Zahra: Bulletin Big data, Data Science and Artificial Intelligence
Vol. 3, No. 2, [December 2024], pp. [83–96]
E-ISSN 2830-7984

Research Article Open Access (CC–BY-SA)

Cost-Sensitive XGBoost with Adaptive Decision Threshold Optimization for Fraud Detection in Highly Imbalanced Digital Transactions

Putrama Alkhairi[1*], Imam Saputra[2], Abdul Karim[3], Erfan Babaee Tırkolaee[4]
[1] *Informatics Management, STIKOM Tunas Bangsa, Pematangsiantar, North Sumatra, Indonesia*
[2] *Management Program, Sekolah Tinggi Ilmu Manajemen Sukma, Medan, Indonesia*
[3] *Information Technology Study Program, Faculty of Science and Technology, Universitas Labuhanbatu, Indonesia*
[4] *İstinye Üniversitesi, Istanbul, Turki*

[1] *putrama@amiktunasbangsa.ac.id;* [2] *saputraimam69@gmail.com;* [3] *Abdulkarim@ulb.ac.id;* [4] *erfan.babaee@engr.istinye.edu.tr*
** Corresponding author*

Abstract

Detecting fraudulent card transactions under extreme class imbalance requires models whose missed-fraud costs and false alarms are evaluated separately. This study assesses cost-sensitive XGBoost with Bayesian optimization and an intended adaptive decision threshold on 284,807 anonymized transactions. A chronological partition provides 170,884 training, 56,961 development, and 56,962 held-out testing records. The comparison includes unweighted XGBoost, class weighting, threshold-only adjustment, their combination, random search, and the proposed Bayesian procedure. Using hypothetical costs of 20 per false negative and one per false positive, the Bayesian model reduces cost per transaction from 0.007795 to 0.006759, a 13.29% relative reduction. Fraud recall increases from 70.67% to 74.67%, while precision falls from 92.98% to 91.80% and false alarms rise from four to five. All reported models retain a decision cutoff of 0.5, preventing any independent claim of improvement from adaptive threshold adjustment. The proposed model performs best on the assumed cost metric in this held-out comparison, but the result remains contingent on the cost ratio, anonymized history, and single temporal split.

Keywords: Credit card fraud; Cost-sensitive learning; XGBoost; Bayesian optimization; Decision threshold.

1. Introduction

Financial transaction fraud detection requires methods that can distinguish rare malicious activity from a large population of legitimate payments. Reviews of fraud modeling document substantial diversity in transaction representations, data sources, and evaluation metrics [1]. A dedicated credit-card survey identifies class imbalance and temporal dependency as persistent barriers to generalization [2]. Comparative work on financial fraud emphasizes that high aggregate accuracy can conceal missed fraudulent events [3]. Recent XGBoost studies show that changing the training loss can alter minority-class detection without guaranteeing favorable operational decisions [4]. Imbalance-oriented research recommends evaluating recall and precision together with the consequences of false alerts [5]. The methodological question is therefore not whether one classifier achieves the highest accuracy, but whether a documented decision policy lowers the cost of the errors it makes.

Cost-sensitive learning and decision-threshold selection address related but distinct parts of this problem. A cost-sensitive credit-card study considers both class-weighted training and post-training threshold optimization [6]. Research on dynamic financial anomaly thresholds illustrates that an operating cutoff can be changed without refitting a classifier [7]. Classifier calibration studies distinguish probabilistic reliability from the discrimination and thresholding of risk scores [8]. Bayesian search can improve tree-model configurations, but its benefit depends on search domains, computational budgets, and evaluation design [9]. A direct study of Bayesian XGBoost fraud detection makes the choice of optimizer particularly relevant [10]. Comparative threshold-based ensembles illustrate why component-level controls are needed to isolate a genuine decision-rule contribution [11]. Reliability must also be interpreted through both ROC and precision-recall analyses because the two metrics respond differently to prevalence [12].

This study examines XGBoost training costs and decision thresholds using an original-data chronological benchmark. A survey of recent deep-learning approaches confirms that complex models should not automatically be preferred to interpretable controls [13]. An ensemble study of 284,807 credit-card transactions highlights the practical tension between missed fraud and false alarms [14]. Comparative cost-aware evaluation shows why ranking, false-positive burden, and computational efficiency should be reported simultaneously [15] . Temporal data can also change after deployment, making a fixed historical test insufficient to establish long-term stability [16]. The present experiment compares the unweighted classifier, weighting-only and cutoff-only alternatives, matched-budget random search, and Bayesian cost-sensitive optimization. Its primary outcome is hypothetical asymmetric classification cost, supported by confusion counts, discrimination, calibration diagnostics, and runtime. An observed improvement is interpreted as benchmark evidence rather than proof of financial savings or an adaptive threshold mechanism.

2. Method

2.1. Related Work

Recent work distinguishes learning from imbalanced data from the subsequent decision to initiate a fraud investigation. Salekshahrezaee et al. examined feature extraction and resampling strategies for transaction classifiers [17]. Breskuvienė and Dzemyda considered feature selection under pronounced imbalance [18]. Kennedy et al. evaluated synthesized labels for rare fraudulent transactions [19]. Later work examined unsupervised feature selection and labeling for similar fraud data [20]. Walauskis and Khoshgoftaar connected SHAP-based feature reduction to unsupervised fraud labeling [21]. These studies broaden the possible preprocessing choices, but do not demonstrate that a changed decision threshold necessarily improves a fixed XGBoost model.

Tree ensembles, mixed model families, and optimization remain prominent competitors in fraud prediction. Almubark evaluated an ensemble approach with two-stage thresholding [22]. Ileberi and Sun investigated a hybrid deep-learning ensemble for credit-card fraud [23]. Huang et al. analyzed clustering-based undersampling together with neural network representations [24]. Salem et al. introduced a metaheuristic selection approach to credit-card fraud detection [25]. Atban et al. considered metaheuristic feature selection with a variational quantum classifier [26]. Alrasheedi compared several machine-learning families for card-transaction fraud [27]. Their heterogeneous architectures motivate including a strong conventional tuning comparator rather than assessing the proposed learner against an untuned default only.

Risk evaluation also involves explainability, uncertainty, and operating conditions. Awad et al. studied ensemble explanations across credit-card fraud benchmarks [28]. Çavdar and Bozanta examined model explainability and feature engineering in actual transactions [29]. Nwafor et al. addressed explainable modeling for automated credit decisions, emphasizing accountability in financial applications [30]. Ghalwash et al. tested augmentation and voting approaches under rare fraud [31]. Alharbi and Alsulami reported the interaction of oversampling, deep ensembling, and minority metrics [32]. Zavvar et al. investigated hybrid deep representations with synthetic oversampling [33]. These findings support a multidimensional interpretation but cannot substitute for independently measured review costs.

