier boosting. Boosting is an Ensemble Learning technique that only uses one type of base-model model by doing sequential learning adaptively, where the results of the base-model depend on the results of the previous base-model which are then combined to get the best result [31]. The Boosting algorithm principle can be seen in Fig. 2 [11].

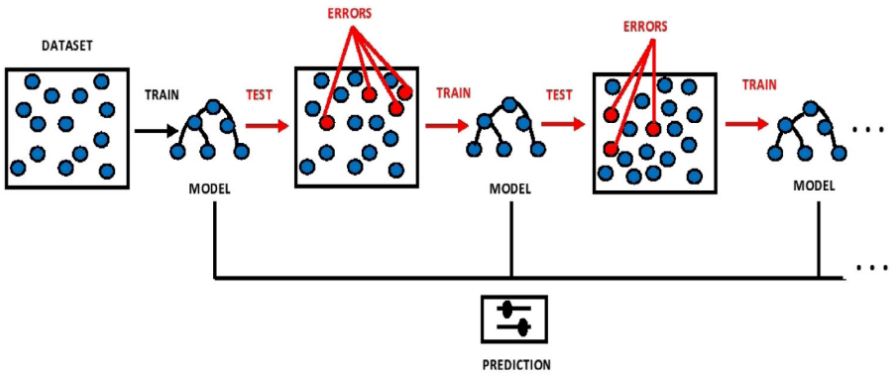

Fig. 2. The Boosting algorithm principle

2.6. Extreme Gradient Boosting

XGBoost stands for Extreme Gradient Boosting. The XGBoost algorithm is an efficient machine learning algorithm based on the Decision Tree algorithm as its main unit. The last model calculated by XGBoost is a Tree that consists of several Decision Trees. XGBoost is more accurate than single decision algorithms and can effectively improve and optimize the Gradient Boosting Decision Tree (GBDT) algorithm [11] and due to its scalability, XGBoost has great potential for widespread use in real-world binary classification tasks which generally involve a large amount of data and unbalanced labels [16]. XGBoost's highly scalable or parallelizable capabilities, fast execution, and superior performance compared to other algorithms are some of its main benefits [14]. In addition, it is proven that XGBoost has higher performance and recall when compared to other commonly used machine learning algorithms [32]. So the use of the XGBoost algorithm is expected to complete the classification in this study properly. The evolution that occurs in the XGBoost algorithm is shown in Fig. 3 [33] and the XGBoost algorithm flow is shown in Fig. 4 [34].

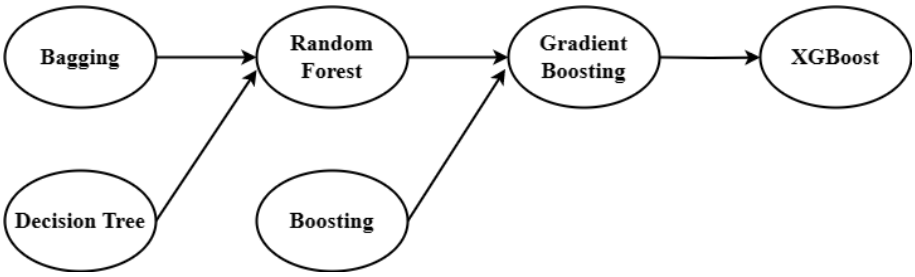

Fig. 3. The Evolution of the XGBoost Algorithm

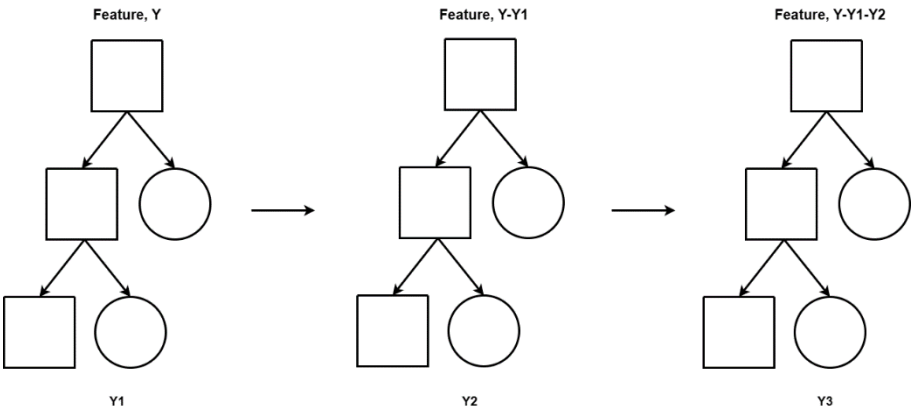

Fig. 4. The Flow of the XGboost Algorithm

The implementation flow of XGBoost according to [35] is as follows:

1. *Input: n*-dimensional data, where $X \in R^n$ and target, $Y \in R$.
2. Initialize the model with a constant value using the equation

$$F_o(x) = argmin_y \sum_{i=1}^{N} L(Y, \gamma) \quad (2)$$

Where $L(Y, F(x))$ is the differentiated *lost function* and N is the number of samples

3. Calculate the *pseudo residual* with the following equation

$$r_{im} = -\left[\frac{[\,\delta L(Y, F(X_i))}{[\,\delta F(X_i\,)}\right] \quad (3)$$

Where i is 1, 2, 3, …, N

4. Fit the *base tree* using training data with the following equation

$$(X_i, rim) \quad (4)$$

Where i is 1, 2, 3, …, N.

