© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722134

Score, Review, Retrain: A Closed-Loop Architecture for Real-Time Fraud Detection with Human-Verified Model Retraining in Digital Banking

AAISHA FAROOQ
Department of Information Technology, Central University of Kashmir, Ganderbal, Jammu and Kashmir, India

Abstract- ***The expansion of digital banking has widened the surface available to transaction fraud, and fraud detection mechanisms built on static, predefined rules have struggled to keep pace with fraud patterns that change faster than the rules describing them. Most published machine learning approaches to this problem evaluate a trained classifier against a fixed, offline dataset, without describing how that classifier would be embedded into a live transaction pipeline or how it would continue to improve once deployed. This paper presents the design, implementation, and evaluation of a fraud detection system in which risk scoring, human review, and model retraining are integrated into a single closed loop within a working digital banking backend, rather than treated as separate offline concerns. Every fund transfer and withdrawal is scored in real time by a Gradient Boosting classifier trained on more than forty behavioral, temporal, and velocity-derived features; transactions exceeding a configured risk threshold are withheld pending review by a human analyst, and confirmed outcomes are incorporated into subsequent, automatically scheduled retraining cycles. The system was evaluated on synthetically generated transaction data of two different sizes, with the final deployed configuration achieving an area under the receiver operating characteristic curve (AUC) of 0.94 and, at a class-balancing ratio tuned specifically to preserve precision, a precision of 0.24 with a recall of 0.57 at an F1-optimal decision threshold. Gradient Boosting is further compared against Logistic Regression and Random Forest baselines trained and evaluated under identical conditions. The results and their trade-offs are discussed in relation to dataset size, class-imbalance correction, and decision-threshold selection, and the paper concludes with a candid account of the system's limitations and directions for future work.***

Index Terms— ***Automated Retraining, Class Imbalance, Digital Banking, Fraud Detection, Gradient Boosting, Human-In-The-Loop, Machine Learning Operations, Real-Time Systems***

I. INTRODUCTION

Digital banking has replaced over-the-counter transactions as the primary channel through which most financial activity now occurs, driven by the widespread adoption of mobile applications, internet banking, and electronic fund transfer systems. This shift has brought considerable convenience and speed to financial services, but it has simultaneously widened the surface available to fraudulent activity. Unauthorized fund transfers, account takeover, and increasingly adaptive fraud patterns have grown in step with the growth of digital financial infrastructure itself.

Financial institutions have historically relied on rule-based detection, in which transactions are flagged according to fixed, predefined conditions such as a transaction amount exceeding a set threshold. Such systems are simple to implement and audit, but they are static by construction: they can only detect patterns that have been explicitly anticipated, and they require continuous manual revision as fraudulent actors adapt their behavior to evade known rules [1]. Rule-based systems are also known to generate a disproportionately high number of false positives, which inconveniences legitimate customers and imposes a substantial review burden on fraud investigation teams.

These limitations have motivated a broad shift toward machine learning-based fraud detection, capable of learning complex patterns directly from historical transaction data. Ensemble methods, and Gradient Boosting in particular, have demonstrated consistently strong performance on structured, tabular transaction data across multiple independent

studies [2], [3], [6]. A separate and equally persistent line of research has addressed the severe class imbalance inherent to fraud data, where confirmed fraud cases routinely represent a small fraction of total transaction volume, and has shown that naive training on such data biases classifiers toward the majority class unless this imbalance is explicitly corrected [9], [11].

A gap that receives comparatively less attention in the literature is architectural rather than algorithmic. Many published fraud detection studies report the performance of a trained classifier against a static, offline dataset without describing how that classifier would be embedded into a live transaction-processing pipeline, how a flagged transaction would be handled operationally, or how the deployed model would continue to improve as fraud patterns shift after deployment. Related work that does examine human feedback in fraud detection has generally studied its effect on model accuracy in isolation, using proprietary or benchmark datasets, rather than as part of a deployed, end-to-end system [12].

This paper addresses that gap directly. We describe and evaluate a complete digital banking backend in which fraud risk scoring occurs synchronously with transaction processing, transactions exceeding a configured risk threshold are withheld and routed to a human reviewer rather than resolved unilaterally by the model, and reviewer outcomes are incorporated into a scheduled, automated retraining pipeline managed through a dedicated experiment-tracking platform [10]. The specific contributions of this paper are as follows.