Temporal stability and experimental separation are central when a transaction model is evaluated on historical data. Lukats et al. benchmarked drift detection on nonstationary data streams [34]. A broader systematic review describes approaches to detecting and adapting to concept drift [16]. The financial anomaly calibration study formalizes dynamic decision boundaries, which differ from a cutoff selected once during development [7]. A recent analysis of fairness in financial prediction adds reasons to report limitations of anonymized features [30]. A cost-sensitive reliability pipeline demonstrates that calibration itself is a separate operation requiring evidence [35]. The Bayesian fraud study provides a relevant but methodologically different optimization comparator [10]. The present study therefore tests a cost-aware XGBoost system while distinguishing what its recorded threshold settings can actually establish.

Table 1 compares six studies that are directly relevant to cost-sensitive fraud classification and Bayesian tuning. Each contribution addresses either model learning, operating decisions, or evaluation design. Yang et al. examine imbalanced XGBoost objectives in financial transactions. Sang evaluates class-sensitive learning and decision thresholds under an explicit cost scenario. Tayebi and El Kafhali investigate Bayesian XGBoost search, while Dastidar and Hafez synthesize broader detection methods. Bischl et al. provide general principles for fair hyperparameter comparison. Their complementary contributions motivate the controlled ablations and chronological holdout employed here.

Table 1. Related methods and the remaining evaluation gap

Author / year	Relevant contribution	Outstanding question for this study
Yang et al. (2026)	Cost-sensitive XGBoost under imbalance	Does not isolate this notebook's locked cutoff effect
Sang (2026)	Cost-sensitive learning and threshold evaluation	Different validation and loss assumptions
Tayebi and El Kafhali (2025)	Bayesian search of XGBoost for card fraud	Threshold effect and chronological test not identical
Dastidar et al. (2024)	Survey of fraud detection methods	Does not select a specific cost policy
Hafez et al. (2025)	Systematic review of AI fraud detection	Heterogeneous datasets and evaluation protocols
Bischl et al. (2023)	Hyperparameter search and validation principles	General guidance, not this cost-sensitive comparison

Table 1 distinguishes method-level advances from the unresolved need to isolate a threshold effect. Prior work on cost-aware model fitting does not imply that a changing cutoff is responsible for every gain. The Bayesian optimization literature provides a useful search comparator but cannot substitute for a fixed-baseline ablation. The survey studies emphasize class imbalance and evaluation diversity across transaction datasets. Their conclusions are not automatically transferable to an anonymized, single-period holdout. The current study therefore evaluates the actual cutoff values and the full confusion profile in addition to weighted error cost. This separation is essential for interpreting the proposed method without overstating the mechanism demonstrated by the experiment.

2.2. Dataset

The experiment uses the ULB credit-card fraud dataset obtained from the public Kaggle repository and retains every source record. The 284,807 transactions include 492 fraudulent observations, making the outcome strongly imbalanced. Thirty predictors consist of the Time and Amount attributes and anonymized V1–V28 components. The transaction identifier and cardholder identity are not available in the released records. The Class label defines fraud as one and legitimate activity as zero. Table 2 documents the executed dataset characteristics, and the time field is used to preserve chronological partition order. The 20:1 cost ratio is an analytical scenario, not an observed financial loss rate.

Table 2. Dataset characteristics and provenance

Dataset attribute	Executed dataset specification
Public source	ULB credit-card transactions; Kaggle mlg-ulb/creditcardfraud
Research endpoint	Binary Class (fraud=1, legitimate=0); 492/284,807 frauds
Analysis unit	Transaction record; Time and Amount plus V1–V28 (30 inputs)
Proposed partition	Chronological 170,884 train / 56,961 development / 56,962 test
Processing / quality audit	Original records retained; fitting confined to train/development
Annotation / scope check	No cardholder IDs; customer-disjoint validation unavailable

Table 2 specifies the binary target, sample size, predictors, and provenance actually used in the experiment. Fraud prevalence is approximately 0.173%, so accuracy alone would obscure minority-class behavior. The absence of customer identifiers prevents assessment of entity-disjoint generalization. Anonymized principal components also limit direct business interpretation of model inputs. Records were retained at their original frequency rather than synthetically rebalanced. The Time variable supports relative ordering but does not provide cardholder identity or a complete calendar timestamp. The experimental conclusions apply only to this benchmark and its declared decision-cost assumptions.

The chronological evaluation allocates the earliest 170,884 transactions to training and the following 56,961 records to model development. The final 56,962 transactions are reserved for held-out assessment. Preprocessing and model selection are confined to the initial two partitions. This separation helps prevent direct exposure of the test labels during optimization. Table 3 describes the source audit, feature preparation, tuning boundary, and locked comparison. No account-level split can be guaranteed because the original data do not include account identifiers. The single chronological partition also cannot quantify variability across time periods.

Table 3. Executed chronological data and optimization protocol

Stage	Required handling
Stage A: audit	Check original creditcard.csv, missingness, target balance and source hash.
Stage B: split	Chronological 60/20/20 with 170,884 / 56,961 / 56,962 records.
Stage C: fit	Fit XGBoost controls on training; select models on development.
Stage D: freeze	32 Bayesian and 32 random trials; all reported cutoffs = 0.5.
Stage E: report	Locked test cost, recall, precision, F1, MCC, ranking and calibration.

Table 3 identifies the actual order of data validation, fitting, selection, and final assessment. All configurations are compared on the same held-out transactions. The recorded optimization budget is 32 evaluations for Bayesian search and 32 for random search. The notebook executes a full-data analysis without row subsampling. Fixed cost coefficients penalize a false negative by 20 and a false positive by one. These coefficients permit consistent within-study comparisons but should not be interpreted as observed transaction losses. Independent reproducibility requires the external project modules used to execute the optimization.

2.3. Research Framework

Figure 1 presents the study framework, which distinguishes an ordinary XGBoost classifier from cost-sensitive modeling and post-training decision rules. The main comparison holds the source data and chronological split constant. The baseline uses the probability threshold 0.5. Additional controls evaluate class weighting, threshold selection, their combination, and random-search tuning. Bayesian optimization explores the proposed cost-sensitive configuration on development data. All selected configurations are then assessed on the same later transaction period. The figure represents the intended architecture, while the executed settings determine which components actually contributed.

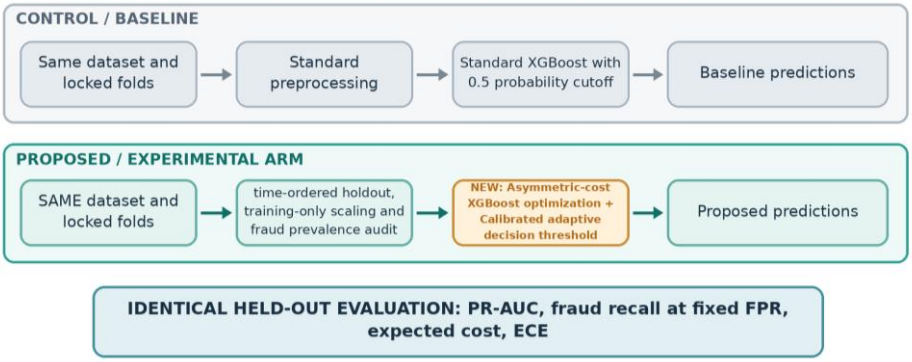


Figure 1. Controlled baseline-versus-proposed research framework

Figure 1 separates training changes from changes to the final binary decision. Class weighting can alter the fitted ranking and probability outputs. Threshold selection changes which scores become alerts without altering the fitted model. These effects require separate ablations to avoid attributing improvement to the wrong component. The recorded result reveals that every selected threshold remained at 0.5, despite the intended adaptive-cutoff design. Consequently, the held-out improvement should be attributed to the selected model configuration, not to a demonstrated threshold movement. The framework remains useful for specifying the comparison, but the observed mechanism is narrower than the complete planned design.