5. Calculate the *multiplier* using the following equation

$$\gamma_m = argmin_\gamma \sum_{i=1}^{N} L(Y_i, F_{m-1}\,(X_i) + \gamma_{hm}(X_i)) \quad (5)$$

6. Update the model using the following equation

$$F_m(x) = F_{m-1}\,(x) + \gamma_m h_m(x) \quad (6)$$

Repeat the steps in points 3 to 6 as many as m where $m = n\_iteration$

Although XGBoost is known to have excellent performance in all aspects, there are still some issues that need attention. One of the problems is having multiple parameters, where different combinations of parameters can produce different evaluation values. The three main parameter sets for XGBoost are Booster, General, and Task. Table 2 contains the important parts of each set of parameters. In most cases, the Booster parameter is used to specify the details of the boosting tree, which refers to the exact definition of each tree [36]. The Grid Search technique can be used in conjunction with a tenfold cross-validation algorithm to search for the most optimal combination of parameter values in each mode [23][37].

Table 2. List of XGBoost parameters

Type	Parameter	Default	Explain
Booster	learning_rate	0.3	Shrinking the weight on each step
	min_child_weight	1	Defines the minimum
	max_depth	6	Control over-fitting
	gamma	0	Specifies the minimum loss reduction required to make a split
	max_delta_step	0	Help in logistic regression
	subsample	1	Control the samples proportion
	comsample_bytree	1	Columns fraction of randomly samples
	colsample_bylevel	1	The subsample ratio of columns for each split, in each level
	lambda	1	L2 regularization term on weights
	alpha	1	L1 regularization term on weights
	scale_pos_weight	1	Helps in faster convergence
General	booster	gbtree	Select the model for each iteration
	silent	0	Output message switch
	nthread	max	Parallel processing and input the system core number
Task	objective	reg:linear	Minimizing the loss function
	eval_metric	according to objective	Validation data
	seed	0	Random seed

2.7. Evaluation

Assessment of indicators is very crucial in evaluating the performance of each machine learning algorithm. There are many scoring indicators available in the field of classification, in this study several scoring indexes were used for classification algorithms such as Accuracy, and for datasets that have unbalanced classes, it is necessary to pay attention to evaluation matrices such as Precision, Recall, F1-Score, and ROC Area and AUCPR in the class minority [7].

Accuracy is the acquisition of truth from every event that is predicted correctly. However, accuracy cannot differentiate the classifying samples from each class. This is especially true for the positive class. The equation used to calculate Accuracy is as follows [38]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Precision indicates how many of the positive predictions are actually positive. The formula used to calculate precision is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

The recall is a metric that counts how many positive predictions were made. To calculate recall, the following equation is used:

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

F1-Score is one of the evaluation metrics that combines both precision and recall values, which are calculated using the following equation:

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{10}$$

ROC (Receiver Operating Characteristic) Area is used as a more precise metric to evaluate the performance of a classification model and helps when comparing two models. The ROC curve displays the relationship between the true positive rate (TPR) and the false positive rate (FPR) in a dimensional plot, with TPR on the y-axis and FPR on the x-axis. This curve gives an idea of how well the model distinguishes the existing classes. The closer the curve is to the upper left corner, the better the predictive ability of the model's classification [33].

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

$$FPR = \frac{FP}{FP + TN} \tag{12}$$

In equations (7), (8), (9), (10), (11) and (12), TP (True Positive) is the number that is correctly classified that an instance is positive, FP (False Positive) is the number of incorrect negative instances classified as positive, FN (False Negative) is the number of positive instances that are incorrectly classified as negative and TN (True Negative) is the number of truly negative instances [7], [26].

AUCPR (Area Under the Curve of Precision-Recall) is a measure that measures the area under the precision-recall curve, which is often used as a metric for the accuracy of classification models in datasets with unbalanced classes [7].

3. RESULTS AND DISCUSSION

In this study, several tests were carried out, namely the application of XGBoost using hyperparameters, the application of XGBoost using hyperparameters with SMOTE using K values of 6 to 10 to determine the effect of K values which give the best performance results based on accuracy, precision, recall, F1-Score, ROC Area, and AUCPR on deposit classification, and the final test is to compare the evaluation results obtained with other studies using the same dataset, namely the UCI Bank Marketing Dataset obtained from a banking institution in Portugal to promote bank deposit products from May 2008 to November 2010 which published in 2012 with the aim of this test is to measure the performance of XGBoost and XGBoost with SMOTE in identifying deposits.

The data that has been preprocessed is then used in testing. In the tests carried out, the distribution of training data and test data was carried out using K-Fold Cross Validation which is a popular way to avoid overfitting, where K is 10 times. The dataset used will be divided into 10 of the same size, of which 1 part is used as test data and the other 9 parts are used as training data. The process will be repeated until each part is used as test data so that in this study it will produce an average value of the total performance results for each fold. In addition, this research also applies hyperparameter tuning using GridSearchCV. GridSearchCV is used in this study to find the best combination values for hyperparameters to be used in XGBoost with a thorough

search. By doing this thorough search, we can find the combination of hyperparameter values that provide the best performance for the machine learning model.

The hyperparameters used in this research are *n_estimator*, *learning_rate*, *max_depth*, *comsample_bytree*, and *subsample*. Then do the hyperparameter settings shown in Table 3.