1) The design and implementation of a fraud detection mechanism that intercepts a transaction before it completes, holding it mid-flow pending review rather than allowing it to finalize and relying on after-the-fact detection, reversal, or chargeback, as is the case in many deployed and academically evaluated systems.

2) A human-in-the-loop review mechanism whose confirmed outcomes are structurally connected to a scheduled, automated model retraining pipeline, closing the loop between human judgement and model improvement.

3) An empirical sensitivity analysis of Gradient Boosting performance across dataset size, class-balancing ratio, and decision-threshold selection, compared against Logistic Regression and Random Forest baselines trained under identical conditions.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed system architecture. Section IV details the feature engineering and model training methodology. Section V describes the experimental setup. Section VI presents and discusses the results. Section VII discusses limitations, and Section VIII concludes.

II. RELATED WORK

Early fraud detection research relied on rule-based and statistical methods, which offered interpretability but required continual manual revision as fraud patterns evolved [1]. Supervised learning subsequently became the dominant approach: comparative studies have repeatedly found that ensemble methods outperform individual classifiers such as Naive Bayes or k-Nearest Neighbours, particularly on the imbalanced data characteristic of fraud detection [2], [3]. Among ensemble techniques, Gradient Boosting has shown consistently strong results; its theoretical basis was established by Friedman [4] as an iterative residual-correction method, later extended by scalable implementations such as XGBoost [5], and a broad evaluation on banking fraud data found gradient boosting-based models to outperform alternative algorithms across AUC, precision, recall, and F1-score [6].

Feature engineering has been identified as a determinant of detection performance that is at least as important as algorithm choice. Bahnsen et al. demonstrated that incorporating behavioral and temporal features, such as transaction timing and historical spending patterns, substantially improves detection accuracy beyond what raw transaction attributes alone provide [7], a finding that directly informed the multi-category feature extraction approach adopted in this work. Manzoor and Aslam

further identified real-time processing, automated retraining, and human review as characteristic of effective modern fraud detection system design, though largely as a set of desirable properties rather than a demonstrated, integrated implementation [8].

Class imbalance remains one of the most persistent challenges in this domain, since confirmed fraud typically constitutes a small fraction of total transaction volume. Dal Pozzolo et al. showed that probability calibration and resampling improve classifier performance specifically on the minority class [9], and the Synthetic Minority Over-sampling Technique (SMOTE) [11] has become a standard tool for addressing this imbalance during training, an approach adopted in the present work and discussed further in Section IV.

A comparatively recent body of practitioner and academic work has examined human-in-the-loop feedback in fraud detection specifically. Kadam studied the effect of subject-matter-expert feedback on fraud classifier accuracy across proprietary and public datasets, introducing a feedback propagation method that extends labeled feedback across a graph of related entities, and found that even modest amounts of human feedback produced measurable accuracy gains, particularly for graph-based techniques [12]. This work provides valuable evidence that human feedback improves detection accuracy, but it studies that effect in isolation, on fixed datasets, rather than within an operational system in which the feedback loop is structurally embedded into live transaction processing and scheduled retraining.

Conceptual system architectures for fraud detection have also been proposed. Huy et al. describe a five-layer AI-driven behavioral analytics framework combining behavioral profiling, anomaly detection, machine learning classification, and risk scoring, and explicitly identify the integration of these components into a single operational architecture as an open gap in the literature [14]; however, the framework is presented at the conceptual level, with an evaluation methodology proposed for future empirical validation rather than a working implementation or reported results. Sany et al. instead report a fully implemented and empirically evaluated system, combining Random Forest and XGBoost through a weighted soft-voting ensemble with SMOTE-based resampling and a streaming deployment architecture, achieving strong classification metrics on a benchmark dataset [15]; this work demonstrates a complete, evaluated pipeline, but does not incorporate a human review stage or a feedback-driven retraining mechanism into the deployed system itself. Borketey similarly reports a fully evaluated pipeline, comparing nine classical and ensemble algorithms on a benchmark European credit card dataset, identifying Random Forest as the strongest performer and applying SHAP-based explainability to the result, though again without an integrated real-time deployment or review mechanism [13]. Table I summarizes these representative works alongside the present study.

TABLE I. SUMMARY OF REPRESENTATIVE RELATED WORK

Study	Real Deployment	Human Review	Auto. Retraining	Dataset
Kadam [12]	No	Yes (offline)	No	Proprietary / public
Huy et al. [14]	Conceptual only	No	Proposed only	Not evaluated
Sany et al. [15]	Streaming sim.	No	No	Simulated benchmark
Borketey [13]	No	No	No	Kaggle (European)
Proposed System	Yes	Yes (integrated)	Yes (scheduled)	Synthetic, generated

A. Comparative Analysis with Prior Work

Positioned against the representative studies in Table I, the system described in this paper differs in three respects.