Table 4 specifies six matched test configurations, including the default model and the proposed search. The four simpler branches isolate weighting and threshold changes. Random search is a competing optimizer with the same 32-evaluation budget as Bayesian search. The primary decision criterion is average hypothetical error cost. Secondary indicators include recall, precision, F1, MCC, ROC-AUC, average precision, calibration, and runtime. These metrics address different dimensions of fraud detection rather than replacing one another. The table preserves a common test comparison across all configurations.

Table 4. Baseline and optimization comparison design

Component	Comparative specification
Baseline	Unweighted XGBoost with fixed cutoff 0.5
Proposed innovations	Cost-sensitive Bayesian XGBoost; intended adaptive threshold

Component	Comparative specification
Controlled factors	Same records, chronological split, assumed 20:1 cost ratio
Required ablations	Weight only; threshold only; weight+threshold; random search
Primary comparisons	Cost per transaction; recall, precision, F1, AP, MCC, runtime

Table 4 differentiates the controls necessary for interpreting cost-aware optimization. The weighting-only branch tests whether a training penalty is sufficient. The threshold-only branch tests whether changing the decision rule improves the default forest of boosted trees. Their combined branch evaluates the interaction without a general hyperparameter search. Random search provides a directly relevant alternative to Bayesian optimization. The full Bayesian branch is compared against every simpler alternative under the same holdout. Identical thresholds across all branches limit the evidence for the claimed adaptive threshold component.

2.4. Proposed Method

Figure 2 illustrates the conceptual integration of cost-sensitive XGBoost training and a validation-selected operating threshold. A candidate model generates scores from the 30 original predictors. Cost weighting affects fitting, while decision rules convert scores into binary fraud alerts. The executed optimizer records 32 Bayesian candidate evaluations and 32 random-search candidates. All model-selection decisions are made before final-test evaluation. The available notebook reports calibrated probability diagnostics but does not independently expose the calibration implementation. The observed 0.5 cutoff must be retained when interpreting the final experiment.

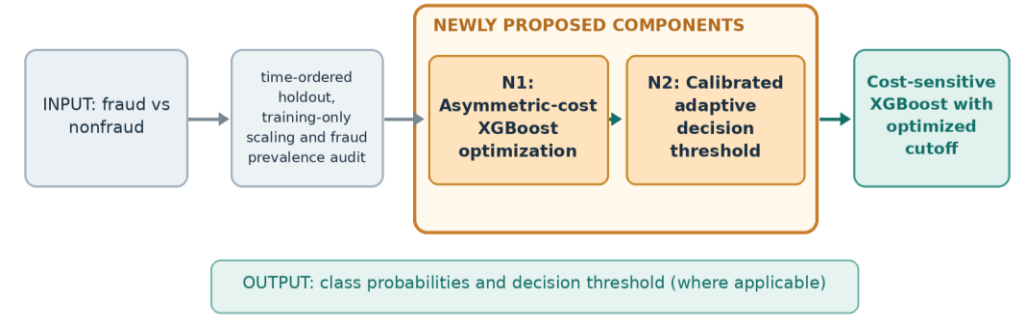


Figure 2. Proposed method architecture with explicitly highlighted novel components

Figure 2 distinguishes the cost-sensitive model from its intended adaptive decision stage. The development record selects threshold 0.5 for the Bayesian model and all ablations. Such a result does not show a gain attributable to moving the decision boundary. Bayesian optimization nevertheless selects a different model configuration from the default baseline. The fixed cutoffs and later chronological test enable a transparent model-level comparison. The decision cost is hypothetical and must be evaluated alongside recall and alert precision. A separate experimental change in threshold would be required to establish the proposed adaptive mechanism as an independently effective component.

Table 5 summarizes the executable inputs, optimization strategy, objective, and evaluation criteria. The learner is an XGBoost binary classifier trained on anonymized transaction features. Candidate fitting is evaluated with a development-set asymmetric cost objective. Candidate cutoffs were included in the experimental naming, although the selected value is 0.5 in every reported branch. The same false-negative and false-positive penalties apply to all candidate rules. The optimization and final assessment use distinct data partitions. Table 5 should therefore be read as the recorded implementation specification, not proof that every planned component improved the result.

Table 5. Executed cost-sensitive decision specification

Design element	Operational definition
Standard input branch	30 original numeric features; original chronological split
Novel block N1	Bayesian cost-sensitive XGBoost, 32 objective evaluations
Novel block N2	Threshold selection represented, selected cutoff = 0.5
Output model	Cost-sensitive Bayesian XGBoost, fixed final cutoff 0.5
Optimization / decision objective	Minimize (20 FN + FP)/N on development data
Ablation comparison	Unweighted; weighting; cutoff; combined; random search
Acceptance endpoints	Held-out cost, recall/precision, AP, F1/MCC, efficiency

Table 5 defines the experimental decision objective as weighted misclassification errors per evaluated transaction. The false-negative penalty is 20 units and the false-positive penalty is one unit. Minimizing this cost differs from maximizing overall accuracy because missed fraud receives greater weight. The selected rule labels a transaction fraudulent when its predicted score exceeds the specified cutoff. All reported final cutoffs are 0.5, so the study does not establish independent threshold improvement. The objective remains meaningful as a comparative benchmark under the stated assumptions. The equations below formalize classification, error cost, discrimination, and relative improvement.

The model fitting and optimizer assessment were executed on the complete source dataset. Transaction order defined an approximate 60:20:20 partition containing 170,884 training, 56,961 development, and 56,962 test records. The configuration used random seed 42 and a 32-trial budget for each of Bayesian and random optimization. The notebook reports 64 development-stage objective evaluations across the two searched strategies. Both selected optimized models retain all 30 predictors. Executed test evaluation follows model selection without reselecting hyperparameters on held-out labels. The optimizer and calibration implementation was unavailable for independent audit, restricting verification of those implementation details.

$$p_i = f\theta(x_i) \quad (1)$$

$$\hat{y}_i(\tau) = 1[p_i \geq \tau] \quad (2)$$

$$C(\theta,\tau) = [20\,FN(\theta,\tau) + FP(\theta,\tau)] / N \quad (3)$$

$$(\theta*,\tau*) \in arg\,min\,Cdev(\theta,\tau) \quad (4)$$

$$Precision = TP/(TP + FP);\ Recall = TP/(TP + FN) \quad (5)$$

$$F1 = 2PR/(P + R);\ Balanced\ accuracy = (TPR + TNR)/2 \quad (6)$$

$$Cost\ reduction\ (\%) = 100 \times (Cbaseline - Cproposed)/Cbaseline \quad (7)$$

$$Brierscore = (1/N)\Sigma_i(p_i - y_i)^2 \quad (8)$$

The binary decision and experimental evaluation follow the definitions in Equations (1)–(8). Equation (1) represents a model-derived fraud score rather than a verified calibrated posterior probability. Equation (2) translates that score into a binary alert at threshold τ. Equations (3) and (4) express development-set cost and candidate selection under the declared penalty ratio. Equations (5) and (6) define confusion-based performance indicators. Equation (7) measures relative cost improvement, while Equation (8) defines the Brier probability-error metric. The mathematical notation does not assert that the observed threshold varied, because all reported cutoffs equal 0.5.

