Institut Riset dan Publikasi Indonesia (IRPI)
IJATIS: Indonesian Journal of Applied Technology and Innovation Science
Journal Homepage: https://journal.irpi.or.id/index.php/ijatis
Vol. 2 Iss. 2 August 2025, pp: 104-115
ISSN(P): 3032-7466 | ISSN(E): 3032-7474

Classification of A Credit Card Fraud Detection Model Using XGBoost with Smote and Gridsearchcv Optimization

Amelia Rahmadani[1*], M. Zacky[2], John Paaul Michael[3]

[1,2]Department of Information Systems, Faculty of Science and Technology,
Sultan Syarif Kasim State Islamic University, Riau, Indonesia
[3]Department of Aeronautical Engineering, Faculty of Aerospace Engineering,
University College of Aviation, Malaysia

E-Mail: [1]12250320334@students.uin-suska.ac.id,
[2]12250315382@students.uin-suska.ac.id, [3]johnpaauljpm@gmail.com

Received Jul 20th 2025; Revised Aug 20th 2025; Accepted Aug 25th 2025; Available Online Aug 31th 2025
Corresponding Author: Amelia Rahmadani

Abstract

The development of digital technology has driven rapid growth in online transactions, resulting in an increase in the volume of digital transactions. This growth also increases the risk of credit card fraud, particularly in transactions where a card is not physically present. By employing the Extreme Gradient Boosting (XGBoost) method in conjunction with the Synthetic Minority Over-sampling Technique (SMOTE) to solve class imbalance and fine-tuning model parameters using GridSearchCV, this study aims to improve a fraud detection system. The dataset, which consists of anonymized credit card transactions, presents a stark imbalance with fraudulent cases accounting for only 0.172% of the data. The study involves several stages: preprocessing the data, balancing class distribution, training the model, and evaluating its performance through metrics such as F1-score, precision, recall, accuracy, and AUC-ROC. Implementation of SMOTE proved effective in enhancing the representation of rare fraud cases without introducing overfitting, while GridSearchCV identified the most effective parameter configuration. The resulting model achieved top-tier performance with 100% accuracy, 0.81 precision, 0.85 recall, an F1-score of 0.83, and an AUC-ROC of 0.979, indicating strong capability in distinguishing fraudulent from legitimate transactions. The novelty of this study lies in the systematic integration of SMOTE, XGBoost, and GridSearchCV into a unified pipeline designed to address extreme class imbalance in real-world credit card transactions. Unlike previous studies that focused solely on algorithm comparison or hyperparameter tuning, this research emphasizes reducing false negatives, which pose the greatest financial and reputational risks. The findings not only demonstrate superior performance metrics but also provide practical contributions for financial institutions, regulators, and e-commerce platforms in developing scalable, reliable, and adaptive fraud detection systems.

Keywords: Classification, Credit Card Fraud, GridsearchCV, SMOTE, XGBoost

1. INTRODUCTION

Online transactions have increased rapidly due to the advancement of digital technology, which is convenient for both businesses and customers. However, these advancements have also increased the risk of cybercrime, particularly credit card fraud. According to a TransUnion report, there was an 80% increase in suspected digital fraud attempts globally between 2019 and 2022, as the volume of digital transactions increased [1]. Fraud involving credit cards, especially when the card is not present, is one of the most common modes, accounting for approximately 65% of total credit card fraud losses [2]. A study by Mauladi et al. (2022) showed that the increasing adoption of cashless society in Indonesia is directly proportional to the increasing cybercrime activities, such as phishing and social engineering, which often target credit card transactions [3]. This is reinforced by the findings of Dewi et al. (2023), They determined that Indonesia's growing online credit card usage has resulted in a rise in credit card-related cyber incidents, affecting banks' reputations and perhaps causing financial losses [4]. In this context, this study is highly relevant for financial institutions such as banks and fintech companies, e-commerce platforms, regulators, and consumers. Financial institutions and e-commerce providers need reliable fraud detection models to protect their operations and reputations, regulators require evidence-based insights to formulate stronger cybersecurity policies, while consumers benefit from safer digital payment ecosystems. The urgency of this study lies in the fact that credit card fraud not only causes direct financial losses but also erodes public trust in digital financial services, which could hinder the transition towards a fully cashless society in Indonesia.

Various approaches have been implemented to detect credit card fraud, including rule-based systems and manual audits. However, these conventional methods have limitations in handling complex and rapidly growing transaction data. The study by Hafez et al. (2025) highlights that traditional approaches are less effective in dealing with ever-changing fraud patterns. In addition, challenges such as data class imbalance, where fraudulent transactions are much less than normal transactions, make it difficult for models to accurately detect fraud [5]. In the meantime, Assabil & Obagbuwa's (2024) assessed various machine learning models' performance,including XGBoost, and used a combined resampling strategy to overcome class imbalance. The necessity for more research into the combination of resampling techniques and machine learning algorithms is highlighted by the fact that XGBoost demonstrated competitive performance even though K-Nearest Neighbors performed the best in detecting fraudulent transactions. Furthermore, [6], to detect credit card fraud, a hybrid model combining three machine learning algorithms Decision Tree, Logistic Regression, and Naive Bayes was presented by Ali et al. (2024). Through the application of the Synthetic Minority Over-sampling Technique (SMOTE) method to address class imbalance, the model achieved a 98.1% accuracy rate and a 98.3% F1-score, indicating the efficiency of the hybrid strategy in enhancing the detection of fraudulent activities [7]. Meanwhile, Gupta et al.'s research (2022) showed XGBoost excels in detecting credit card fraud with imbalanced data, achieving an F1-score of 0.998 after Random Oversampling [8]. Then, research conducted by Joses & Saikhu, (2024) showed that XGBoost consistently outperformed CatBoost in the initial scenario. However, the most significant results were seen after hyperparameter tuning using GridSearchCV, where CatBoost combined SMOTE and FS3 to obtain the greatest accuracy of 98.01%, while XGBoost recorded 96.68% accuracy with SMOTE and FS1. These findings confirm that hyperparameter tuning and feature selection are critical in improving model performance [9]. The references reviewed in this study primarily focus on machine learning algorithms and handling class imbalance. This emphasis was chosen because the main research gap lies in the effectiveness of classification models when dealing with highly imbalanced datasets and the need for adaptive solutions in real-time fraud detection. By analyzing previous work on XGBoost, SMOTE, and hyperparameter optimization, this study identifies the strengths and limitations of existing approaches and positions itself to contribute a more comprehensive solution to the problem.