- Real deployment rather than a conceptual or offline pipeline: unlike Huy et al.'s five-layer framework, which is proposed at the conceptual level with evaluation left to future

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722134

work [14], the system here is a working, containerized banking backend that has actually scored, withheld, and processed transactions end to end.

- An integrated, structurally connected review loop: Kadam demonstrates that human feedback improves fraud classifier accuracy, but studies this effect offline, on fixed datasets, separately from any deployed system [12]; the present work instead routes flagged transactions to a live reviewer and structurally connects confirmed outcomes to scheduled retraining.
- Retraining as a first-class, scheduled mechanism: Sany et al. and Borketey both report strong, fully evaluated classification pipelines, but neither incorporates a mechanism for the deployed model to update itself from new outcomes [15], [13]; the present system treats retraining as an automated, auditable part of the architecture rather than a one-time offline training step.

No single prior system in Table I combines real deployment, an integrated human review stage, and scheduled automated retraining simultaneously; the present work's contribution is this combination, rather than a claim of superior raw classification performance on any one metric.

III. PROPOSED SYSTEM ARCHITECTURE

The proposed system is a containerized digital banking backend that scores fraud risk as part of the live transaction path itself, not as a separate offline process run afterward. Figure 1 shows the deployed architecture. Client requests are received through a reverse proxy and routed to the application server, which implements core banking operations including account management, deposits, withdrawals, and fund transfers, alongside the fraud detection and human-review endpoints described in this paper.

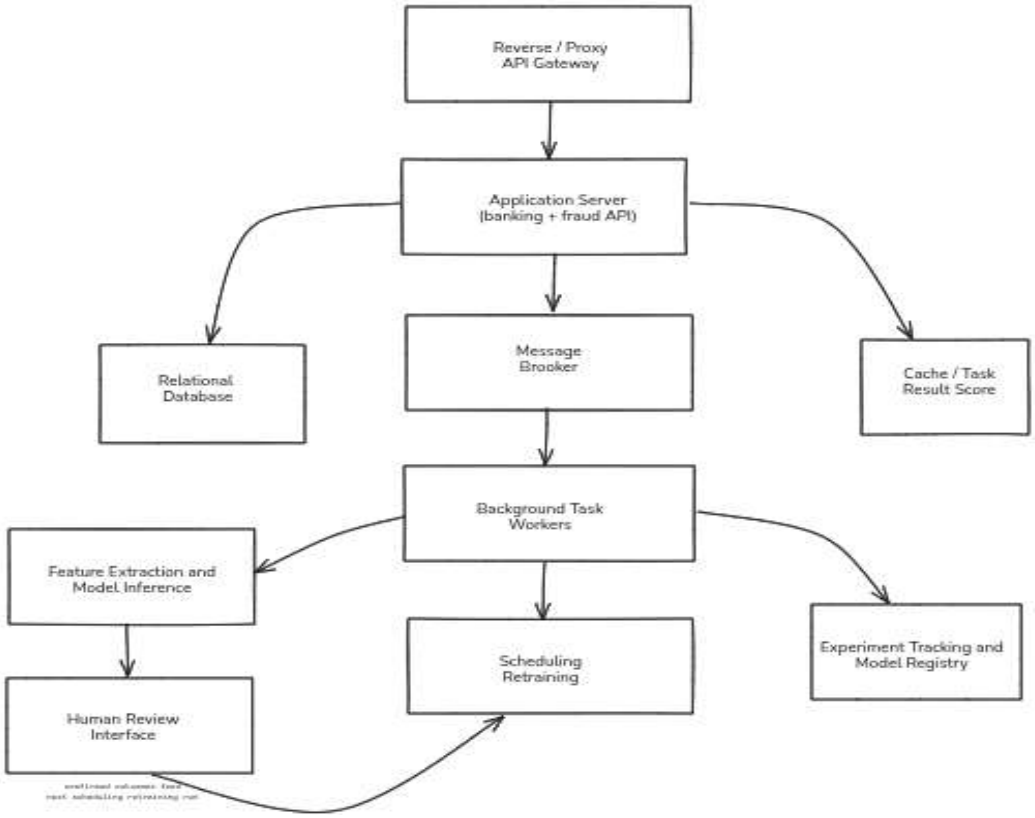

Fig. 1. Deployed system architecture, from client request through transaction processing, background retraining, and human review.