Table 3. Hyperparameter setup round 1

Round 1	
Hyperparameter	**Value**
n_estimators	[100, 200, 300]
learning_rate	[0.01, 0.1]
max_depth	[3, 5, 7]
colsample_bytree	[0.6, 0.8, 1.0]
subsample	[0.6, 0.8, 1.0]
Best Hyperparameter	**Value**
n_estimators	[300]
learning_rate	[0.1]
max_depth	[5]
colsample_bytree	[0.8]
subsample	[0.8]
Train Accuracy	**0.9455**
Test Accuracy	**0.9089**

Table 3 shows the best hyperparameter results where these hyperparameters can still be increased because the values are at the end of their range, such as *n_estimators* and *learning_rate*. So values can still be explored and searched for the best combination again using GridSearchCV, where the hyperparameter settings from round 2 are shown in Table 4.

Table 4. Hyperparameter setup round 2

Round 2	
Hyperparameter	**Value**
n_estimators	[300, 500, 1000]
learning_rate	[0.1, 0.25, 0.3]
max_depth	[5]
colsample_bytree	[0.8]
subsample	[0.8]
Best Hyperparameter	**Value**
n_estimators	[300]
learning_rate	[0.1]
max_depth	[5]
colsample_bytree	[0.8]
subsample	[0.8]
Train Accuracy	**0.9455**
Test Accuracy	**0.9089**

Table 4 shows that the best hyperparameter results are *n_estimator* 300, *learning_rate* 0.1, *max_depth* 5, *colsample_bytree* 0.8, and *subsample* 0.8. After getting the best combination value of the hyperparameters, then enter it into the model that will be used. Before entering hyperparameters in the model, testing is carried out with XGBoost without hyperparameters, where the results will be compared with the XGBoost model with hyperparameters. The results of the XGBoost test comparison without hyperparameters and with hyperparameters are shown in Table 5.

Table 5 shows that there is an increase in performance for each evaluation result obtained when XGBoost uses hyperparameters. Hyperparameters are proven to have a good effect on improving performance for each evaluation result and will also be used when XGBoost implements SMOTE.

Furthermore, testing of the application of SMOTE to XGBoost was carried out using different K values, namely 6, 7, 8, 9, and 10 to know the K value which gives the best performance results in deposit classification. As for the results of the test obtained with K = 6, shown in Table 6.

Table 6 shows the results of the XGBoost test with SMOTE using a K=6 value, namely accuracy 0.90983, precision 0.78635, recall 0.75448, F1-Score 0.58854, ROC Area 0.93705, and AUCPR 0.63497. These results

indicate that when compared with the test results with XGBoost without SMOTE, only the recall value and F1-Score experience an increase in performance. The results of the test obtained with K=7, are shown in Table 7.

Table 5. XGBoost test results without hyperparameters and with hyperparameters

Testing	XGBoost (without hyperparameters)						XGBoost (with hyperparameters)					
	Accuracy	Precision	Recall	F1-Score	ROC Area	AUCPR	Accuracy	Precision	Recall	F1-Score	ROC Area	AUCPR
1	0.9023	0.7686	0.7167	0.5318	0.9307	0.5836	0.9047	0.7799	0.7066	0.5238	0.934	0.6027
2	0.9049	0.7759	0.7235	0.5445	0.9364	0.6321	0.9117	0.7979	0.7356	0.5723	0.9426	0.6492
3	0.9073	0.7821	0.736	0.564	0.9384	0.626	0.9135	0.8014	0.7469	0.5889	0.9424	0.6445
4	0.9137	0.8069	0.7347	0.5761	0.9393	0.6672	0.9168	0.8214	0.7332	0.5813	0.9433	0.6831
5	0.9018	0.7697	0.7033	0.5142	0.9305	0.6058	0.9049	0.7771	0.7207	0.5416	0.932	0.6096
6	0.9069	0.7822	0.7292	0.5554	0.931	0.6095	0.9111	0.7943	0.7406	0.5768	0.9352	0.6247
7	0.9073	0.7855	0.7229	0.549	0.9369	0.6245	0.9095	0.7924	0.7282	0.5597	0.9381	0.6322
8	0.9062	0.7881	0.7017	0.5204	0.9293	0.6245	0.9095	0.8001	0.7085	0.5358	0.9334	0.6173
9	0.9086	0.785	0.7433	0.5747	0.9356	0.6514	0.9075	0.7838	0.732	0.56	0.9391	0.6626
10	0.9095	0.7913	0.7315	0.5635	0.9408	0.6444	0.9124	0.7993	0.7405	0.5796	0.943	0.6627
Average	0.90685	0.78353	0.72428	0.54936	0.93489	0.6269	0.91016	0.79476	0.72928	0.56198	0.93831	0.63886

Table 6. The XGBoost test results with SMOTE use the value K=6

Testing	XGBoost + SMOTE (K=6)					
	Accuracy	Precision	Recall	F1-Score	ROC Area	AUCPR
1	0.9069	0.7801	0.7382	0.5655	0.9313	0.602
2	0.9102	0.7863	0.7585	0.5932	0.9422	0.6503
3	0.9095	0.7829	0.7684	0.6018	0.9401	0.6228
4	0.9151	0.8022	0.7617	0.6074	0.9403	0.6751
5	0.902	0.7652	0.7346	0.5521	0.9305	0.6155
6	0.9102	0.786	0.7622	0.5972	0.9359	0.6272
7	0.9115	0.7911	0.758	0.596	0.9365	0.6238
8	0.9056	0.777	0.7317	0.5557	0.9294	0.5973
9	0.91	0.7856	0.7604	0.595	0.9401	0.6639
10	0.9173	0.8071	0.7711	0.6215	0.9442	0.6718
Average	0.90983	0.78635	0.75448	0.58854	0.93705	0.63497