Although previous studies have shown that the XGBoost algorithm has superior performance in detecting fraudulent transactions on unbalanced data [8]. there are still a number of important challenges that have not been fully addressed. The main issue raised in this research is the effectiveness of the XGBoost model in dynamic real-time online transaction scenarios. These challenges include the risk of data leakage when sampling techniques are not applied properly [10], as well as changes in fraud patterns or concept drift that cause the model to use [11], as well as interpretability limitations that make prediction results difficult to explain to stakeholders, Thus, more investigation is required to examine the combination of data balancing strategies, machine learning algorithms such as XGBoost with data resampling techniques using SMOTE, as well as It has been demonstrated that using hyperparameter adjustment in XGBoost model optimization greatly enhances model performance [12] [13].

Given this context, the goal of this study is to reevaluate how well the XGBoost algorithm detects credit card fraud using a methodology that takes into account data balance, possible data leakage, and measures that accurately represent real-world operating conditions. Credit card transaction datasets from public repositories are used in this study. Data preprocessing, class imbalance analysis, data balancing methods like the SMOTE, XGBoost model training, hyperparameter tuning with GridSearchCV, and results evaluation using metrics for accuracy, precision, recall, and F1-score are all steps in the process.

Finding the best combination of data balancing techniques to greatly enhance XGBoost's performance without introducing overfitting or assessment bias is the anticipated outcome of this study. Furthermore, it is anticipated that this research will aid in the creation of a fraud detection system for the finance sector that is more precise, flexible, and trustworthy.

The novelty of this study lies in the systematic integration of SMOTE, XGBoost, and GridSearchCV into a unified pipeline designed to address extreme class imbalance in real-world credit card transactions. Unlike previous studies that focused solely on algorithm comparison or hyperparameter tuning, this research emphasizes reducing false negatives, which pose the greatest risk in fraud detection. By validating the model on a dataset with only 0.172% fraud cases, the study not only demonstrates superior performance metrics but also provides practical contributions for financial institutions in developing scalable, reliable, and adaptive fraud detection systems.

This essay is divided into multiple sections: The problem and introduction of the study are covered in Section 1; the materials, fraud detection techniques, and algorithms employed are covered in Section 2; the experimental results and model performance analysis of the study are presented in detail in Section 3; and the conclusions and recommendations for additional research are summed up in Section 4.

2. MATERIAL AND METHOD

This research is conducted for credit card fraud classification using the XGBoost algorithm so that it can be used as one of the decision supporters to determine fraud against credit card usage. The research methodology in this study broadly consists of 7 stages carried out. The following are the stages of research that can be seen in Figure 1.

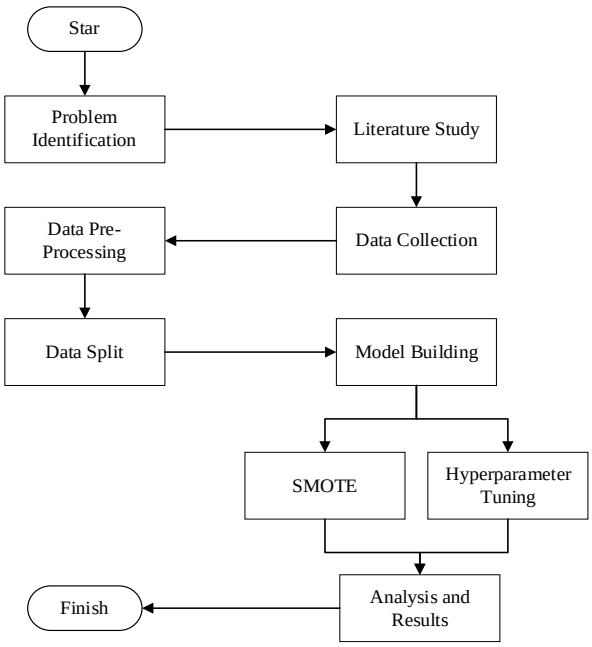

Figure 1. Research Methodology

2.1. Problem Identification

This research begins with the identification of the main problem, namely the imbalance of data on credit card transactions which causes the classification model to have difficulty detecting fraud accurately.

2.2. Literature Study

A review of previous research using machine learning algorithms, especially XGBoost, as well as approaches in handling unbalanced data such as SMOTE and Hyperparameter Tuning. The literature study process was carried out by accessing several websites such as Google Scholar, Academia, Science Direct, and other actual news sources related to credit card fraud.

2.3. Data Collection

The anonymized and publicly available "Credit Card Fraud Detection" dataset, which is kept up to date on Kaggle, provided the data used in this phase of the study. This specific dataset has been extensively cited in previous research, such as studies by Pozzolo et al. (2015) and Carcillo et al. (2018) [14][15][16]. The dataset comprises credit card transaction records from European users collected over two days in September 2013. Out of 284,807 total transactions, only 492 are identified as fraudulent, representing a highly skewed distribution where fraudulent activity makes up just 0.172% of the data.

It features 28 anonymized numerical variables produced via Principal Component Analysis (PCA), along with two original, untransformed fields: “Time” and “Amount.” The “Time” feature denotes the elapsed seconds since the initial transaction, while “Amount” reflects the transaction's monetary value. Due to confidentiality, the PCA-transformed variables (V1 to V28) are not explained in detail. The target variable, labeled “Class,” is binary, distinguishing legitimate transactions (0) from fraudulent ones (1).

For the purpose of building and assessing the model, the dataset is split at random into 80% training and 20% testing. The training set enables the model to discover transaction patterns, while the testing set evaluates the model's performance on new, unknown data to ensure accuracy, generalizability, and consistency in detecting fraud.

2.4. Data Pre-Processing

his stage includes data cleaning, removing attributes, and identifying and handling duplicate data and missing values. This process ensures that the data to be used is clean and ready for the next processing stage, such as removing unnecessary columns, namely the ID column [17]. This is followed by adjusting the target labels into two classes: fraud and non-fraud.