The application server persists transaction and account state to a relational database and dispatches longer-running work, including email notification, statement generation, and model training, to a distributed task queue backed by a message broker, with an in-memory store used for caching and task-result storage. This separation ensures that transaction-processing latency is not affected by background operations such as scheduled model retraining.

Two scheduled background processes are of particular relevance to this paper. First, a feature extraction and inference component evaluates every fund transfer and withdrawal at the moment it is initiated, prior to completion, using the currently deployed classifier. Second, a scheduled retraining process periodically retrains the classifier on newly accumulated transaction data, incorporating labeled outcomes from the human review process described in Section III-A, and logs all training runs, parameters, and metrics to an integrated experiment-tracking and model registry platform [10]. A model is promoted to production only if its held-out performance exceeds that of the currently deployed model, and the previous model is archived rather than discarded, preserving a complete, queryable history of model versions.

A. Human-in-the-Loop Review

Transactions whose fraud probability exceeds a configured threshold are not rejected automatically; instead, the transaction is temporarily withheld and made available to a designated reviewer, who examines the transaction and the specific risk factors that contributed to its score. The reviewer's decision, either confirming the transaction as fraudulent or clearing it as legitimate, is recorded together with free-text justification, and a cleared transaction resumes and completes automatically. Critically, a flagged transaction does not complete and no funds are transferred until this review concludes: the transaction is held mid-flow at the point of initiation, rather than being permitted to finalize and subsequently corrected through a reversal or chargeback, which is how fraud is typically addressed once detected in many deployed banking systems and in much of the reviewed literature. This design reflects the position, also noted in earlier work on active learning for fraud, that human verification is often a practical or regulatory necessity for confirming fraud rather than a discretionary addition [12], while automated scoring alone determines which transactions warrant that verification in the first place.

IV. FEATURE ENGINEERING AND MODEL TRAINING

A. Feature Extraction

When a transfer or withdrawal is initiated, a feature extraction component derives more than forty features spanning five categories, summarized in Table II. This multi-category approach follows the finding that behavioral and temporal context substantially improves detection accuracy beyond raw transaction attributes alone [7].

TABLE II. FEATURE CATEGORIES EXTRACTED PER TRANSACTION

Category	Representative Features
Temporal	Hour of day, day of week, weekend flag, banking-hours flag
Amount / metadata	Transaction amount, currency conversion flag, conversion ratio
Account-level	Sender/receiver balance, account age, historical amount statistics
User history	90-day transaction count, average and maximum amount, type ratios
Velocity	Transaction count and volume over 1-hour, 1-day, 7-day, 30-day windows

B. Class Imbalance Correction

Confirmed fraud constitutes a small minority of all transactions, consistent with the imbalance widely reported in the literature [9]. The extracted feature set is oversampled on the training split only, using SMOTE [11], with the target minority-class ratio treated as a tunable parameter rather than the conventional default of a full one-to-one balance. Section VI reports results at both a default, fully-balancing configuration and a moderated configuration, since the two were found empirically

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722134

to trade recall against precision in a manner consistent with the class-imbalance literature.

C. Model Training

The prepared dataset is partitioned into training and held-out test subsets using a stratified 80/20 split, preserving the original fraud ratio in both subsets. A Gradient Boosting classifier [4] is fit to the resampled training data, using 100 boosting stages, a maximum tree depth of three, a learning rate of 0.1, and subsampling of 0.8 per stage. Model performance is evaluated on the untouched test split using the area under the receiver operating characteristic curve, precision, recall, and F1-score, and a decision-threshold sweep over the precision-recall curve identifies the threshold that maximizes F1-score, reported alongside the conventional 0.5 default in Section VI. Figure 2 summarizes the complete pipeline, from data collection through feature engineering, model training and evaluation, deployment, and the human review feedback loop described in Section III-A.