Table 7. XGBoost test results with SMOTE using a value of K=7

Testing	XGBoost + SMOTE (K=7)					
	Accuracy	Precision	Recall	F1-Score	ROC Area	AUCPR
1	0.9065	0.7787	0.7379	0.5644	0.9321	0.6144
2	0.9124	0.7918	0.7655	0.6048	0.9427	0.6537
3	0.9111	0.7875	0.7684	0.6051	0.9394	0.6276
4	0.9153	0.804	0.7585	0.6047	0.9399	0.6787
5	0.9056	0.7755	0.7399	0.5647	0.9309	0.6128
6	0.9104	0.786	0.7656	0.601	0.9351	0.6242
7	0.9113	0.7916	0.753	0.5904	0.9368	0.6247
8	0.9051	0.7777	0.7216	0.5431	0.9299	0.6023
9	0.9109	0.7873	0.7659	0.6022	0.94	0.6586
10	0.9186	0.8082	0.7825	0.6349	0.9455	0.6763
Average	0.91072	0.78883	0.75588	0.59153	0.93723	0.63733

Table 7 shows the results of the XGBoost test with SMOTE using a K=7 value, namely accuracy 0.91072, precision 0.78883, recall 0.75588, F1-Score 0.59153, ROC Area 0.93723, and AUCPR 0.63733. These results indicate that when compared with the test results with XGBoost without SMOTE, only accuracy, recall and F1-Score values experience an increase in performance. The results of the test obtained with K=8, are shown in Table 8.

Table 8 shows the results of the XGBoost test with SMOTE using a K=8 value, namely accuracy 0.91004, precision 0.78714, recall 0.75411, F1-Score 0.58849, ROC Area 0.93709, and AUCPR 0.63363. These results indicate that when compared with the test results with XGBoost without SMOTE, only the recall value and F1-Score experience an increase in performance. As for the results of the tests obtained with K=9, they are shown in Table 9.

Table 8. XGBoost test results with SMOTE using a value of K=8

Testing	XGBoost + SMOTE (K=8)					
	Accuracy	Precision	Recall	F1-Score	ROC Area	AUCPR
1	0.9062	0.7787	0.7345	0.5602	0.9328	0.6015
2	0.912	0.7922	0.7579	0.5963	0.9426	0.648
3	0.9095	0.7833	0.7659	0.5994	0.9408	0.6375
4	0.9151	0.8013	0.765	0.6105	0.9396	0.6702
5	0.9044	0.7723	0.7376	0.5601	0.9305	0.6126
6	0.9093	0.7832	0.7625	0.5957	0.935	0.6241
7	0.912	0.793	0.7566	0.5955	0.9362	0.6235
8	0.9067	0.7816	0.7282	0.5539	0.9311	0.6
9	0.9093	0.7837	0.7601	0.5933	0.9379	0.6525
10	0.9159	0.8021	0.7728	0.62	0.9444	0.6664
Average	0.91004	0.78714	0.75411	0.58849	0.93709	0.63363

Table 9. The results of the XGBoost test with SMOTE use the value K=9

Testing	XGBoost + SMOTE (K=9)					
	Accuracy	Precision	Recall	F1-Score	ROC Area	AUCPR
1	0.9056	0.7779	0.7276	0.551	0.9316	0.6031
2	0.9115	0.7902	0.7601	0.5976	0.9413	0.6458
3	0.9122	0.791	0.7683	0.6073	0.9397	0.6394
4	0.9135	0.799	0.7542	0.5965	0.9392	0.6646
5	0.9069	0.7798	0.7398	0.5673	0.9308	0.6163
6	0.912	0.7898	0.7706	0.609	0.937	0.6351
7	0.9133	0.7958	0.7631	0.6048	0.9381	0.6321
8	0.9075	0.7834	0.7336	0.5618	0.9301	0.6012
9	0.9091	0.7823	0.764	0.5967	0.938	0.6536
10	0.9164	0.8039	0.7715	0.6197	0.9437	0.6724
Average	0.9108	0.78931	0.75528	0.59117	0.93695	0.63636

Table 9 shows the results of the XGBoost test with SMOTE using a K=9 value, namely accuracy 0.9108, precision 0.78931, recall 0.75528, F1-Score 0.59117, ROC Area 0.93695, and AUCPR 0.63636. These results indicate that when compared with the test results with XGBoost without SMOTE, only accuracy, recall and F1-Score values experience an increase in performance. As for the results of the tests obtained with K=10, they are shown in Table 10.

Table 10. XGBoost test results with SMOTE using a value of K=10

Testing	XGBoost + SMOTE (K=10)					
	Accuracy	Precision	Recall	F1-Score	ROC Area	AUCPR
1	0.9058	0.7779	0.731	0.5553	0.9321	0.6017
2	0.9117	0.7911	0.7594	0.5974	0.9419	0.6433
3	0.9089	0.7818	0.7631	0.5953	0.94	0.6294
4	0.9157	0.8042	0.7629	0.61	0.9401	0.6737
5	0.9075	0.7814	0.7427	0.5717	0.932	0.6211
6	0.9095	0.7834	0.7651	0.5986	0.9351	0.6252
7	0.9109	0.7896	0.7552	0.5917	0.9366	0.6292
8	0.9062	0.7792	0.7321	0.5574	0.9303	0.6026
9	0.9078	0.7791	0.7592	0.5892	0.939	0.6558
10	0.9188	0.8096	0.7802	0.6334	0.945	0.6777
Average	0.91028	0.78773	0.75509	0.59	0.93721	0.63597

Table 10 shows the results of the XGBoost test with SMOTE using a K=10 value, namely accuracy 0.91028, precision 0.78773, recall 0.75509, F1-Score 0.59, ROC Area 0.93721, and AUCPR 0.63597. These results indicate that when compared with the test results with XGBoost without SMOTE, only accuracy, recall, and F1-Score values experience an increase in performance. To facilitate understanding of the results of testing the K value, the results are visualized based on each evaluation. Visualization of the results of testing the K value is shown in Fig. 6, Fig. 7, Fig. 8, Fig. 9, and Fig. 10.