2.5. Data Split

To guarantee fair model evaluation and prevent overfitting, the dataset is split into training and test data at a preset ratio. Hold-out validation is an easy and reliable validation technique to estimate model accuracy, especially on large datasets [18].

2.6. Model Building

To classify transactions, the XGBoost model was developed. Given the large class imbalance, the SMOTE is used to create artificial examples of the minority class in the training dataset. The XGBoost model's hyperparameters were then optimized using GridSearchCV via a cross-validation procedure.

2.6.1. Machine Learning

The construction and application of statistical models and algorithms that allow computer systems to recognize patterns in data and make judgments or predictions without the need for explicit rule-based programming is known as machine learning in the context of artificial intelligence [19]. As a branch of artificial intelligence, machine learning focuses on equipping systems with the ability to automatically improve performance based on data-driven insights [20].

Supervised learning, one of the main methods in machine learning, involves training a model on a labeled dataset in which every input has a known output. Based on these instances, the model learns to map inputs to outputs. This method works very well for classification and regression applications.Common algorithms employed in supervised learning include neural networks, decision trees, and support vector machines (SVM). When the training data is accurate and representative of the broader context, supervised learning offers the significant advantage of delivering high prediction accuracy [20].

Second, Unsupervised learning is used when the data is unlabeled. The model attempts to find hidden structures or patterns in the data. This technique is often used for clustering and dimension reduction. Algorithms such as K-Means and PCA are examples of this approach. Although it does not require labeled data, the main challenge in unsupervised learning is the interpretation of the results which may not always be clear [21]. Thirdly, elements of supervised and unsupervised learning are combined in semi-supervised learning. This approach involves training the model with a large amount of unlabeled data in addition to a small amount of labeled data. When data labeling is expensive or time-consuming, this method works well. When the model uses unlabeled data instead of just labeled data, it can increase prediction accuracy [22].

Fourth, an agent that learns by interacting with its surroundings is a part of reinforcement learning. Depending on the acts performed, the agent obtains feedback through rewards or punishments. The primary aim is to develop a policy for actions that maximizes the overall cumulative reward. This method is frequently employed in the creation of autonomous systems, including computer games and robots [23].

2.6.2. Extreme Gradient Boosting (XGBoost)

XGBoost is a highly efficient and effective algorithm for solving large-scale problems. It uses a gradient tree reinforcement approach, which combines multiple decision trees to gradually correct the previous tree's errors until optimal accuracy is achieved [24]. XGBoost was used in this study because it is known to have superior performance in improving accuracy, training time efficiency, and computational resource utilization. In addition, this algorithm is also effective in minimizing the risk of overfitting. XGBoost works by incrementally building the model through a layered additive approach, and optimizing an objective function that includes both a prediction error measure and a penalty to control model complexity. The prediction calculation in XGBoost is described through a certain equation [25]. The form of the XGBoost algorithm equation is presented in Equations 1, 2, and 3.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),\ f_k \in \mathcal{F} \tag{1}$$

$$\mathcal{F} = \{ f(x) = w_{q(x)} \}, (q : \mathbb{R}^m \to T, w \in \mathbb{R}^T \tag{2}$$

$$L(\emptyset) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{3}$$

Description: $\hat{y}_i$ is the predicted value of the model for the i-th sample, f_k is the kth decision tree out of a total of K trees in the ensemble, $\mathcal{F}$ is the function space of all regression trees, $f(x) = w_{q(x)}$ is the function that generates the predicted value of the leaf weights (ω\omegaω) based on the tree structure ($q(x)$), $q : \mathbb{R}^m \to T$ is the mapping function from input features to leaf indices, $w \in \mathbb{R}^T$ is the weight of each leaf of the tree, T is the number of leaves in a tree, $l(\hat{y}_i, y_i)$ is the loss function between predicted and actual values, and $\Omega(f_k)$ is the complexity function of the kth tree (usually regularization).

2.6.3. Implementation of Synthetic Minority Over-Sampling Technique (SMOTE)

To mitigate data imbalance issues, SMOTE, an oversampling technique, is employed by producing synthetic data of the minority class using its nearest neighbors determined by Euclidean distance. Because the synthetic data is produced using pre-existing features, SMOTE increases the number of samples, making it appear more like the real data [26]. The form of the SMOTE equation is presented in Equation 4.

$$x_{new} = x_i + \lambda(x_j - x_i) \tag{4}$$

Description: $x_j - x_i$ are two examples drawn at random from the minority class, λ is a random number selected at intervals from a uniform distribution, x_{new} is the generated synthetic data.

2.6.4. Hyperparameter Tuning with Grid Search Cross-Validation

Values known as hyperparameters are established prior to the start of the model training procedure. Selecting the right machine learning algorithm is not enough to produce reliable prediction results; setting the right hyperparameter values is also required. Hyperparameter tuning involves modifying these values to determine the optimal set of hyperparameters that enhances model performance. By modifying the customizable parameters, this procedure seeks to optimize the model's accuracy, generalizability, or efficacy [27].

One technique for figuring out the ideal set of parameters for a model is Grid Search Cross-Validation (GridSearchCV). This approach evaluates every conceivable combination of factors in a methodical manner before validating each combination. If the range of each parameter's lowest and maximum values is preset, GridSearchCV will produce the best results [28]. The model will be optimized for the following seven parameters, as shown in Table 1, which are expected to enhance the model's classification performance.

Table 1. Best value result for hyperparameter

Hyperparameter	Purpose of hyperparameter
n_estimators	The number of trees used for the classification process
max_depth	Tree depth
eta (learning_rate)	Helps shorten the steps in model updating
subsample	Instance ratio of training data
colsample_bylevel	Ratio of training data used to create trees

The selection of hyperparameters in Table 1 refers to configurations that have been widely used in previous studies on credit card fraud detection [29]. These parameters were not arbitrarily chosen but adapted from existing literature as initial candidates. Furthermore, in this study, the parameter ranges were re-examined through experimental trials and optimized using GridSearchCV to ensure their suitability for the dataset's characteristics.