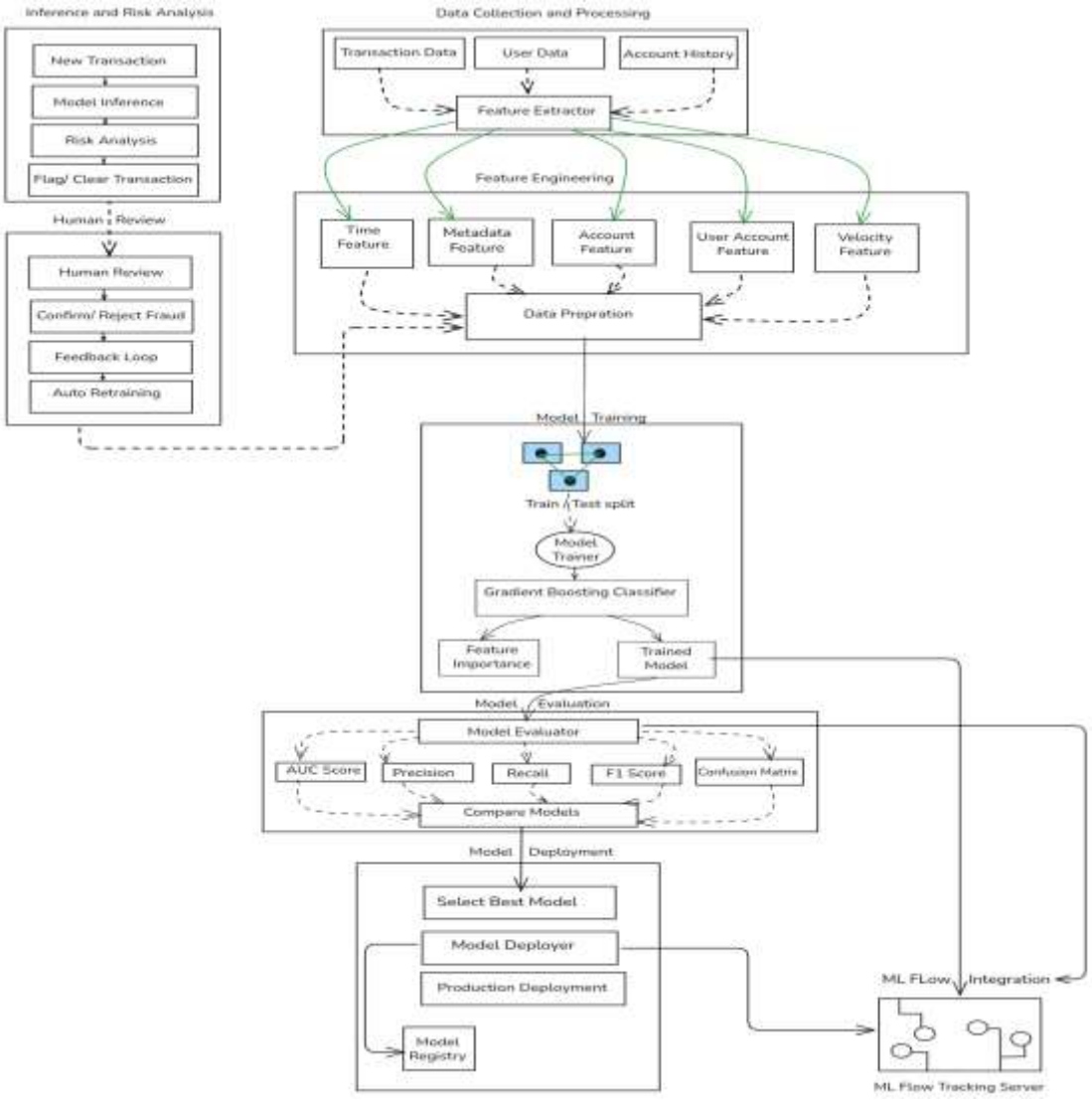

Fig. 2. End-to-end machine learning pipeline, from data collection through model deployment and human-in-the-loop feedback.

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722134

For comparison, Logistic Regression and Random Forest classifiers are trained and evaluated on the identical resampled training set and the identical held-out test set, using class-balanced weighting, to establish whether Gradient Boosting's performance is attributable to the algorithm itself rather than to the shared feature set and resampling procedure.

V. EXPERIMENTAL SETUP

Due to the confidentiality and regulatory constraints governing real banking transaction data, this study employs a synthetically generated transaction dataset produced by a custom simulation pipeline within the deployed system itself. This is consistent with established practice in this research area, where widely used benchmark datasets are themselves synthetic or heavily transformed rather than raw financial records [3], [13]. The simulation generates users, bank accounts, and transactions with randomized but realistic amounts, timestamps, and counterparties, with a configurable subset of transfers labeled as confirmed fraud to emulate the outcome of the human review process described in Section III-A. Two dataset configurations were generated to examine the sensitivity of the trained model to dataset scale and fraud prevalence, summarized in Table III.