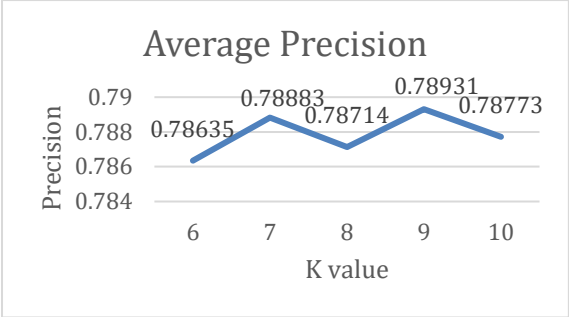

Fig. 5. Graph of average Precision for all test scenarios

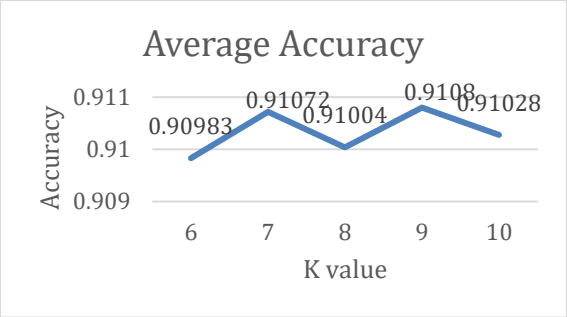

Fig. 6. Graph of average Accuracy for all test scenarios

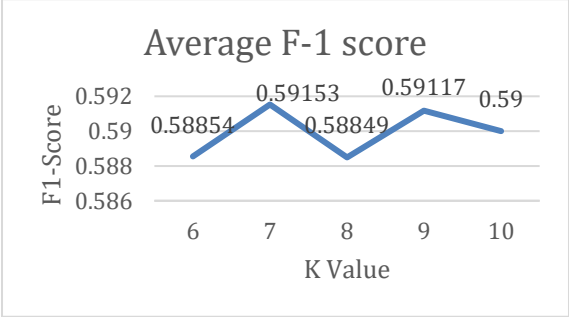

Fig. 7. Graph of average F-1 Score for all test scenarios

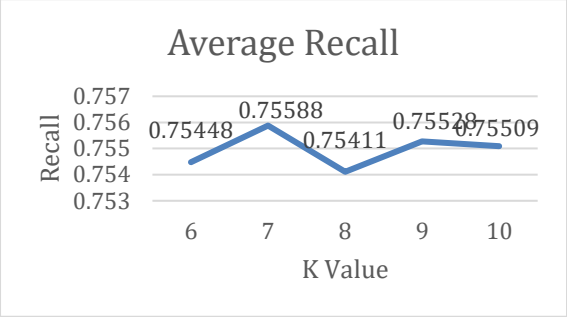

Fig. 8. Graph of average Recall for all test scenarios

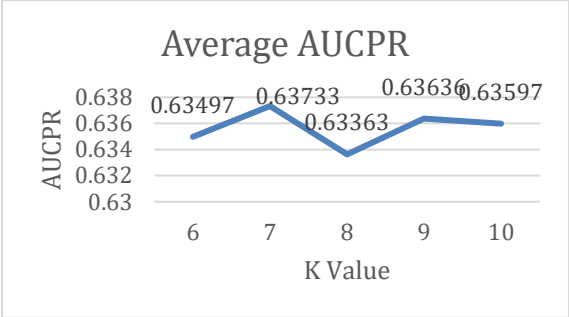

Fig. 9. Graph of average AUCPR for all test scenarios

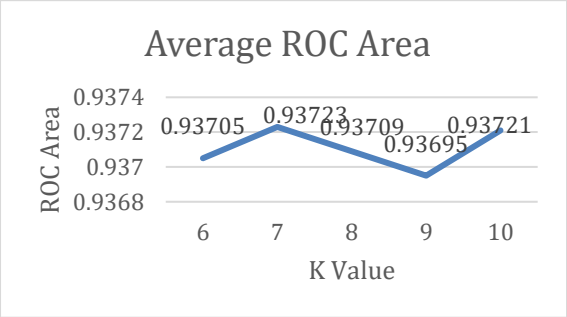

Fig. 10. Graph of average ROC Area for all test scenarios

Fig. 6, Fig. 7, Fig. 8, Fig. 9, and Fig. 10 indicate that the value of K=7 gives the best recall, F1-Score, ROC Area, and AUCPR evaluation results. Meanwhile, the best accuracy and precision results are obtained with a value of K=9. So, from these results, the value of K used to compare XGboost and XGboost with SMOTE is K=7. The results of the comparison of XGBoost and XGBoost with SMOTE are shown in Table 11.

The test results in Table 11 show that SMOTE can improve performance for accuracy, recall, and F1-Score results. However, SMOTE still cannot improve the performance of precision, ROC Area, and AUCPR results. SMOTE provides a small performance increase, this is due to factors such as the characteristics of the dataset: although SMOTE can help improve class balance in unbalanced datasets, there is a very large imbalance between the number of cases of the majority and minority classes in a dataset. Under these conditions, the probability that the nearest neighbors of minority class cases are majority class cases is very high, which results in models tending to predict minority class cases as majority cases which can result in inaccurate results [39] and information overload: the UCI Bank Marketing dataset has several features. relatively few, so information overload (overfitting) may not be a big problem in this dataset. Therefore, SMOTE may not bring much change to the model's performance.