3. Results and Discussion

3.1. Held-Out Cost and Classification Performance

The chronological test contains 56,962 transactions, including 75 fraudulent observations. All six alternatives use the same held-out labels and scenario costs. The default classifier identifies 53 frauds while the optimized Bayesian classifier identifies 56. This gain reduces missed events from 22 to 19 but increases false alarms from four to five. The random-search model misses 25 frauds and produces three false alerts. The fixed 0.5 cutoff in every branch shows that the selection benefit is associated with the model configuration rather than demonstrated threshold adaptation. Table 6 provides the joint cost and predictive comparison.

Table 6. Held-out baseline versus cost-sensitive optimization (FN cost 20, FP cost 1)

Model	Cost/sample	Recall	Precision	F1	FP	FN
Default XGBoost	0.007795	0.7067	0.9298	0.8030	4	22
Weight only	0.007795	0.7067	0.9298	0.8030	4	22
Threshold only	0.007795	0.7067	0.9298	0.8030	4	22
Weight + threshold	0.007795	0.7067	0.9298	0.8030	4	22
Random search + threshold	0.008830	0.6667	0.9434	0.7813	3	25
Bayesian XGBoost + threshold	0.006759	0.7467	0.9180	0.8235	5	19

Table 6 shows that the Bayesian configuration has the lowest cost per transaction among the six alternatives. Its cost is 0.006759 compared with 0.007795 for the default classifier and 0.008830 for random search. The difference arises from 19 rather than 22 missed frauds despite one additional false alert. All three simple ablations reproduce the default test counts and performance in the available output. Precision decreases to 0.9180, whereas recall rises to 0.7467 and F1 increases to 0.8235. The proposed model improves the primary assumed-cost endpoint by 13.29% relative to baseline. These are descriptive differences from one time-separated test, not independently established statistical superiority.

Additional evaluation indicators distinguish the cost-oriented gain from other performance attributes. Baseline balanced accuracy equals 0.853298, whereas the proposed configuration reaches 0.873289. The proposed Matthews correlation coefficient rises to 0.827729 from 0.810393 for the baseline. Specificity remains very high because 56,887 legitimate transactions dominate the held-out set. The Bayesian classifier labels five of those legitimate records fraudulent and correctly identifies 56 of 75 frauds. The random-search control identifies only 50 frauds but generates three false alarms. These observations explain why ranking quality, threshold-dependent accuracy, and economic cost need separate interpretation.

3.2. Development Selection and Computational Trade-offs

Table 7 reports the development-stage search results and the selected decision settings. Bayesian optimization and random search each execute 32 objective evaluations, totaling 64 trial scores. Their development objective values are −0.005319 and −0.005337, respectively. The unweighted reference returns −0.006215, while class weighting returns −0.006022. The development scores are signed negative costs, so a value nearer zero is preferred. Each optimized branch retains all 30 predictor columns. The table also distinguishes optimization time from the later refit-and-test duration.

Table 7. Development selection, thresholds, and computation

Configuration	Dev objective	Threshold	Search trials	Search fit (s)	Test refit (s)	Features
Default	-0.006215	0.500	—	7.219	9.781	30
Weight only	-0.006022	0.500	—	7.046	9.812	30
Random search	-0.005337	0.500	32	264.799	9.641	30
Bayesian search	-0.005319	0.500	32	374.941	15.390	30

Table 7 indicates that Bayesian search achieves a slightly better development objective than matched random search. Bayesian optimization requires 374.941 seconds of recorded search-related fitting, compared with 264.799 seconds for random search. Its final refit-and-test step takes 15.390 seconds, compared with 9.781 seconds for the baseline. The reported prediction latency is approximately 7.67 ms per thousand transactions for the proposed model and 4.11 ms for the baseline. These differences indicate that lower assumed decision cost was accompanied by greater computational expense. Every recorded cutoff remains 0.5, showing no observed movement in the threshold-selection stage. Search-time differences should not be interpreted as a hardware-independent efficiency estimate.

3.3. Diagnostic Figures and Error Structure

Figure 3 introduces the original class distribution and missingness profile before examining predictive performance. The public credit-card benchmark is markedly imbalanced, with only 492 positives among 284,807 source rows. The dataset contains anonymized transaction components together with Time and Amount. This profile motivates evaluation based on fraud recall and precision instead of accuracy alone. The source audit also establishes that model training can use complete records rather than an artificially balanced subset. The figure provides the empirical starting point for the comparisons that follow.

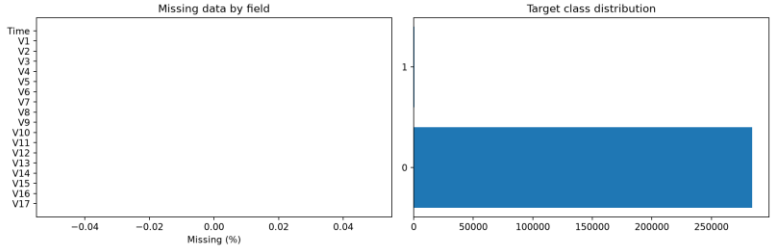


Figure 3. Dataset composition and missingness

The source overview displays the distribution of fraud labels and checks missingness for the retained predictors. Fraud accounts for a very small proportion of the 284,807 source records. The panel therefore makes the limitations of ordinary accuracy visually apparent. No missingness is displayed for the audited input variables. Anonymized V1–V28 features cannot be interpreted as specific cardholder behaviors. The source representation is consistent across all compared learning branches. This audit provides context for subsequent error and cost interpretation.

Figure 4 compares test performance across the default, ablation, random-search, and Bayesian models. Each branch was evaluated on the same chronological holdout. Cost, minority recall, F1, and discrimination summarize distinct aspects of prediction quality. The primary hypothesis concerns weighted decision cost, while the other measures reveal what changes accompany its reduction. Numerical differences are interpreted using the actual confusion counts. The visual comparison makes it possible to inspect any degradation that a favorable cost statistic might obscure.

