Handling Imbalanced Classification with XGBoost and Imbalanced-learn

1st Ziqiqi Wang

***Abstract*—This study tackles extreme class imbalance in credit card fraud detection using data resampling (RandomOverSampler/SMOTE) and XGBoost class weighting. Evaluated via F1 and PR-AUC, RandomOverSampler optimizes minority recognition, while XGBoost weighting balances efficiency and performance. Ensemble methods boost recall but reduce precision. The work offers a reusable framework, advocating class-sensitive metrics and context-driven strategy selection for imbalanced classification.**

***Index Terms*—XGBoost, Imbalanced, Fraud**

I. Introduction

This study addresses the issue of class imbalance in fields such as fraud detection, where there is a high risk of misclassification in minority classes and traditional models are dominated by majority classes. An efficient classification framework is constructed through data level resampling (RandomOverSampler/SMOTE) and model level optimization (XGBoost class weight adjustment), and evaluated using class sensitive indicators after feature processing. Research and compare sample augmentation and loss function weighting methods to verify their effectiveness in improving minority class recognition and reveal ensemble strategy bias. The core objective is to evaluate the effectiveness of the technology, optimize the strategy, and provide a reusable methodology to provide engineering guidance for XGBoost optimization and precise minority class recognition in actual imbalanced scenarios.

II. Method

A. Problem Formulation

This study addresses a highly imbalanced binary classification task aiming to identify credit card fraud transactions (minority class, 0.172% of 284,807 transactions, class imbalance ratio 579:1). The dataset comprises European cardholder transactions from September 2013, with input features including 28 PCA-transformed principal components (V1–V28), transaction time ("Time"), and amount ("Amount"). Given a feature matrix $X \in \mathbb{R}^{n\times 30}$, the task is to predict binary labels $y \in \{0, 1\}$, with core objective being the correct identification of minority-class samples.

B. Data Preprocessing

1) Data Partitioning and Stratification: Stratified sampling is employed to divide the dataset into a training set (227,845 samples) and a test set (56,962 samples) at an 8:2 ratio. This ensures that the proportion of the minority class in both the training and test sets remains consistent (approximately 0.172%), thereby avoiding sampling bias that could affect model evaluation.

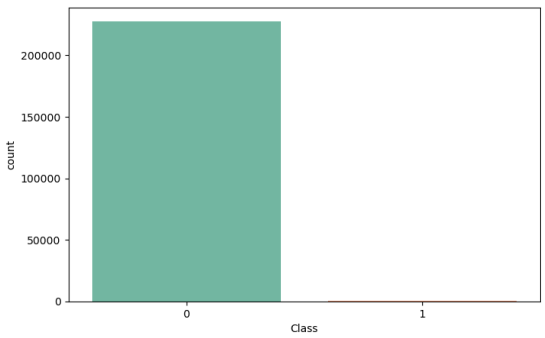

Fig. 1. Distribution of the target variable

C. Feature Engineering

1) Time Feature Handling: The original time difference feature "Time" is removed because it only reflects the transaction order and has no direct association with fraud patterns.

2) Amount Standardization: Z - score standardization is applied to the "Amount" feature, generating "Amount_scaled" to eliminate the influence of dimensionality and avoid interference from differences in numerical ranges during model training.

D. Imbalanced Data Strategies

1) Data - level Resampling Techniques (Imbalanced - learn): Given the extreme imbalance in the training set (394 minority class instances and 227,451 majority class instances, with an imbalance ratio of 577:1), the following oversampling methods are employed to generate a balanced dataset:

1) RandomOverSampler This method randomly duplicates the minority class samples until the number of minority class samples is equal to that of the majority class (227,451 instances), directly increasing the quantity of minority class samples.
2) SMOTE Based on the k - nearest neighbors of the minority class samples (default $k = 5$), this method generates synthetic samples. By interpolating in the feature space, it enhances data diversity and mitigates the risk of overfitting.
3) ADASYN This method dynamically generates synthetic samples according to the sample density, focusing on

the hard - to - learn minority class instances and optimizing the classification performance near the decision boundary.

2) Model - level Weight Adjustment (XGBoost Optimization): The "scale_pos_weight" parameter in XGBoost is used to adjust the class weights. The calculation formula is:

$$\text{scale\_pos\_weight} = \frac{\text{Number of majority class samples}}{\text{Number of minority class samples}}$$

This parameter guides the model to pay more attention to the minority class samples by increasing the loss weight of misclassifying the minority class.