2.7. Analysis and Result

Binary classification is the predictive modeling method that is employed. Accordingly, there will be four different kinds of findings from the developed model: true negative (TN), false negative (FN), false positive (FP), and true positive (TP). The model's performance is assessed by analyzing the confusion matrix and comparing the outcomes before and after the data tuning procedure [30]. These outcomes, along with evaluation metrics like precision, accuracy, recall, and F1-score in equations 5, 6, 7, and 8, will be used to gauge how well the model separates fraudulent transactions from real transactions. These measures make sure that the algorithm used is the best one for the job at hand and show the model's overall strengths and shortcomings. Additionally, they exhibit model optimization, which entails modifying the fundamental algorithm parameter values in order to optimize performance [31].

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \text{ x } 100\% \tag{5}$$

$$\text{Precition} = \frac{\text{TP}}{\text{TP+FP}} \text{ x } 100\% \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \text{ x } 100\% \tag{7}$$

$$\text{F1} - \text{Score} = \frac{2\text{TP}}{2\text{TP+FN+FN}} \text{ x } 100\% \tag{8}$$

3. RESULTS AND DISCUSSION

The dataset, which includes 284,807 transactions, was acquired via Kaggle.com. 492 of these transactions were fraudulent, demonstrating a substantial disparity in class. Before performing the classification process, several stages of data processing were performed to prepare the data, namely: (1) Removing ID columns that have no effect on classification, (2) looking for missing values and duplicate data. However, models still struggle with data imbalance since they often anticipate the majority class (non-fraud) while ignoring the representation of the minority class (fraud). Many strategies, including undersampling and oversampling, have been investigated to lessen this issue. While oversampling may raise the danger of overfitting the model because of duplicate samples from the minority class, undersampling strategies run the risk of omitting crucial information from the majority class [32] [33]. By producing synthetic samples for the minority class, the SMOTE tackles the problem of class imbalance, particularly in cases of fraud [34].

Using a predetermined dataset, every experiment's performance was meticulously documented during the training and assessment stages. Cross-validation approaches were used to find and address problems like overfitting and underfitting that could negatively impact predicted accuracy. To find the best method for identifying fraud in digital payment systems, several models were created and contrasted. This comparison improved overall fraud detection capabilities by methodically adjusting hyperparameters using Scikit-learn's GridSearch functionality. In order to optimize accuracy, precision, and recall and eventually increase the model's effectiveness in accurately spotting fraudulent transactions, the hyperparameter optimization process was carried out with considerable thought.

3.1. Check the Distribution of Fraud and Non-fraud on Selected Variables

To select variables with a good distribution of fraud and non-fraud, this is done. The capacity of the feature to differentiate patterns among fraudulent and non-fraudulent transactions informs the choice of variables used for modeling and visualization. Certain features, including V1, V2, V4, V11, V12, V14, V17, and V18, were selected because they clearly differ in distribution between the two classes and contribute significantly to the classification process, as determined by feature importance analysis in the XGBoost model. The distribution of fraud and non-fraud on selected variables can be seen in Figure 2.

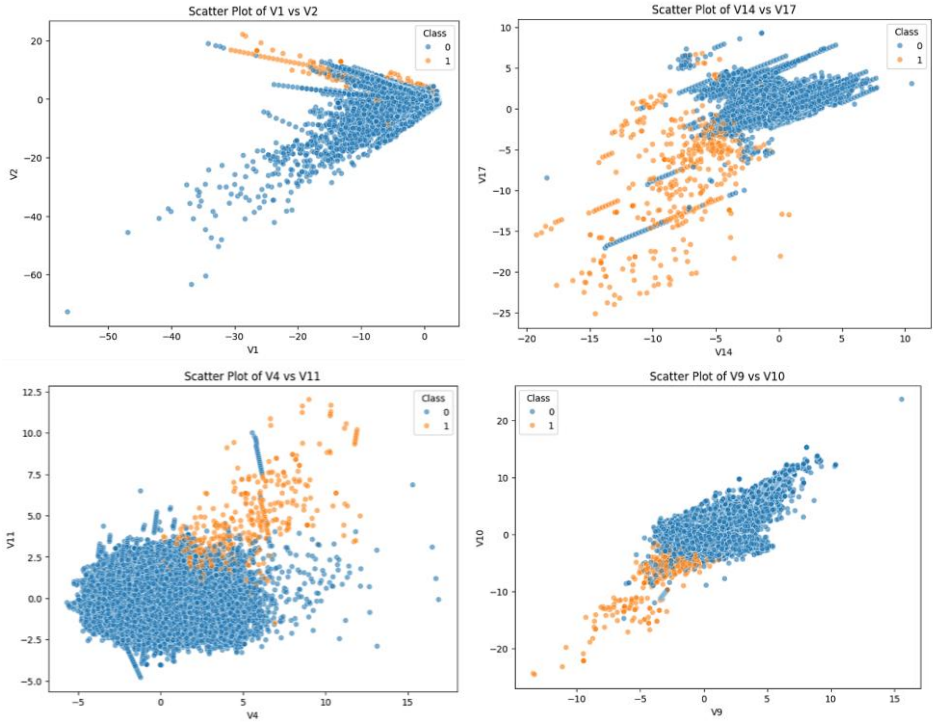

Figure 2. Distribution of Fraud and Non-Fraud on Selected Variables

3.2. Data Pre-Processing

Columns V1 to V28 are the result of PCA transformation and are already uniformly scaled so there is no need to re-standardize. However, the features 'Amount' and 'Time' are not included in this transformation, so they are standardized so that the value scale is in line with other features. Missing value checking showed that there were no missing values so imputation was not required, deletion of the 'Time' feature which was considered insignificant, and normalization of the 'Amount' feature using StandardScaler so that the scale of the features was balanced. Furthermore, duplicate data totaling 1,081 rows were removed to improve data

quality. After that, the dataset was divided into characteristics (X) and targets (Y), with a 'Class' label designating transactions that were fraudulent (1) and regular (0).

Following normalization, an 8:2 ratio was employed to separate the dataset was partitioned into training and testing subsets. In order to evaluate the model using the original test data and produce findings that are more reflective of the model's performance under actual circumstances, this division is completed prior to the use of sampling techniques.