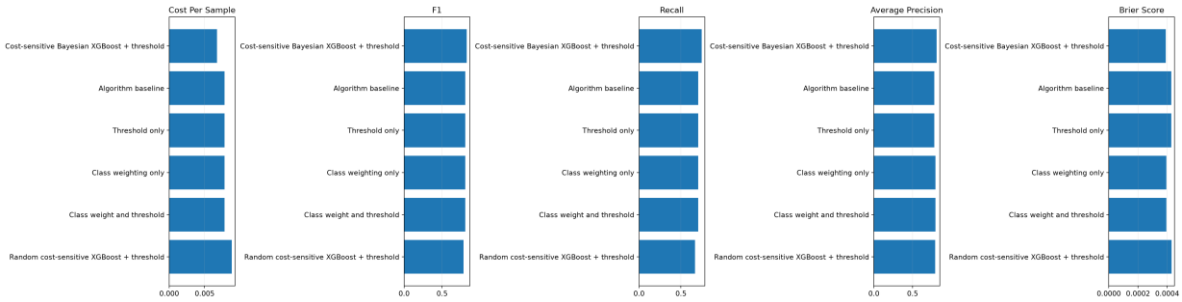


Figure 4. Comparison of cost, recall, F1 and probability metrics

The model comparison juxtaposes several metrics that respond differently to imbalance. Bayesian optimization achieves the lowest assumed cost and the highest recall among the recorded branches. The unweighted baseline maintains higher precision than the proposed configuration. The random-search control does not improve the primary cost metric on the held-out period. Accuracy changes only slightly because legitimate records dominate the total. Ranking and probability measures offer additional diagnostic information but cannot establish operating cost independently. The panel supports a multidimensional rather than a single-score conclusion.

Figure 5 plots ROC curves for all six models on the locked test partition. The curves examine sensitivity against the fraction of legitimate transactions classified as fraud. The operating region near zero false-positive rate is particularly important because legitimate records vastly outnumber frauds. The curves represent score ranking rather than the outcome under one specific threshold. A high global ROC-AUC therefore does not determine the minimum-cost alert rule. This distinction motivates the complementary precision-recall analysis.

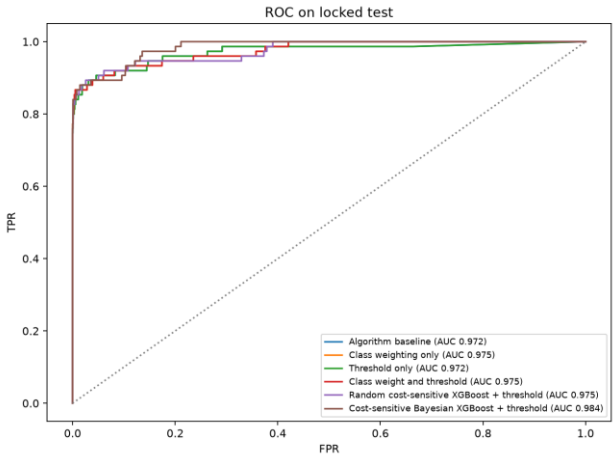


Figure 5. Receiver operating characteristic curves

The ROC curves characterize sensitivity across false-positive operating points. Their overall shapes suggest high separation between the majority of legitimate and fraudulent transactions. A high ROC-AUC does not by itself indicate that a particular cutoff minimizes assumed review cost. The observed fraud outcomes are sparse relative to the negative transactions. Low false-positive rates can still correspond to actionable numbers of alerts in large-scale

systems. The model curves should therefore be interpreted with the confusion matrices and precision-recall results. No ROC-based threshold has been selected using this held-out plot.

Figure 6 shows precision and recall across the spectrum of model decision cutoffs. For rare fraud, precision quantifies the proportion of alerts that truly correspond to fraudulent transactions. Recall measures the proportion of known frauds that would be flagged. Their trade-off becomes visible as progressively smaller scores are admitted as positive predictions. Average precision is evaluated on the fixed test set and is not used to retune the models. The displayed comparison complements the fixed-cutoff operating statistics reported in Table 6.

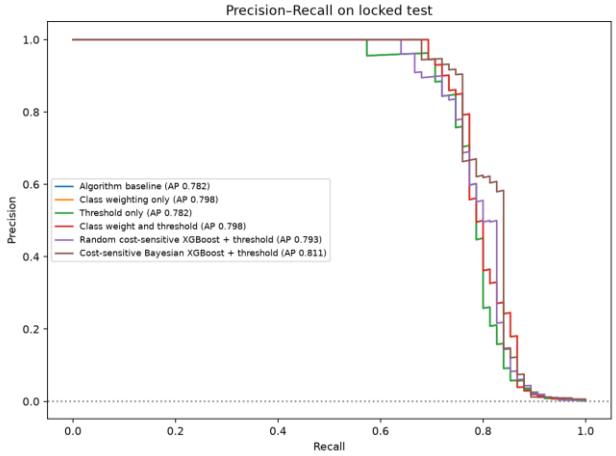


Figure 6. Precision-recall curves

The precision-recall curves focus on the positive class across decision thresholds. Their shape reveals the relationship between additional fraud detection and reduced alert precision. The proposed model has a favorable precision-recall ranking relative to several controls in the recorded figure. A curve-level improvement does not imply that all fixed operating thresholds are preferable. The exact deployment decision still depends on the assumed penalty ratio and inspection capacity. Positive prevalence differs across datasets, which complicates comparison of average precision without context. The held-out curve was used for description, not for reselecting the operating point.

Figure 7 presents the held-out confusion matrices, allowing the primary cost calculations to be verified directly. The positive class contains 75 observed frauds in the test partition. The default classifier misses 22 of these events, whereas the Bayesian model misses 19. The additional detection comes with one more false alert. The random-search alternative misses 25 frauds while reducing false alerts to three. These discrete events determine the cost ranking under the assumed 20:1 penalty ratio.

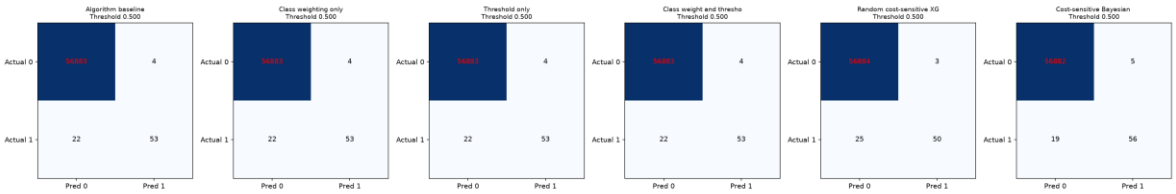


Figure 7. Locked-test confusion matrices

The confusion matrices expose the operational difference behind the cost reduction. The default model produces 53 true fraud alerts, four false alarms, and 22 missed fraudulent cases. Bayesian optimization raises true fraud detection to 56 while producing five false alarms and 19 misses. Its additional true positives outweigh the single extra false alert under the assumed 20:1 penalty. The random-search branch identifies 50 frauds and misses 25, despite producing only three false alarms. These counts reconcile the observed recall, precision, and cost values. They also show why reporting only overall accuracy would be misleading.

Figure 8 evaluates the correspondence between estimated fraud scores and observed event frequencies. The reliability diagram uses the fixed holdout rather than a fresh validation dataset. Because fraud is rare, some score bins contain very few positive outcomes. Apparent departures from perfect reliability may therefore reflect both model behavior and sparse-bin variation. The plot characterizes probability quality, which differs from precision,

recall, and ROC-AUC. It does not independently establish which calibration algorithm was actually run inside the unavailable external module.