Further testing is carried out by comparing the evaluation results obtained with other studies using the same dataset the aim of this test is to measure the performance of XGBoost and XGBoost with SMOTE in identifying deposits. A comparison of results is shown in Table 12.

Table 11. Comparison of XGBoost and XGBoost test results with SMOTE

Testing	XGBoost (with hyperparameter)						XGBoost + SMOTE (with hyperparameters)					
	Accuracy	Precision	Recall	F1-Score	ROC Area	AUCPR	Accuracy	Precision	Recall	F1-Score	ROC Area	AUCPR
1	0.9047	0.7799	0.7066	0.5238	0.934	0.6027	0.9065	0.7787	0.7379	0.5644	0.9321	0.6144
2	0.9117	0.7979	0.7356	0.5723	0.9426	0.6492	0.9124	0.7918	0.7655	0.6048	0.9427	0.6537
3	0.9135	0.8014	0.7469	0.5889	0.9424	0.6445	0.9111	0.7875	0.7684	0.6051	0.9394	0.6276
4	0.9168	0.8214	0.7332	0.5813	0.9433	0.6831	0.9153	0.804	0.7585	0.6047	0.9399	0.6787
5	0.9049	0.7771	0.7207	0.5416	0.932	0.6096	0.9056	0.7755	0.7399	0.5647	0.9309	0.6128
6	0.9111	0.7943	0.7406	0.5768	0.9352	0.6247	0.9104	0.786	0.7656	0.601	0.9351	0.6242
7	0.9095	0.7924	0.7282	0.5597	0.9381	0.6322	0.9113	0.7916	0.753	0.5904	0.9368	0.6247
8	0.9095	0.8001	0.7085	0.5358	0.9334	0.6173	0.9051	0.7777	0.7216	0.5431	0.9299	0.6023
9	0.9075	0.7838	0.732	0.56	0.9391	0.6626	0.9109	0.7873	0.7659	0.6022	0.94	0.6586
10	0.9124	0.7993	0.7405	0.5796	0.943	0.6627	0.9186	0.8082	0.7825	0.6349	0.9455	0.6763
Average	0.91016	0.79476	0.72928	0.56198	0.93831	0.63886	0.91072	0.78883	0.75588	0.59153	0.93723	0.63733

Table 12. Comparison of Accuracy Results with other algorithms

Algoritma	Accuracy(%)
KNN [9]	86.229
RF [9]	90.884
LR [9]	86.185
ANN [9]	90.286
NB [9]	86.868
SVM [9]	89.670
XGBoost	91.016
XGBoost+SMOTE	91.072

The comparison results in Table 12 show that the accuracy values obtained by XGBoost and SMOTE are 91.016% and 91.072%. This shows that the accuracy results produced by the XGBoost and SMOTE algorithms are superior when compared to other algorithms such as K-Nearest Neighbor, Random Forest, Logistic Regression, Artificial Neural Network, Naïve Bayes, and Support Vector Machine in deposit classification using K-Fold Cross Validation for training data and test data with the same dataset, namely the UCI Bank Marketing dataset obtained by a banking institution in Portugal.

4. CONCLUSION

In conclusion, this study focuses on addressing the class imbalance in the Marketing Bank data set using the XGBoost and SMOTE techniques. The test results reveal that the best hyperparameters for XGBoost are n_estimator 300, learning_rate 0.1, max_depth 5, colsample_bytree 0.8, and subsample 0.8 with the resulting accuracy of 0.91016. If you do not use the resulting accuracy hyperparameter 0.90685. For the best SMOTE test results based on recall, F1-Score, ROC Area, and AUCPR using a value of K=7 and if based on accuracy and precision using a value of K=9. Based on this, the XGBoost test with SMOTE uses a value of K=7. Comparing the performance of XGBoost and SMOTE with other algorithms, including K-Nearest Neighbor, Random Forest, Logistic Regression, Artificial Neural Network, Naïve Bayes, and Support Vector Machine, it is evident that both XGBoost and SMOTE consistently outperform other algorithms in terms of accuracy. However, it is important to recognize the limitations of our study, particularly the limited testing using only XGBoost and SMOTE. Therefore, further research is needed to explore optimization methods, for example, such as feature selection [38] and try other techniques to handle class imbalances in data, for example, ADASYN [40].

Despite gradual improvements through the use of SMOTE, overall accuracy for classifying deposits remained relatively stable, with an increase of only 0.00056 from 0.91016 to 0.91072. These findings highlight the complexity of class imbalance and the impact of methods and datasets on performance. To promote progress in this field, future studies should aim to address these limitations by expanding the applicability of the algorithm and exploring additional techniques to deal with class imbalance. In short, XGBoost and SMOTE have proven to be effective in dealing with a class imbalance in the Marketing Bank dataset. While the results provide valuable insights, further research is needed to optimize XGBoost performance, incorporate feature

selection methods, and explore alternative techniques for dealing with imbalances such as ADASYN. Thus, efforts are continuously being made to improve the accuracy and applicability of classification models in similar domains.

REFERENCES

[1] M. Rashid Farooqi and N. Iqbal, "Performance Evaluation for Competency of Bank Telemarketing Prediction using Data Mining Techniques," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2, pp. 5766–5774, 2019, https://doi.org/10.35940/ijrte.A1269.078219.