3.3. Synthetic Minority Over-sampling Technique (SMOTE)

The training data's class distribution is displayed in Figure 3 both before and after the SMOTE approach is applied. Prior to SMOTE, the training data consisted of 227,845 samples, with a significantly unequal distribution between the 488 examples present in the fraud class and the 227,357 samples in the non-fraud class. After applying the SMOTE technique, synthetic samples are generated and incorporated into the minority class (fraud) dataset, increasing the total training data to 454,902 samples. With 227,451 samples for the fraud class and 227,451 samples for the non-fraud class, the class distribution is now balanced. This condition lessens the bias that typically results from class imbalance and enables the model to learn fraud transaction patterns more effectively. This outcome validates how well the SMOTE approach works to improve credit card fraud detection models' performance.

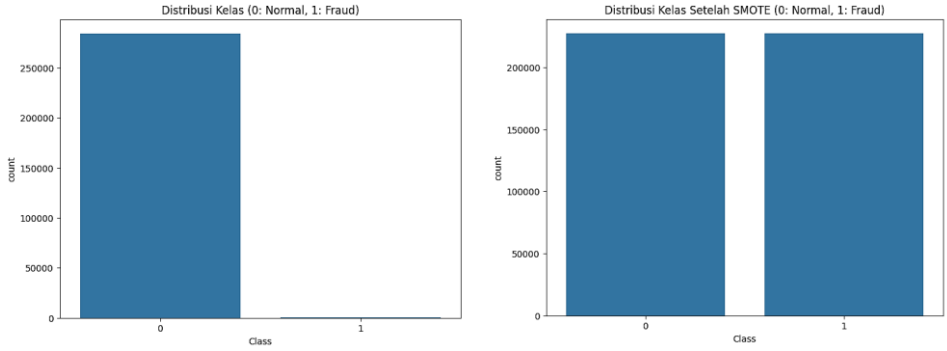

Figure 3. Distribution Dataset Before and After Using SMOTE

3.4. Test Results without using ScridsearchCV Hyperparameter Tuning

In the initial evaluation of classification employing the XGBoost algorithm, the entire training dataset is utilized to build the model, while the test dataset serves to measure the model's performance based on indicators such as accuracy, precision, and recall. This assessment is conducted without applying any hyperparameter tuning. Table 2 presents the performance outcomes obtained from the XGBoost model under default hyperparameter settings.

Table 2. XGBoot Classification Report without HyperParameter

Model	Precision	Recall	F1-score	Accuracy	AUC-ROC
XGBoost	0.69	0.87	0.77	1.00	0.975

The XGBoost model without hyperparameter adjustment yields an accuracy of 1.00 and an AUC-ROC of 0.975, according to Table 2. For the fraud class, the model's precision and recall are 0.69 and 0.87, respectively.

1. Precision of 0.69 indicates that out of all transactions predicted as fraud, 69% of them are true fraud transactions.
2. With a recall of 0.87, the model identified 87% of all real fraudulent transactions.

With 28431 non-fraud transactions correctly predicted (True Negatives), 9 non-fraud transactions inaccurately predicted as fraud (False Positives), 41 fraud transactions inaccurately wrongly classified as non-fraud (False Negatives), and 45 fraud transactions correctly predicted (True Positives), the Confusion Matrix (Figure 4) displays the detailed prediction distribution. This sheds light on the kinds of mistakes the model makes.

The link between True Positive Rate and False Positive Rate across a range of classification thresholds is depicted by the ROC curve (see Figure 5). The model demonstrates a strong discriminatory capability between legitimate and fraudulent transactions, as evidenced by an AUC-ROC score of 0.975. Figure 5 depicts this trade-off at different threshold settings, confirming the model's overall effectiveness in distinguishing fraudulent cases.

Despite these promising results, particularly following the prior data balancing efforts, further enhancement through hyperparameter optimization is necessary. The primary goal the objective of this tuning phase is to increase the model's precision by minimizing the incidence of False Positives transactions mistakenly classified as fraudulent while sustaining or elevating the model's overall performance metrics.

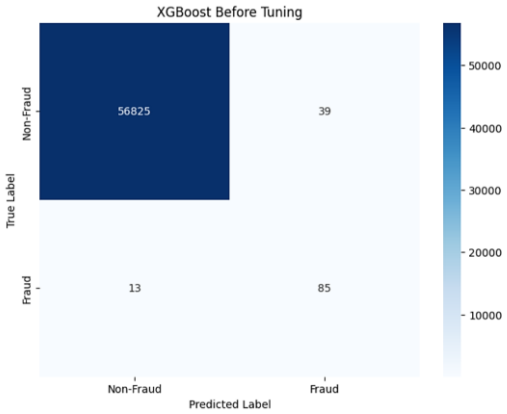

Figure 4. Confusion Matrix of XGBoost Model Without Hyperparameter Tuning

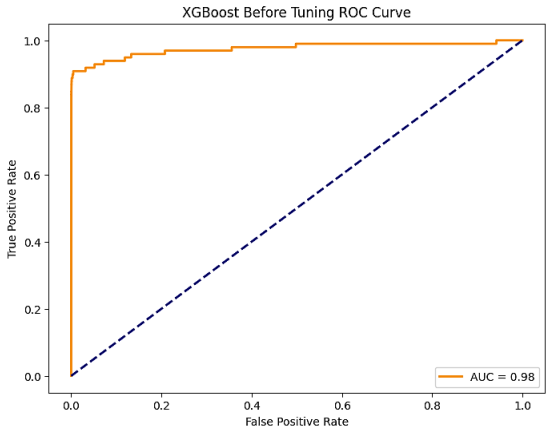

Figure 5. ROC Curve of XGBoost Model Without Hyperparameter Tuning

3.5. Primary Testing using ScridsearchCV Hyperparameter Tuning

Five important parameters that are thought to enhance the model's effectiveness in classifying fraudulent transactions are hyperparameter tuned in this study. The tuning process utilizes the Grid Search Cross-Validation (GridSearchCV) method, In order to identify the ideal configuration, this methodical and thorough approach investigates several combinations of predetermined parameter values. [35].

The selection of the best hyperparameter configuration is based on the highest cross-validation accuracy value obtained from the evaluation of a number of candidate parameters. The tuning results obtained are presented in Table 3.