TABLE III. DATASET CONFIGURATIONS

Configuration	Transactions	Confirmed Fraud	Fraud Ratio
Small (initial)	987	10	≈1.0%
Large (expanded)	5,957	200	≈3.4%

All experiments were conducted on the deployed system's own infrastructure, with training, evaluation, and model registration executed as an asynchronous background task and logged in full to the experiment-tracking platform, so that every reported result in this paper corresponds to an auditable training run rather than an offline notebook experiment.

VI. RESULTS AND DISCUSSION

A. Training Progression on the Initial Dataset

Three successive training runs were conducted on the small dataset configuration over the course of development. Table IV summarizes the results, and Fig. 3 illustrates the progression across all four evaluated metrics.

TABLE IV. METRIC PROGRESSION ACROSS TRAINING RUNS (SMALL DATASET)

Run	Status	AUC	Precision	Recall	F1
1 (initial)	Archived	0.4746	0.0000	0.0000	0.0000
2 (revised)	Archived	0.5252	0.1667	0.1538	0.1600
3 (final)	Deployed	0.7649	0.7200	0.5806	0.6429

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722134

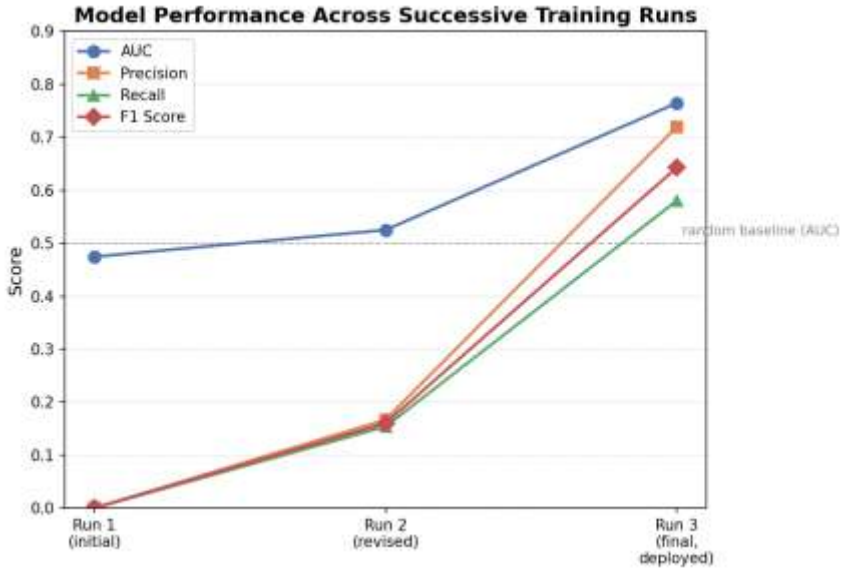

Fig. 3. Metric progression across three successive training runs on the small dataset.

The first run produced an AUC of 0.4746, marginally below the value expected of a non-discriminative classifier, together with precision, recall, and F1-score of zero, indicating that the classifier assigned no transaction a positive prediction on the held-out test set. This was traced to severe class imbalance in the training data, which at that stage contained only ten confirmed fraud cases out of 987 transactions; following the stratified split, the resulting training set left too few positive examples for the classifier to learn a useful decision boundary, and a constant "not fraud" prediction would itself have achieved 99% raw accuracy on such data despite being entirely uninformative. Correcting the class-balancing procedure across the subsequent two runs produced a substantial and consistent improvement, with the final run achieving an AUC of 0.7649, a precision of 0.72, and a recall of 0.58, as shown by the rightmost point in Fig. 3. This progression is reported in full, including the initial non-functional result, because it constitutes direct evidence of the iterative, diagnosis-driven methodology described in Section IV, rather than a single reported outcome selected after the fact.

B. Comparison Against Baseline Classifiers

To determine whether the observed performance is attributable to Gradient Boosting specifically, rather than to the shared feature set, Logistic Regression and Random Forest classifiers were trained and

evaluated under identical conditions on the expanded dataset configuration (5,957 transactions, 200 confirmed fraud cases). Table V and Fig. 4 report the comparison.