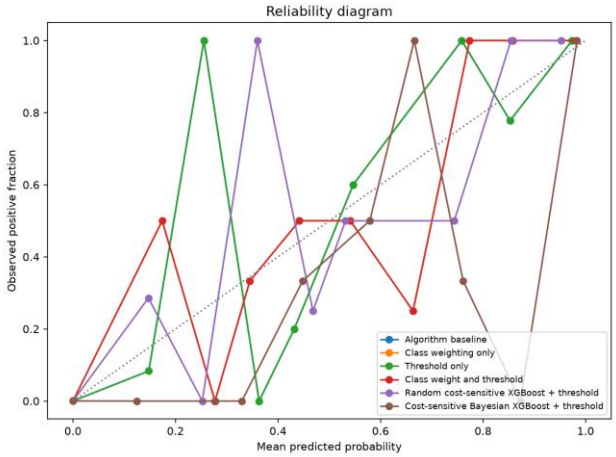


Figure 8. Probability reliability diagram

The reliability diagram compares predicted fraud scores with empirical event frequencies. Its irregular shape reflects a limited number of positive cases at some score ranges. Visual calibration diagnostics do not prove that a specific post-hoc calibration algorithm was executed correctly. The supplied notebook invokes external project modules that were not included with the manuscript materials. This prevents independent verification of any internal sigmoid calibration implementation. Probability reliability is particularly important if thresholds are to be interpreted as economic risk cutoffs. The figure therefore supports descriptive probability assessment rather than a claim of validated calibration.

Figure 9 displays score distributions separately for the two true transaction classes. Most legitimate records have very low assigned fraud scores. A subset of frauds receives similarly low scores, which creates false negatives at the fixed cutoff. Other fraudulent observations occupy higher-score regions and can be detected more reliably. The class-conditional overlap helps explain the precision-recall trade-off. This view supplements, rather than replaces, the confusion matrix used in the primary cost calculation.

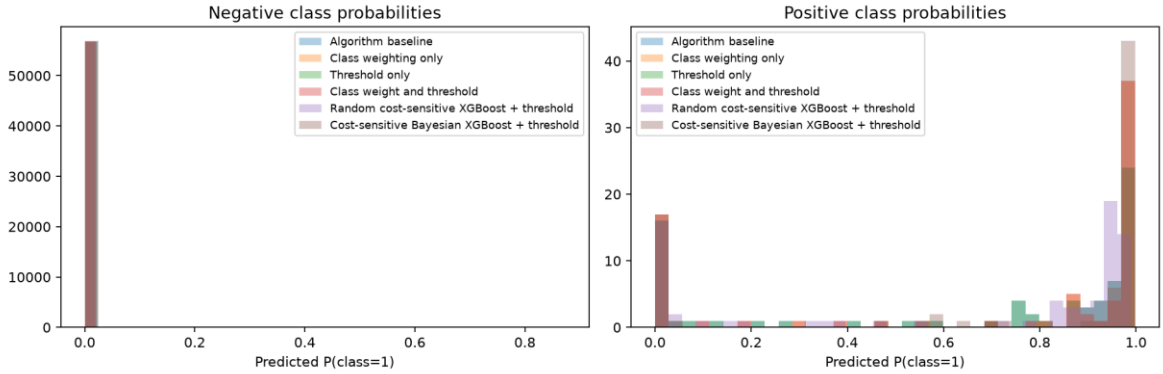


Figure 9. Fraud-score distributions by true class

The class-conditional score distributions show substantial concentration of legitimate observations near low predicted fraud probability. Fraudulent observations have a wider score distribution and overlap with legitimate records. This overlap explains why simultaneously maximizing recall and precision is difficult. Rare false-positive events remain consequential because of the dominant negative-class volume. The observed risk scores remain interpretable only within the benchmark and evaluation period. The plot does not identify a transaction-specific financial consequence. Its primary value is diagnostic support for the confusion-based trade-off.

Figure 10 illustrates the sensitivity of weighted classification cost to the choice of cutoff. The curves are generated using the previously frozen test predictions. They are not an additional optimization stage because no test-based threshold changes were applied. The optimization records retain the default decision boundary of 0.5 for all reported configurations. The shape of each curve demonstrates that a different cutoff could alter the relative balance

of missed fraud and false alarms. Formal testing of that possibility requires a genuinely development-selected operating threshold.

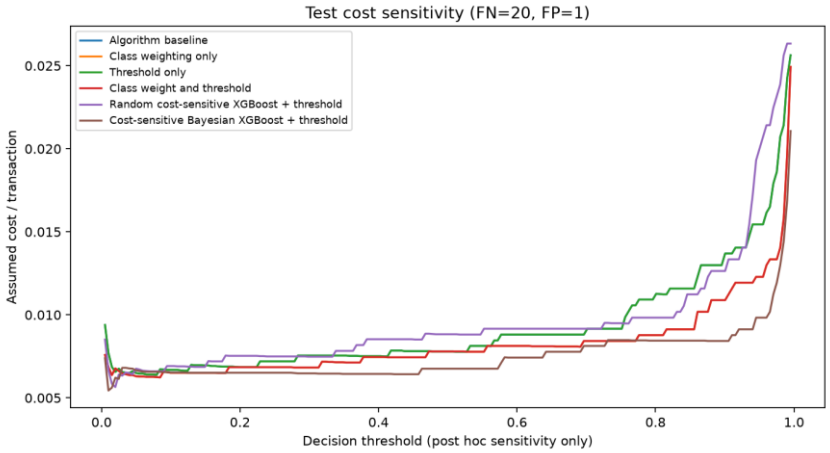


Figure 10. Illustrative held-out cost across thresholds

The threshold-cost curves evaluate sensitivity to alternative operating cutoffs on the held-out labels. They are descriptive and must not be treated as an additional model-selection procedure. The reported development-selected thresholds remain 0.5 for every final configuration. The curve shapes nevertheless show that hypothetical cost can change as more transactions become flagged. A different scenario penalty could move the preferred operating point. The figure does not establish that the deployed method adapted its threshold dynamically. Independent development-only threshold selection would be required to test that contribution directly.

Figure 11 displays the completed development objective evaluations from the two search methods. Bayesian and random search each contribute 32 candidate evaluations. Candidate quality fluctuates as model configurations are explored. The best-so-far record summarizes the search process but not the statistical stability of its final outcome. The corresponding selected configurations are tested once on chronologically later observations. The trace provides evidence of actual optimization work without serving as an independent test.