Table 3. XGBoost Grid search Range

Hyperparameter	Grid Search Values	Value Hyperparameter Best
'n_estimators'	100. 200, 300	200
'max_depth'	3, 5, 7	5
'eta (learning_rate)'	0.01, 0.1, 0.2	0.1
'subsample'	0.7, 0.8, 0.9	0.9
'colsample_bylevel'	0.7, 0.8, 0.9	0.8

The ideal parameter combination is found based on the outcomes of hyperparameter tuning with GridSearchCV and the AUC-ROC assessment tool. By striking a balance between the model's generalizability and complexity, these parameters lower the possibility of overfitting and enhance the identification of fraudulent transactions. This is demonstrated by the strong AUC-ROC value and the well-balanced recall and precision.

Table 4. XGBoost Classification Report with Hyperparameter Tuning

Model	Precision	Recall	F1-score	Accuracy	AUC-ROC
XGBoost	0.81	0.85	0.83	1.00	0.979

The XGBoost model's performance evaluation results following hyperparameter adjustment with GridSearchCV are displayed in Table 4. The accuracy value of 0.81, recall of 0.85, and F1-score of 0.83 that the model produced show a balance between precision and the model's ability to detect affirmative classes (fraud). The accuracy rating of 1.00 shows how well the model can classify the data overall. Additionally, the AUC-ROC score of 0.979 indicates the model's exceptionally excellent discriminative capacity to distinguish between fraud and non-fraud classes.

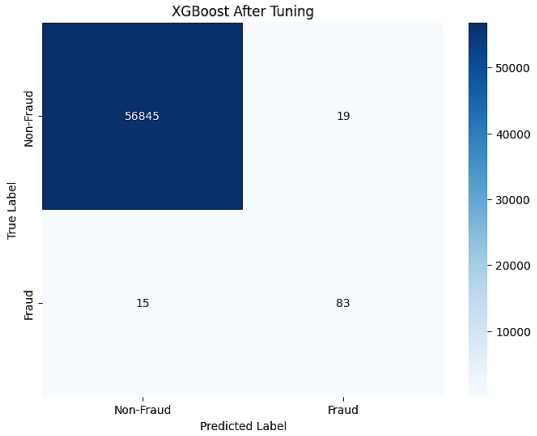

Figure 6. Confusion Matrix Model XGBoost with Hyperparameter Tuning

With hyperparameter adjustment, the XGBoost model's confusion matrix is optimized, as shown in Figure 6. Just 19 non-fraud transactions were incorrectly projected to be fraud (False Positive), compared to 56,845 non-fraud transactions that were accurately categorized as non-fraud (True Negative). Furthermore, 83 fraudulent transactions were accurately identified by the model (True Positive), although there were still 15 fraudulent transactions that were not detected and classified as non-fraud (False Negative). This distribution shows that the model not only has a high accuracy rate in recognizing legitimate transactions, but is also quite sensitive in detecting anomalies in the form of fraud, which is naturally rare but has a large impact. The balance among type I and type II errors indicates that the model has achieved optimal performance, making it suitable for use in fraud detection systems that demand high accuracy and reliability.

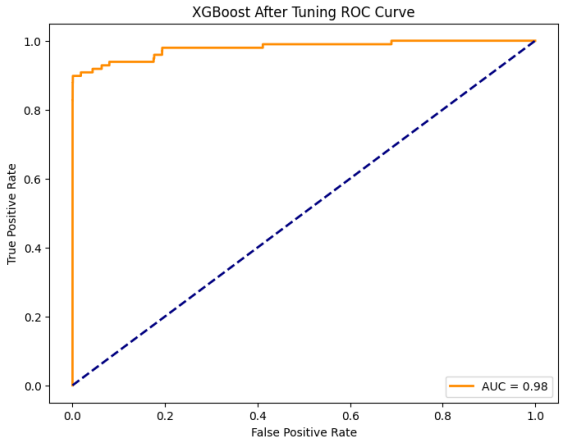

Figure 7. ROC Curve of XGBoost Model Without Hyperparameter Tuning

The XGBoost model's Receiver Operating Characteristic (ROC) curve following hyperparameter adjustment is shown in Figure 7. The correlation between the true positive rate (TPR) and the false positive rate (FPR) across a range of categorization thresholds is depicted by this curve. The model's exceptional performance in class distinction is indicated by its Area Under the Curve (AUC) score of 0.98. Given that this number is near to 1, there is a very high chance that the model will accurately categorize one fraudulent transaction and one non-fraudulent transaction chosen at random.

Additionally, the ROC curve near the graph's upper left corner demonstrates the model's excellent sensitivity without compromising specificity, making it very suitable for fraud detection systems that require high accuracy and quick and precise response to anomalies.

3.6. Discussion

This study demonstrates that the integration of SMOTE and GridSearchCV into the XGBoost model significantly improves fraud detection performance. The model achieved a precision of 0.81, recall of 0.85, F1-score of 0.83, and an AUC-ROC of 0.979. These results indicate that the model not only excels in identifying fraudulent transactions but also maintains robustness in classifying legitimate transactions, which is essential for minimizing disruptions in real-world financial operations. The balanced performance between precision and recall reflects the model's capability to reduce false positives while at the same time minimizing false negatives, a trade-off that is particularly critical in fraud detection.

When compared to previous studies, this research shows superior performance. Asnawi et al. (2025), for example, reported an F1-score of 0.77 and an AUC of 0.97, which are slightly lower than the results achieved in this study. Similar to Asnawi et al. (2025), this study also employed the Kaggle Credit Card Fraud Detection dataset, normalization preprocessing, and SMOTE oversampling before training. Furthermore, GridSearchCV was utilized to optimize XGBoost hyperparameters, ensuring more robust generalization performance. The higher F1-score obtained here highlights the advantage of systematically combining SMOTE and hyperparameter optimization, leading to a more reliable model.

Gupta et al. (2022) also demonstrated the effectiveness of oversampling strategies with XGBoost, achieving an F1-score close to 0.998. However, their approach did not emphasize hyperparameter optimization, which may limit generalization in dynamic fraud detection environments. Likewise, the work of Pozzolo et al. (2015) highlighted the extreme imbalance in fraud datasets and proposed resampling methods, but their study did not combine systematic hyperparameter tuning with ensemble methods. By contrast, the present study emphasizes the importance of reducing false negatives, as undetected fraud cases can lead to substantial financial losses for banks and e-commerce providers [36][37][38].