[2] C. Yan, M. Li, and W. Liu, "Prediction of bank telephone marketing results based on improved .w,hale algorithms optimizing S_Kohonen network," *Appl. Soft Comput. J.*, vol. 92, p. 106259, 2020, https://doi.org/10.1016/j.asoc.2020.106259

[3] F. Safarkhani and S. Moro, "Improving the Accuracy of Predicting Bank Depositor's Behavior Using a Decision Tree," *Appl. Sci.*, vol. 11, no. 19, p. 9016, 2021, https://doi.org/10.3390/app11199016

[4] K. Chitra and B. Subashini, "Data Mining Techniques and its Applications in Banking Sector," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 8, pp. 219–226, 2013, https://www.scinapse.io/papers/2396932848.

[5] K. Wisaeng, "A Comparison of Different Classification Techniques for Bank Direct Marketing," *Int. J. Soft Comput. Eng.*, vol. 3, no. 4, pp. 116–119, 2013, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0e85310bc60dfcf2cbe9c8d303129adb20c8ef23.

[6] B. Valarmathi, T. Chellatamilan, H. Mittal, J. Jagrit, and S. Shubham, "Classification of imbalanced banking dataset using dimensionality reduction," in *2019 International Conference on Intelligent Computing and Control Systems, ICCS 2019*, pp. 1353–1357, 2019, https://doi.org/10.1109/ICCS45141.2019.9065648.

[7] A. Verma, "Evaluation of classification algorithms with solutions to class imbalance problem on bank marketing dataset using WEKA," *Int. Res. J. Eng. Technol.*, vol. 6, no. 3, pp. 54–60, 2019, https://www.irjet.net/archives/V6/i3/IRJET-V6I308.pdf.

[8] S. B. Yang and T. L. Chen, "Uncertain decision tree for bank marketing classification," *J. Comput. Appl. Math.*, vol. 371, p. 112710, 2020, https://doi.org/10.1016/j.cam.2020.112710.

[9] E. Zeinulla, K. Bekbayeva, and A. Yazici, "Comparative study of the classification models for prediction of bank telemarketing," *IEEE 12th Int. Conf. Appl. Inf. Commun. Technol. AICT 2018 - Proc.*, pp. 1-5, 2018, https://doi.org/10.1109/ICAICT.2018.8747086.

[10] T. R. Mahesh, V. Vinoth Kumar, V. Muthukumaran, H. K. Shashikala, B. Swapna, and S. Guluwadi, "Performance Analysis of XGBoost Ensemble Methods for Survivability with the Classification of Breast Cancer," *J. Sensors*, vol. 2022, pp. 1–8, 2022, https://doi.org/10.1155/2022/4649510.

[11] S. K. Kiangala and Z. Wang, "An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment," *Mach. Learn. with Appl.*, vol. 4, pp. 1–15, 2021, https://doi.org/10.1016/j.mlwa.2021.100024.

[12] S. Li and X. Zhang, "Research on orthopedic auxiliary classification and prediction model based on XGBoost algorithm," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 1971–1979, 2020, https://doi.org/10.1007/s00521-019-04378-4.

[13] N. H. N. B. M. Shahri, S. B. S. Lai, M. B. Mohamad, H. A. B. A. Rahman, and A. Bin Rambli, "Comparing the Performance of AdaBoost, XGBoost, and Logistic Regression for Imbalanced Data," *Math. Stat.*, vol. 9, no. 3, pp. 379–385, 2021, https://doi.org/10.13189/ms.2021.090320.

[14] N. K. Sinha, M. Khulal, M. Gurung, and A. Lal, "Developing A Web based System for Breast Cancer Prediction using XGboost Classifier," *Int. J. Eng. Res.*, vol. 9, no. 6, pp. 852–856, 2020, https://doi.org/10.17577/IJERTV9IS060612.

[15] A. Prabha, J. Yadav, A. Rani, and V. Singh, "Design of intelligent diabetes mellitus detection system using hybrid feature selection based XGBoost classifier," *Comput. Biol. Med.*, vol. 136, pp. 1–9, 2021, https://doi.org/10.1016/j.compbiomed.2021.104664.

[16] C. Wang, C. Deng, and S. Wang, "Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost," *Pattern Recognit. Lett.*, vol. 136, pp. 190–197, 2020, https://doi.org/10.1016/j.patrec.2020.05.035.

[17] M. S. Islam, M. Arifuzzaman, and M. S. Islam, "SMOTE Approach for Predicting the Success of Bank Telemarketing," in *2019 4th Technology Innovation Management and Engineering Science International .Conference (TIMES-iCON)*, pp. 1–5, 2019, https://doi.org/10.1109/TIMES-iCON47539.2019.9024630.

[18] Z. Hengyu, "Improved SMOTE algorithm for imbalanced dataset," in *Proceedings - 2020 Chinese Automation Congress, CAC 2020*, pp. 693–697, 2020, https://doi.org/10.1109/CAC51589.2020.9326603.

[19] A. J. Mohammed, M. M. Hassan, and D. H. Kadir, "Improving Classification Performance for a Novel Imbalanced Medical Dataset using SMOTE Method," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 3, pp. 3161–3172, 2020, https://doi.org/10.30534/ijatcse/2020/104932020.

[20] W. Du, "Application of Improved SMOTE and XGBoost Algorithm in the Analysis of Psychological Stress Test for College Students," *J. Electr. Comput. Eng.*, vol. 2022, pp. 1–8, 2022, https://doi.org/10.1155/2022/2760986.