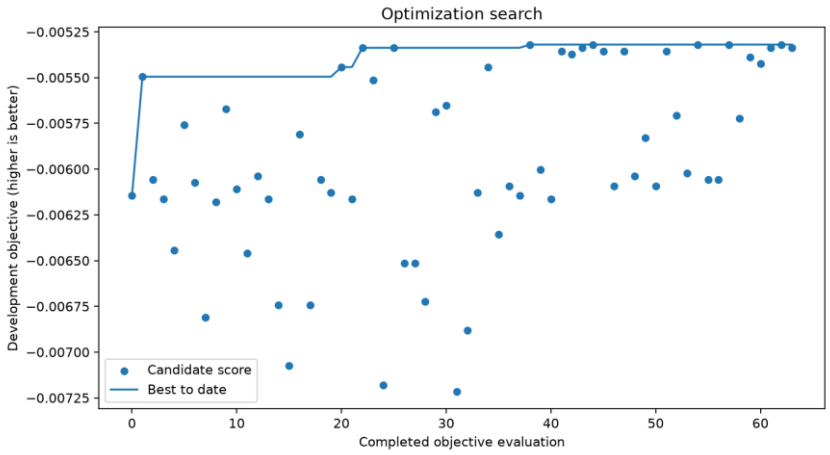


Figure 11. Development optimizer trace

The optimizer trace displays the 64 development objective evaluations recorded across the search procedures. Candidate scores vary with the tested hyperparameters and model configurations. The best-so-far trajectory improves during search on the development partition. Such an improvement is not itself evidence of held-out generalization. Final model selection remains constrained to development results and a fixed search budget. The test comparison must be consulted to establish whether candidate search improved outcomes. The trace documents the optimization effort without suggesting statistical significance.

Figure 12 examines the effect of assigning different hypothetical penalties to missed fraud. Each cost curve is derived from fixed model predictions and altered decision-cost coefficients. The analysis distinguishes the reported 20:1 scenario from other potential operating conditions. The model that misses fewer fraudulent cases becomes

increasingly favored as the false-negative penalty grows. The resulting rankings should not be interpreted as measured bank revenue or loss. The figure instead tests the sensitivity of the benchmark conclusions to declared financial assumptions.

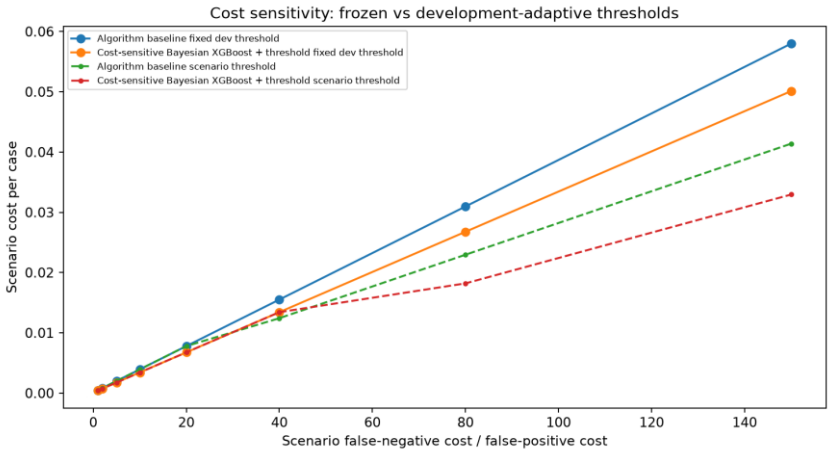


Figure 12. Sensitivity to hypothetical false-negative penalties

The cost-scenario plot varies the assumed penalty assigned to missed fraud. Relative method performance changes as false negatives become more expensive. The baseline and Bayesian model have different false-negative counts, so their cost curves exhibit different slopes. The observed primary comparison uses a single 20:1 assumption rather than audited bank losses. Alternative cost ratios are sensitivity analyses, not extra training experiments. The figure is therefore relevant to decision-policy robustness but not to measured monetary savings. Economic conclusions require independently justified false-negative and alert-review costs.

3.4. Interpretation and Limitations

The proposed Bayesian configuration provides an observed advantage on the prespecified assumed-cost endpoint. The 13.29% relative reduction corresponds to three fewer missed frauds and one additional false alarm. Its higher recall and F1 are accompanied by lower precision and increased computational expense. The strongest test control is the default XGBoost configuration, whereas matched-budget random search performs worse than both. The nominal threshold-adjustment branches are identical to their respective underlying fitted models because every selected cutoff is 0.5. The experiment therefore supports tuning-related performance improvement, not a separately validated adaptive threshold contribution. These results cannot be extrapolated to bank operations without actual cost estimates and external validation.

Several methodological limitations constrain the interpretation of the favorable held-out comparison. The account and card identifiers needed to test customer-disjoint generalization are absent from the anonymized dataset. The chronological partition represents one specific historical transfer rather than a series of independent future deployments. The false-negative and false-positive costs are illustrative and might not match transaction amounts or investigation costs. Model calibration cannot be independently audited because the supporting implementation modules were not supplied. A paired uncertainty analysis was not available for independent verification from the uploaded evidence. Repeated temporal windows, explicit decision-threshold ablation, and transparent cost estimation are needed before stronger claims can be considered.

4. Conclusion

The full-data chronological experiment observes lower hypothetical decision cost for Bayesian-optimized cost-sensitive XGBoost than for the default model. The reduction from 0.007795 to 0.006759 per transaction corresponds to 13.29% under a 20:1 false-negative to false-positive penalty ratio. The proposed configuration detects three additional fraudulent transactions and produces one additional false alert on 56,962 held-out transactions. Its recall, F1, and balanced accuracy increase, while precision decreases and computational cost rises. Random search performs worse than the Bayesian configuration at the same recorded number of objective evaluations. The identical 0.5 thresholds across the compared models provide no independent evidence that adaptive threshold selection contributed to the gain. The result supports an observed benchmark improvement in weighted classification cost, not verified financial savings, calibrated probabilities, or deployment readiness.

The absence of an observed threshold change is the study's most consequential mechanistic limitation. A complete evaluation of the proposed adaptive decision component would require a development-selected cutoff differing from the default or a documented operating-policy update. Further evidence should include matched calibration controls, time-window replication, and an externally verifiable optimizer implementation. Transaction-dependent decision costs should be estimated from operational data rather than selected exclusively for benchmark convenience. Additional analysis should establish uncertainty around the paired cost difference and monitor rare-event recall after distribution shift. The present findings support an encouraging cost-sensitive tuning result on the ULB benchmark while leaving adaptive-threshold effectiveness and real-world generalizability unresolved.

References

[1] L. Hernandez Aros, L. X. Bustamante Molano, F. Gutierrez-Portela, J. J. Moreno Hernandez, and M. S. Rodríguez Barrero, "Financial fraud detection through the application of machine learning techniques: a literature review," *Humanit. Soc. Sci. Commun.*, 2024, doi: 10.1057/s41599-024-03606-0.

[2] K. G. Dastidar, O. Caelen, and M. Granitzer, "Machine Learning Methods for Credit Card Fraud Detection: A Survey," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3487298.

[3] 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*, 2025, doi: 10.1186/s40537-024-01048-8.

[4] J. Yang, D. Zhu, Z. Wang, and Z. Zhang, "Decoupled focal loss and bayesian temperature fusion XGBoost for imbalanced credit card fraud detection," *Sci. Rep.*, 2026, doi: 10.1038/s41598-026-64196-2.

[5] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, "A survey on imbalanced learning: latest research, applications and future directions," *Artif. Intell. Rev.*, 2024, doi: 10.1007/s10462-024-10759-6.

[6] V. N. T. Sang, "Enhancing Credit Card Fraud Detection under Severe Class Imbalance using Cost-Sensitive Learning and Threshold Optimization," *EAI Endorsed Transactions on Intelligent Systems and Machine Learning Applications*, 2026, doi: 10.4108/eetismla.12078.