The implications of these findings are significant for financial institutions, e-commerce platforms, and regulators. A model that achieves high recall ensures that fraudulent transactions are less likely to pass undetected, protecting institutions from financial damage and preserving consumer trust. At the same time, maintaining relatively high precision reduces the number of legitimate transactions incorrectly flagged as fraudulent, which is crucial for ensuring a smooth customer experience. This balance is what makes the proposed approach suitable for real-time fraud monitoring systems that demand both accuracy and reliability.

In addition, the study reinforces the limitations of traditional rule-based fraud detection systems. Unlike static systems that struggle to adapt to evolving fraud patterns, the combination of XGBoost with SMOTE and GridSearchCV provides a flexible and adaptive mechanism that can capture subtle and rare fraud behaviors. This adaptability is essential in the context of digital transactions, where fraud tactics evolve rapidly, and detection systems must be able to respond dynamically.

Despite these contributions, the study also has limitations. The dataset used originates from European credit card transactions in 2013, which may not fully represent the latest transaction behaviors or fraud patterns, especially in regions such as Indonesia where digital payments are growing rapidly. Future research should test the model using more recent and localized datasets to validate its robustness across different environments. Additionally, the integration of deep learning architectures or ensemble methods could further enhance detection capabilities, particularly in handling large-scale real-time transactions [36].

4. CONCLUSION

This study set out to enhance credit card fraud detection by systematically integrating SMOTE, GridSearchCV, and XGBoost into a unified pipeline. The results demonstrated that this approach effectively addressed extreme class imbalance and significantly improved model performance, achieving a precision of 0.81, recall of 0.85, F1-score of 0.83, and an AUC-ROC of 0.979. These outcomes confirm that the proposed model is capable of reducing false negatives while maintaining relatively high precision, making it both accurate and practical for deployment in financial institutions' fraud monitoring systems.

The main contribution of this research lies in demonstrating that combining data balancing and hyperparameter optimization can yield a more robust and adaptive fraud detection system compared to prior works that focused on algorithm comparison or oversampling alone. In addition, incorporating feature importance–based selection before model training further enhances the system's efficiency in recognizing hidden fraud patterns.

Nevertheless, this study has limitations. The dataset used originates from European credit card transactions in 2013, which may not fully capture current fraud patterns or transaction behaviors, particularly in rapidly growing digital economies such as Indonesia. Moreover, the evaluation was limited to a single dataset and did not test real-time scalability under production-level transaction volumes.

Future research should therefore consider validating the model on more recent and localized datasets to strengthen its external validity. Expanding the framework by integrating deep learning architectures, ensemble approaches, or hybrid anomaly detection methods could further enhance performance in handling complex fraud scenarios. Exploring real-time deployment strategies, such as online learning or streaming data analysis, is also recommended to ensure the model's applicability in practical, large-scale financial systems.

REFERENCES

[1] TransUnion, "2023 state of omnichannel fraud:Trends and strategies for enabling trusted commerce," *Transunion*, 2023.

[2] I. Mekterović, M. Karan, D. Pintar, and L. Brkić, "Credit card fraud detection in card-not-present transactions: Where to invest?," *Applied Sciences (Switzerland)*, vol. 11, no. 15, 2021, doi: 10.3390/app11156766.

[3] K. F. Mauladi, I. M. L. M. Jaya, and M. A. Esquivias, "Exploring the Link between Cashless Society and Cybercrime in Indonesia," *Journal of Telecommunications and the Digital Economy*, vol. 10, no. 3, pp. 58–76, 2022, doi: 10.18080/jtde.v10n3.533.

[4] Y. Dewi, H. Suharman, P. S. Koeswayo, and N. D. Tanzil, "Factors influencing the effectiveness of credit card fraud prevention in Indonesian issuing banks," *Banks and Bank Systems*, vol. 18, no. 3, pp. 44–60, 2023, doi: 10.21511/bbs.18(4).2023.05.

[5] I. Y. Hafez, A. Y. Hafez, A. Saleh, A. A. Abd El-Mageed, and A. A. Abohany, "A systematic review of AI-enhanced techniques in credit card fraud detection," *J Big Data*, vol. 12, no. 1, 2025, doi: 10.1186/s40537-024-01048-8.

[6] J. J. Assabil and I. C. Obagbuwa, "Credit Card Fraud Detection Using Machine Learning Algorithms : A Comparative Study of Six Models," *International Journal of Intelligent Systems And Applications In Engineering*, vol. 12, pp. 862–875, 2024.

[7] N. T. Ali, S. J. Hasan, A. Ghandour, and Z. S. Al-Hchimy, "Improving credit card fraud detection using machine learning and GAN technology," *BIO Web Conf*, vol. 97, pp. 1–18, 2024, doi: 10.1051/bioconf/20249700076.

[8] 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, 2022, doi: 10.1016/j.procs.2023.01.231.

[9] S. Joses and A. Saikhu, "Enhancing XGBoost and CatBoost Methods for Diagnosing Parkinson's Disease Through the Integration of SMOTE and Feature Selection Techniques," in *2024 8th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia: IEEE, 2024, pp. 487–492. doi: 10.1109/ICITISEE63424.2024.10729906.

[10] S. Kabane, "Impact of Sampling Techniques and Data Leakage on XGBoost Performance in Credit Card Fraud Detection," *Journal of Cornell University*, pp. 1–19, 2024.

[11] G. K. Kulatilleke, "Challenges and Complexities in Machine Learning based Credit Card Fraud Detection," *Journal of Cornell University*, pp. 1–17, 2022.

[12] S. B. Punuri *et al.*, "Efficient Net-XGBoost: An Implementation for Facial Emotion Recognition Using Transfer Learning," *Mathematics*, vol. 11, no. 3, pp. 1–24, 2023, doi: 10.3390/math11030776.