TABLE V. COMPARISON OF GRADIENT BOOSTING AGAINST BASELINE CLASSIFIERS (LARGE DATASET)

Model	AUC	Precision	Recall	F1
Gradient Boosting	0.9430	0.1892	0.6667	0.2947
Logistic Regression	0.9131	0.0820	0.7619	0.1481
Random Forest	0.9472	0.9480	0.0953	0.1143

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722134

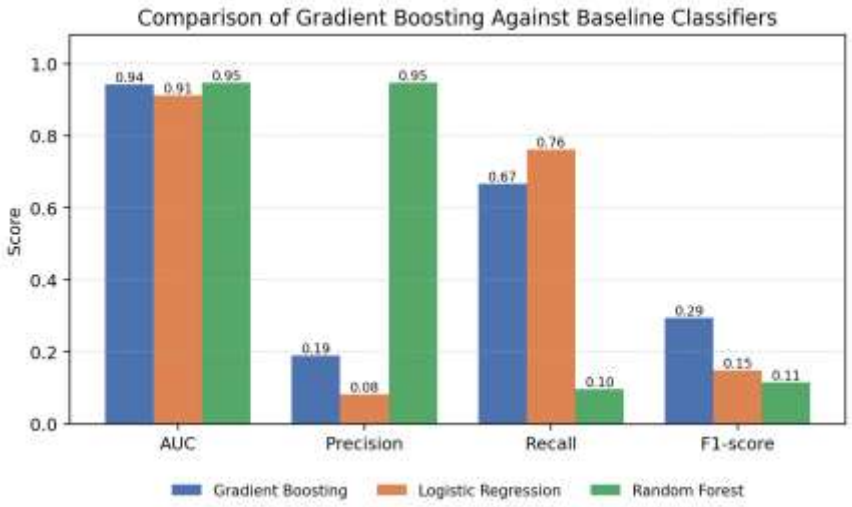

Fig. 4. Gradient Boosting compared against Logistic Regression and Random Forest baselines.

All three classifiers achieve an AUC above 0.91, indicating that the extracted feature set itself carries strong discriminative signal largely independent of the choice of classifier, consistent with the emphasis placed on feature engineering in prior work [7]. The three models nonetheless differ sharply in their precision-recall trade-off. Random Forest achieves the highest precision (0.948) but a very low recall (0.095), indicating an overly conservative classifier that misses the great majority of actual fraud cases. Logistic Regression exhibits the opposite behavior, achieving the highest recall (0.762) at the cost of very low precision (0.082), indicating substantial over-flagging of legitimate transactions. Gradient Boosting occupies a middle position, with the second-highest recall (0.667) and a precision (0.189) markedly higher than Logistic Regression's, yielding the most balanced trade-off of the three models on this configuration, though not the highest score on any single metric in isolation. For fraud detection specifically, this matters more than it might first appear: choosing a model here is really a decision about which mistake the business can tolerate more, a false alarm or a missed fraud case, and AUC alone does not capture that trade-off.

C. Effect of Resampling Ratio and Threshold Tuning

The precision obtained on the large dataset (0.189) is markedly lower than that obtained on the small dataset (0.72), despite a higher AUC. This is attributed to the default SMOTE configuration fully balancing the training set to a one-to-one class ratio; with 200 real fraud examples available for interpolation rather than ten, this produced a substantially larger synthetic minority class and a correspondingly more aggressive classifier. Two corrective adjustments were evaluated: reducing the SMOTE target ratio to 0.3 rather than full balance, and selecting a decision threshold that maximizes F1-score on the precision-recall curve rather than using the conventional 0.5 default. Table VI reports the effect of each adjustment.

TABLE VI. EFFECT OF RESAMPLING RATIO AND THRESHOLD SELECTION ON GRADIENT BOOSTING (LARGE DATASET)

Configuration	AUC	Precision	Recall	F1
Full SMOTE balance, default threshold	0.9430	0.1892	0.6667	0.2947
Moderated SMOTE (0.3), default threshold	0.9164	0.2444	0.5238	0.3333
Moderated SMOTE (0.3), tuned threshold (0.480)	0.9164	0.2449	0.5714	0.3429

Moderating the resampling ratio recovers a meaningful share of precision (0.189 to 0.244) and improves F1-score (0.295 to 0.333), at a measured

cost to AUC and recall, consistent with the expected effect of reducing the volume of synthetic minority examples used during training. Threshold tuning subsequently identified an F1-optimal cutoff of 0.480, close to the conventional default of 0.5, and produced a further, smaller improvement in recall (0.524 to 0.571) with F1-score rising to 0.343. The near-default optimal threshold indicates that, for this configuration, the model's default operating point was already close to F1-optimal, and that the SMOTE ratio was the more consequential parameter of the two. Considered together with Section VI-A, these results indicate that dataset scale, class-balancing ratio, and decision threshold each materially affect the resulting precision-recall trade-off, and that no single configuration in this study dominates across all four metrics simultaneously; this trade-off, rather than a single headline number, is treated as the primary empirical finding of this evaluation.