[7] A. Huang, Y. Wang, P. Zhou, X. Zhang, S. Tsai, and L. Chen, "Dynamic Calibration of Decision Thresholds for Financial Anomaly Detection: Verification With Payment Platform Information and Data," *Journal of Global Information Management*, 2025, doi: 10.4018/JGIM.395852.

[8] T. de Menezes e Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, "Classifier calibration: A survey on how to assess and improve predicted class probabilities," *Mach. Learn.*, 2023, doi: 10.1007/s10994-023-06336-7.

[9] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining and Knowledge Discovery*, 2023, doi: 10.1002/widm.1484.

[10] M. Tayebi and S. El Kafhali, "A novel approach based on XGBoost classifier and Bayesian optimization for credit card fraud detection," *Cyber Security and Applications*, 2025, doi: 10.1016/j.csa.2025.100093.

[11] B. Demir Sezgin and Ş. M. Kaya, "A Constraint-Aware Soft Voting Ensemble Framework for Credit Card Fraud Detection under Operational Threshold Optimization," *Turkish Journal of Science and Technology*, 2026, doi: 10.55525/tjst.1931327.

[12] E. Richardson, R. Trevizani, J. A. Greenbaum, H. Carter, M. Nielsen, and B. Peters, "The receiver operating characteristic curve accurately assesses imbalanced datasets," *Patterns*, 2024, doi: 10.1016/j.patter.2024.100994.

[13] I. D. Mienye and N. Jere, "Deep Learning for Credit Card Fraud Detection: A Review of Algorithms, Challenges, and Solutions," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3426955.

[14] M. A. Talukder, M. Khalid, and M. A. Uddin, "An integrated multistage ensemble machine learning model for fraudulent transaction detection," *J. Big Data*, 2024, doi: 10.1186/s40537-024-00996-5.

[15] M. Z. Yüzügüldü, H. O. Altun, A. Çoşkun, and E. Şener, "A multi-criteria ranking framework for machine learning algorithms: credit card fraud detection," *International Transactions in Operational Research*, 2026, doi: 10.1111/itor.70245.

[16] S. Arora, R. Rani, and N. Saxena, "A systematic review on detection and adaptation of concept drift in streaming data using machine learning techniques," *WIREs Data Mining and Knowledge Discovery*, 2024, doi: 10.1002/widm.1536.

[17] Z. Salekshahrezaee, J. L. Leevy, and T. M. Khoshgoftaar, "The effect of feature extraction and data sampling on credit card fraud detection," *J. Big Data*, 2023, doi: 10.1186/s40537-023-00684-w.

[18] D. Breskuvienė and G. Dzemyda, "Enhancing credit card fraud detection: highly imbalanced data case," *J. Big Data*, 2024, doi: 10.1186/s40537-024-01059-5.

[19] R. K. L. Kennedy, F. Villanustre, T. M. Khoshgoftaar, and Z. Salekshahrezaee, "Synthesizing class labels for highly imbalanced credit card fraud detection data," *J. Big Data*, 2024, doi: 10.1186/s40537-024-00897-7.

[20] R. K. L. Kennedy, F. Villanustre, and T. M. Khoshgoftaar, "Unsupervised feature selection and class labeling for credit card fraud," *J. Big Data*, 2025, doi: 10.1186/s40537-025-01154-1.

[21] M. A. Walauskis and T. M. Khoshgoftaar, "Scalable unsupervised labeling with SHAP feature selection for fraud detection in imbalanced data," *J. Big Data*, 2025, doi: 10.1186/s40537-025-01248-w.

[22] I. Almubark, "Advanced Credit Card Fraud Detection: An Ensemble Learning Using Random Under Sampling and Two-Stage Thresholding," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3519335.

[23] E. Ileberi and Y. Sun, "A Hybrid Deep Learning Ensemble Model for Credit Card Fraud Detection," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3502542.

[24] H. Huang, B. Liu, X. Xue, J. Cao, and X. Chen, "Imbalanced credit card fraud detection data: A solution based on hybrid neural network and clustering-based undersampling technique," *Appl. Soft Comput.*, 2024, doi: 10.1016/j.asoc.2024.111368.

[25] A. H. Salem, S. M. Azzam, O. E. Emam, and A. A. Abohany, "From chaos to clarity: unraveling credit card fraud with BGVOA-LS," *J. Big Data*, 2025, doi: 10.1186/s40537-025-01274-8.

[26] F. Atban, M. Y. Küçükkara, and C. Bayılmış, "Enhancing variational quantum classifier performance with meta-heuristic feature selection for credit card fraud detection," *Eur. Phys. J. Spec. Top.*, 2025, doi: 10.1140/epjs/s11734-025-01703-y.

[27] M. A. Alrasheedi, "Enhancing Fraud Detection in Credit Card Transactions: A Comparative Study of Machine Learning Models," *Comput. Econ.*, 2026, doi: 10.1007/s10614-025-11071-3.

[28] S. S. Awad, A. A. Hamza, M. A. Sobh, and A. M. Bahaa-Eldin, "Applying explainable artificial intelligence to interpret supervised ensemble learning models for robust credit card fraud detection," *Sci. Rep.*, 2026, doi: 10.1038/s41598-026-49939-5.

[29] E. Z. Çavdar and A. Bozanta, "AI-based credit card fraud detection: a machine learning approach with model explainability on real-world data," *Knowl. Inf. Syst.*, 2026, doi: 10.1007/s10115-026-02773-7.

[30] C. N. Nwafor, O. Nwafor, and S. Brahma, "Enhancing transparency and fairness in automated credit decisions: an explainable novel hybrid machine learning approach," *Sci. Rep.*, 2024, doi: 10.1038/s41598-024-75026-8.

[31] M. A. Ghalwash, S. M. Abdelrazek, N. H. Eladawi, and H. A. Ghalwash, "Enhancing credit card fraud detection using DBSCAN-augmented disjunctive voting ensemble," *Sci. Rep.*, 2025, doi: 10.1038/s41598-025-22960-w.

[32] L. A. Alharbi and F. Alsulami, "A SMOTE enhanced multi stream deep learning ensemble for credit card fraud detection," *Sci. Rep.*, 2026, doi: 10.1038/s41598-026-75271-z.

[33] M. Zavvar *et al.*, "A hybrid deep learning framework using synthetic oversampling, autoencoder, convolutional neural networks, and an attention mechanism for credit card fraud detection," *J. Big Data*, 2026, doi: 10.1186/s40537-025-01331-2.

[34] D. Lukats, O. Zielinski, A. Hahn, and F. Stahl, "A benchmark and survey of fully unsupervised concept drift detectors on real-world data streams," *Int. J. Data Sci. Anal.*, 2025, doi: 10.1007/s41060-024-00620-y.

[35] D. Danang and T. W. A. Putra, "Cost-Sensitive Fraud Detection with Reliability Calibration: A Practical Pipeline with XGBoost and Focal-Proxy Reweighting," *International Journal of Information Technology and Business*, 2025, doi: 10.24246/ijiteb.812025.13-24.