[13] A. A. Syahputra and R. E. Saputro, "Application of the XGBoost Model with Hyperparameter Tuning for Industry Classification for Job Applicants," *Sinkron*, vol. 8, no. 3, pp. 1920–1931, 2024, doi: 10.33395/sinkron.v8i3.13840.

[14] A. Dal Pozzolo, O. Caelen, Y. A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst Appl*, vol. 41, no. 10, pp. 4915–4928, 2014, doi: 10.1016/j.eswa.2014.02.026.

[15] F. Carcillo, A. Dal Pozzolo, Y. A. Le Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, "SCARFF: A scalable framework for streaming credit card fraud detection with spark," *Information Fusion*, vol. 41, no. May 2019, pp. 182–194, 2018, doi: 10.1016/j.inffus.2017.09.005.

[16] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," *Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015*, no. December, pp. 159–166, 2015, doi: 10.1109/SSCI.2015.33.

[17] S. Wang *et al.*, "Advances in Data Preprocessing for Biomedical Data Fusion: An Overview of the Methods, Challenges, and Prospects Shuihua," *Science Direct*, 2021.

[18] D. Pramana and M. Mustakim, "Prediksi Status Penanganan Pasien Covid-19 dengan Algoritma Naïve Bayes Classifier di Provinsi Riau," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 3, no. 2, p. 202, 2021, doi: 10.30865/json.v3i2.3570.

[19] E. F. Okagbue *et al.*, "A comprehensive overview of artificial intelligence and machine learning in education pedagogy: 21 Years (2000–2021) of research indexed in the scopus database," *Social Sciences and Humanities Open*, vol. 8, no. 1, p. 100655, 2023, doi: 10.1016/j.ssaho.2023.100655.

[20] Y. Matsuo *et al.*, "Machine Learning: A Review of Learning Types," *Neural Networks*, vol. 7, no. 1, pp. 267–275, 2020, doi: 10.20944/preprints202007.0230.v1.

[21] N. M. Noor Mathivanan, N. A. MdGhani, and R. M. Janor, "A comparative study on dimensionality reduction between principal component analysis and k-means clustering," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 752–758, 2019, doi: 10.11591/ijeecs.v16.i2.pp752-758.

[22] J. M. Ramírez-Sanz, J. A. Maestro-Prieto, Á. Arnaiz-González, and A. Bustillo, "Semi-supervised learning for industrial fault detection and diagnosis: A systemic review," *ISA Trans*, vol. 143, no. August, pp. 255–270, 2023, doi: 10.1016/j.isatra.2023.09.027.

[23] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, *An introduction to deep reinforcement learning*, vol. 11, no. 3–4. 2018. doi: 10.1561/2200000071.

[24] S. Sankar, A. Potti, G. Naga Chandrika, and S. Ramasubbareddy, "Thyroid Disease Prediction Using XGBoost Algorithms," *Journal of Mobile Multimedia*, vol. 18, no. 3, pp. 917–934, 2022, doi: 10.13052/jmm1550-4646.18322.

[25] I. Muslim Karo Karo, "Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan," *Journal of Software Engineering, Information and Communication Technology*, vol. 1, no. 1, pp. 11–18, 2020.

[26] R. Chen *et al.*, "A study on predicting the length of hospital stay for Chinese patients with ischemic stroke based on the XGBoost algorithm," *BMC Med Inform Decis Mak*, vol. 23, no. 1, pp. 1–10, 2023, doi: 10.1186/s12911-023-02140-4.

[27] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019, doi: 10.11989/JEST.1674-862X.80904120.

[28] M. M. Ramadhan, I. S. Sitanggang, F. R. Nasution, and A. Ghifari, "Parameter Tuning in Random Forest Based on Grid Search Method for Gender Classification Based on Voice Frequency," *DEStech Transactions on Computer Science and Engineering*, no. cece, 2017, doi: 10.12783/dtcse/cece2017/14611.

[29] E. Pujo, A. Akhmad, K. Adi, and A. P. Widodo, "Enhancing the Accuracy of Airline Review Classification Using SMOTE and Grid Search with Cross Validation for Hyperparameter Tuning," 2024. [Online]. Available: https://www.jisem-journal.com/

[30] O. Shobayo, O. Zachariah, M. O. Odusami, and B. Ogunleye, "Prediction of Stroke Disease with Demographic and Behavioural Data Using Random Forest Algorithm," *Analytics*, vol. 2, no. 3, pp. 604–617, 2023, doi: 10.3390/analytics2030034.

[31] V. Chang, B. Ali, L. Golightly, M. A. Ganatra, and M. Mohamed, "Investigating Credit Card Payment Fraud with Detection Methods Using Advanced Machine Learning," *Information (Switzerland)*, vol. 15, no. 8, pp. 1–20, 2024, doi: 10.3390/info15080478.

[32] H. C. Du, L. Lv, H. Wang, and A. Guo, "A novel method for detecting credit card fraud problems," *PLoS One*, vol. 19, no. 3 March, pp. 1–26, 2024, doi: 10.1371/journal.pone.0294537.

[33] T. Majumder, "Financial Fraud Detection for Credit Card Using XGBoost & SMOTE," *Nanotechnol Percept*, pp. 32–50, doi: 10.62441/nano-ntp.vi.3425.

[34] S. Rabbani, D. Safitri, N. Rahmadhani, A. A. F. Sani, and M. K. Anam, "Perbandingan Evaluasi Kernel SVM untuk Klasifikasi Sentimen dalam Analisis Kenaikan Harga BBM," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 153–160, 2023, doi: 10.57152/malcom.v3i2.897.

[35] D. Trisanto, N. Rismawati, M. F. Mulya, and F. I. Kurniadi, "Modified Focal Loss in Imbalanced XGBoost for Credit Card Fraud Detection," *International Journal of intelligent Engineering & Systems*, vol. 14, no. 4, 2021, doi: 10.1080/10462938809365891.

[36] M. Fuat Asnawi, N. Fitriyanto, and M. Agoeng Pamoengkas, "THE APPLICATION OF XGBOOST CLASSIFICATION FOR FRAUD DETECTION IN CREDIT CARD TRANSACTIONS," *Clean Energy and Smart Technology*, vol. 03, p. 2, 2025, doi: 10.58641/e-ISSN.

[37] 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, doi: 10.1016/j.procs.2023.01.231.