VII. LIMITATIONS

Several limitations affect how the results in this paper should be read. The evaluation relies on synthetically generated transaction data rather than real banking transactions, a choice driven by the confidentiality and regulatory constraints discussed in Section V. The simulation was built to reflect realistic behavioral and temporal patterns, but whether performance holds up on genuine transaction data is a question this study cannot settle. Dataset size is a related concern: both configurations, and particularly the small number of confirmed fraud cases relative to real-world production volumes, constrain how statistically robust the reported metrics can be considered; larger and more varied labeled data would give more stable estimates. There is also a gap between what the system is designed to do and what has been fully implemented and measured so far — the architecture supports feeding confirmed human-review outcomes back in as ground-truth labels for post-deployment monitoring, but this loop is not yet connected in the current build, and the results reported here come from a single held-out test split rather than a cross-validated estimate. Closing that loop, alongside cross-validated evaluation on a larger and more diverse dataset, is the most immediate next step for this work.

VIII. CONCLUSION AND FUTURE WORK

This paper set out to close a gap between how fraud detection is usually evaluated and how it actually needs to run in production: as a live, closed-loop system rather than a classifier tested once against a fixed dataset. The system built and evaluated here scores every fund transfer and withdrawal in real time, routes high-risk transactions to a human reviewer before they complete, and feeds confirmed reviewer outcomes back into a scheduled retraining pipeline — so human judgement and model improvement are structurally linked rather than handled as separate concerns. Across the two dataset configurations tested, the deployed Gradient Boosting classifier reached an AUC as high as 0.94, and its best precision-recall balance (0.72 precision, 0.58 recall) came on the smaller dataset. Comparisons against Logistic Regression and Random Forest, together with the sensitivity analysis of resampling ratio and decision threshold in Section VI, showed Gradient Boosting striking a more even balance between the two baselines rather than dominating on any single metric. Going forward, the priority is connecting confirmed human-review outcomes into post-deployment evaluation as real ground-truth labels, growing the training dataset in both scale and diversity, and — where regulatory and confidentiality constraints allow — testing the system against real-world transaction data.

REFERENCES

[1] E. Minastireanu and G. Mesnita, "An analysis of the most used machine learning algorithms for online fraud detection," Informatica Economica, vol. 23, no. 1, 2019.

[2] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in Proc. Int. Conf. Computing Networking and Informatics (ICCNI), 2017, pp. 1-9.

[3] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Credit card fraud detection — Machine learning methods," in

Proc. 18th Int. Symp. INFOTEH-JAHORINA, 2019, pp. 1-5.

[4] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 2001.

[5] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[6] S. K. Hashemi, S. L. Mirtaheri, and S. Greco, "Fraud detection in banking data by machine learning techniques," IEEE Access, vol. 10, pp. 1-15, 2022.

[7] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," Expert Systems with Applications, vol. 51, pp. 134-142, 2016.

[8] M. F. Manzoor and M. F. Aslam, "Enhancing banking fraud detection: Role of machine learning and deep learning methods," International Journal of Artificial Intelligence, 2025.

[9] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in Proc. IEEE Symp. Series on Computational Intelligence, 2015, pp. 159-166.

[10] M. Zaharia et al., "Accelerating the machine learning lifecycle with MLflow," IEEE Data Engineering Bulletin, vol. 41, no. 4, pp. 39-45, 2018.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[12] P. Kadam, "Enhancing financial fraud detection with human-in-the-loop feedback and feedback propagation," arXiv preprint arXiv:2411.05859, 2024.

[13] B. Borketey, "Real-time fraud detection using machine learning," Journal of Data Analysis and Information Processing, vol. 12, pp. 189-209, 2024.

[14] S. Huy, S. Ang, M. Ho, and V. Balasubramaniam, "An AI-driven behavioral analytics framework for real-time fraud detection in financial institutions," in Proc. Int. Conf. Recent Innovations in Science and Technology (ICRIST), 2026.

[15] M. Z. I. Sany, Z. Wubo, and S. Alam, "Optimized hybrid machine learning model for real-time financial fraud detection," Control Systems and Optimization Letters, vol. 4, no. 2, pp. 205-210, 2026.