[21] R. C. Chen, C. Dewi, S. W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *J. Big Data*, vol. 7, no. 1, pp. 2–26, 2020, https://doi.org/10.1186/s40537-020-00327-4

[22] D. A. Rusdah and H. Murfi, "XGBoost in handling missing values for life insurance risk prediction," *SN Appl. Sci.*, vol. 2, no. 8, pp. 1–10, 2020, https://doi.org/10.1007/s42452-020-3128-y.
[23] P. Cerda and G. Varoquaux, "Encoding High-Cardinality String Categorical Variables," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1164–1176, 2022, https://doi.org/10.1109/TKDE.2020.2992529.
[24] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Comput. Secur.*, vol. 70, pp. 255–277, 2017, https://doi.org/10.1016/j.cose.2017.06.005.
[25] H. Wei, C. Hu, S. Chen, Y. Xue, and Q. Zhang, "Establishing a software defect prediction model via effective dimension reduction," *Inf. Sci. (Ny).*, vol. 477, pp. 399–409, 2019, https://doi.org/10.1016/j.ins.2018.10.056.
[26] J. S. Clark, *Model Assessment and Selection*. 2020, https://doi.org/10.2307/j.ctv15r5dgv.9.
[27] D. Elreedy and A. F. Atiya, "A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance," *Inf. Sci. (Ny).*, vol. 505, pp. 32–64, 2019, https://doi.org/10.1016/j.ins.2019.07.070.
[28] 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, no. 2021, p. 100267, 2022, https://doi.org/10.1016/j.mlwa.2022.100267.
[29] E. Ileberi, Y. Sun, and Z. Wang, "Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost," *IEEE Access*, vol. 9, pp. 165286–165294, 2021, https://doi.org/10.1109/ACCESS.2021.3134330.
[30] I. Syarif, E. Zaluska, A. Prugel-Bennett, and G. Wills, "Application of Bagging, Boosting and Stacking to Intrusion Detection," in *Machine Learning and Data Mining in Pattern Recognition*, vol. 7376, pp. 593–602, 2012, https://doi.org/10.1007/978-3-642-31537-4_46.
[31] J. Wang, Z. Sun, B. Bao, and D. Shi, "Malicious synchrophasor detection based on highly imbalanced historical operational data," *CSEE J. Power Energy Syst.*, vol. 5, no. 1, pp. 11–20, 2019, https://doi.org/10.17775/CSEEJPES.2018.00200.
[32] A. Paleczek, D. Grochala, and A. Rydosz, "Artificial breath classification using xgboost algorithm for diabetes detection," *Sensors*, vol. 21, no. 12, pp. 1–18, 2021, https://doi.org/10.3390/s21124187,
[33] M. Guo, Z. Yuan, B. Janson, Y. Peng, Y. Yang, and W. Wang, "Older pedestrian traffic crashes severity analysis based on an emerging machine learning xgboost," *Sustain.*, vol. 13, no. 2, pp. 1–26, 2021, https://doi.org/10.3390/su13020926.
[34] K. Budholiya, S. K. Shrivastava, and V. Sharma, "An optimized XGBoost based diagnostic system for effective prediction of heart disease," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4514–4523, 2022, https://doi.org/10.1016/j.jksuci.2020.10.013.
[35] K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76516–76531, 2020, https://doi.org/10.1109/ACCESS.2020.2989857.
[36] Y. Jiang, G. Tong, H. Yin, and N. Xiong, "A Pedestrian Detection Method Based on Genetic Algorithm for Optimize XGBoost Training Parameters," *IEEE Access*, vol. 7, pp. 118310–118321, 2019, https://doi.org/10.1109/ACCESS.2019.2936454.
[37] M. U. N. Nisa, D. Mahmood, G. Ahmed, S. Khan, M. A. Mohammed, and R. Damaševičius, "Optimizing prediction of youtube video popularity using xgboost," *Electron.*, vol. 10, no. 23, 2021, https://doi.org/10.3390/electronics10232962.
[38] N. Hasan and Y. Bao, "Comparing different feature selection algorithms for cardiovascular disease prediction," *Health Technol. (Berl).*, vol. 11, no. 1, pp. 49–62, 2021, https://doi.org/10.1007/s12553-020-00499-2.
[39] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004, https://doi.org/10.1145/1007730.1007735
[40] I. Tasin, T. U. Nabil, S. Islam, and R. Khan, "Diabetes prediction using machine learning and explainable AI techniques," *Healthc. Technol. Lett.*, pp. 1–10, 2022, https://doi.org/10.1049/htl2.12039.

BIOGRAPHY OF AUTHORS

Dina Arifah is an undergraduate student in Department of Computer Science, Lambung Mangkurat University. Her research interest is centered on Data Science. Email: dinaarifah40@gmail.com

Triando Hamonangan Saragih is a lecturer in Department of Computer Science, Lambung Mangkurat University. His research interest is centered on Data Science. Email: triando.saragih@ulm.ac.id

Dwi Kartini is a lecturer in Department of Computer Science, Lambung Mangkurat University. Her research interest is centered on Data Science. Email: dwikartini@ulm.ac.id

Muliadi is a lecturer in Department of Computer Science, Lambung Mangkurat University. His research interest is centered on Data Science. Email: muliadi@ulm.ac.id

Muhammad Itqan Mazdadi is a lecturer in Department of Computer Science, Lambung Mangkurat University. His research interest is Data Science, Computer Network, and Digital Forensics. Email: mazdadi@ulm.ac.id