3) Ensemble Learning Strategy (BalancedBaggingClassifier) : A balanced bagging ensemble method is adopted. Each base learner (XGBoost) is trained on a balanced subset obtained through bootstrapping, which reduces the dominant effect of the majority - class samples.

Let param_grid be a set of hyperparameter combinations. Then,

$$\text{param\_grid} = \left\{ \begin{array}{l} \text{estimator\_\_learning\_rate} \in \{0.01, 0.1, 0.2\} \\ \text{estimator\_\_n\_estimators} \in \{50, 100, 200\} \\ \text{estimator\_\_max\_depth} \in \{3, 5, 7\} \\ \text{n\_estimators} \in \{10, 20, 30\} \end{array} \right\} \quad (1)$$

E. Evaluation Metrics for Models

1) Class-Sensitive Evaluation Metrics:

1) F1 Score
   The F1 score harmonizes precision and recall to balance the prediction performance of the two classes. The formula is given by:
$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2)$$
2) Recall
   Recall measures the ability to correctly identify fraud transactions. It is defined as:
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$
3) PR-AUC
   The area under the precision-recall curve (PR-AUC) is sensitive to imbalanced data and outperforms traditional accuracy metrics (e.g., accuracy from the confusion matrix is meaningless in this scenario).

III. Results

The experiment is based on the credit card fraud detection dataset. After stratified partitioning, the training set contains 227,845 samples (394 samples of the minority class, with an imbalance ratio of 577:1), and the test set contains 56,962 samples (98 samples of the minority class). Data pre - processing includes removing the time feature "Time" and standardizing the transaction amount "Amount". The model configurations are as follows:

1) Baseline Model
   XGBoost with default parameters, directly inputting the original training set.
2) Data - level Methods
   XGBoost is trained after generating balanced datasets (with a ratio of 1:1) using RandomOverSampler, SMOTE, and ADASYN.
3) Model - level Method
   XGBoost adjusts the class weights through "scale_pos_weight = 577".
4) Ensemble Method
   BalancedBaggingClassifier wraps XGBoost, and hyperparameters are optimized through grid search.

A. Key Performance Metrics

Table 1 summarizes the class-sensitive metrics (F1, minority class recall, PR-AUC) of different strategies on the test set:

TABLE I
CLASS-SENSITIVE EVALUATION RESULTS OF DIFFERENT MODELS

Method	F1	Recall (%)	PR-AUC
XGBoost (Baseline)	0.834	79.59	0.698
RandomOverSampler	0.875	85.71	0.766
SMOTE	0.802	88.78	0.649
ADASYN	0.758	84.69	0.581
XGBoost (Class Weighted)	0.865	84.69	0.748
BalancedBaggingClassifier	0.162	91.84	0.082
Best BalancedBagging	0.162	90.82	0.081

B. Key Findings

The experiment evaluated the effectiveness of different methods in addressing class imbalance in credit card fraud detection. Among data - level oversampling methods, RandomOverSampler performed best in F1 and PR - AUC, SMOTE had the highest recall but a lower PR - AUC, and ADASYN had the weakest performance. Model - level weight adjustment increased the PR - AUC of XGBoost to a level close to that of RandomOverSampler at a lower cost. The ensemble method had a high recall but extremely low F1 and PR - AUC. The baseline model had lower metrics than all optimization methods, highlighting the necessity of dealing with class imbalance.

IV. Conclusion

This study evaluates extreme class imbalance (0.172% minority class) in credit card fraud detection, comparing data-level resampling (RandomOverSampler, SMOTE), model-level weight adjustment, and ensemble methods. RandomOverSampler achieved optimal F1 (0.875) and PR-AUC (0.766), while SMOTE improved minority recall to 88.78% but posed noise risks in complex data. Model-level adjustment balanced performance (PR-AUC = 0.748) with minimal overhead, suitable for deployment. Ensemble strategies like BalancedBaggingClassifier showed high minority recall (91.84%) but sacrificed F1 (0.162) due to majority class neglect. Baseline results emphasized the necessity of addressing imbalance. Practical guidance: use class-sensitive metrics (F1, PR-AUC), prioritize resampling/weighting based on resources, and future work may explore multi-class imbalance and ensemble optimization with diverse learners or domain knowledge integration.