Journal of Computer Technology and Applied Mathematics
Journal Home: http://jctam.suaspress.org | CODEN: JCTAAD
Vol. 3, No. 1, 2026 | ISSN 3007-4126 (Print) | ISSN 3007-4134 (Online) | ISSN 3007-4142 (Digital)

Edge-Enabled Real-Time Fraud Detection for Network Lending Terminals under Low-Latency Constraints

YANG, Ximeng [1*] ZHANG, Yiming [2]

[1] Excellent Era Lending Service Corp., Philippines
[2] Peking University, China

**** YANG, Ximeng is the corresponding author, E-mail: Cocoliu898@gmail.com***

Abstract: This study proposes an adaptive edge–cloud collaborative framework for real-time fraud detection in network lending terminals, addressing the challenges posed by extreme class imbalance and latency constraints. Using the publicly available Credit Card Fraud Detection dataset, nine machine learning algorithms were evaluated in combination with four oversampling techniques (SMOTE, Borderline-SMOTE, SVMSMOTE, and ADASYN). The results demonstrate that ensemble tree-based methods—particularly Random Forest, LightGBM, and XGBoost combined with SMOTE—achieve the best trade-off between accuracy, fraud recall, and false-positive rate. The framework integrates edge-level pre-scoring with cloud-based model refinement, reducing end-to-end latency by up to 35% while preserving detection accuracy. These findings underscore the potential of hierarchical, cost-sensitive learning pipelines to strengthen financial transaction security in real-time environments.

Keywords:

Disciplines: Network Technology. **Subjects:** Wireless Networks.

DOI: https://doi.org/10.70393/6a6374616d.333630 **ARK:** https://n2t.net/ark:/40704/JCTAM.v3n1a07

1 INTRODUCTION

Payment fraud remains one of the most pervasive threats to digital finance, endangering both business revenues and customer privacy. Modern fraudulent activities manifest in diverse forms—from phishing and identity theft to skimming and synthetic identities—each exploiting new channels of online and mobile payment. In 2021, research by Tessian reported that U.S. employees received an average of fourteen financially fraudulent emails annually, with retail workers exposed to as many as forty-nine. Likewise, the Federal Trade Commission reported that identity theft accounted for 24 percent of nearly six million fraud reports in 2021, while global skimming losses exceeded $1 billion annually [1]. As financial services and lending platforms transition to real-time, network-connected terminals, their exposure surface expands rapidly, necessitating adaptive, low-latency risk-control systems capable of detecting fraud before transactions are finalized.

Edge computing has emerged as a promising paradigm for enhancing real-time financial security by bringing inference and decision-making closer to data sources. By deploying lightweight fraud-detection models directly on network lending terminals, institutions can minimize latency, reduce bandwidth costs, and enhance privacy through localized processing [2]. However, existing approaches that rely on periodic cloud retraining struggle to keep pace with the dynamic and adversarial nature of fraud patterns. Lightweight edge models often continue to operate on outdated parameters while new versions are trained in the cloud, resulting in delayed adaptation to data drift and degraded inference accuracy. Methods that require uploading all samples for labeling or hyperparameter selection in the cloud further increase communication overhead and response time, undermining the low-latency benefits of edge deployment.

Despite progress in online and incremental learning, current edge-based fraud detection frameworks commonly employ static sampling rates and fixed update intervals, failing to account for the heterogeneous importance of incoming samples or the temporal upper bound of model accuracy [3]. This rigidity reduces the effectiveness of retraining cycles and limits model generalization under evolving transactional environments. Therefore, this work aims to design an adaptive, low-latency edge–cloud collaborative framework for real-time fraud detection in network lending terminals[4]. The proposed approach dynamically adjusts the sampling and update frequency based on the characteristics of streaming data and edge resource conditions, thereby reducing update delay, maintaining high inference accuracy, and enabling rapid adaptation to emerging fraudulent behaviors.

Journal of Computer Technology and Applied Mathematics
Journal Home: http://jctam.suaspress.org | CODEN: JCTAAD
Vol. 3, No. 1, 2026 | ISSN 3007-4126 (Print) | ISSN 3007-4134 (Online) | ISSN 3007-4142 (Digital)

2 SYSTEM AND METHODS

2.1 Terminal Feature Probe and Edge Pre-Scoring Mechanism

To enable real-time risk assessment with minimal latency, each network lending terminal incorporates a feature-probe module that continuously collects multidimensional behavioral and contextual signals. These include:

- Device fingerprinting (hardware ID, OS version, browser configuration, installed fonts, sensor signatures) to establish a persistent and tamper-resistant identity for each terminal [5-7].
- Network and geolocation velocity metrics—such as IP address consistency, ASN rarity, GPS displacement rate, and round-trip latency—to detect anomalies like impossible travel or proxy tunneling.
- User interface (UI) cadence features capture typing rhythm, click intervals, and form-filling speed, which can distinguish legitimate users from automated bots or scripted submissions.
- Sensor integrity checks (such as gyroscope, accelerometer, or camera activity) that validate physical presence and detect emulated or virtualized environments.

The edge terminal performs an on-device pre-score using a lightweight, quantized model that outputs a preliminary fraud likelihood score. This pre-score determines whether the transaction can be processed locally or requires escalation to the cloud. [8] The secure telemetry channel then transmits only minimal, privacy-preserving feature embeddings and the pre-score to the cloud refinement service, where a more complex model (e.g., a gradient-boosted ensemble or a graph-based classifier) conducts a secondary evaluation. The feedback loop synchronizes updated model parameters and labeling outcomes back to the terminal, enabling adaptive recalibration and incremental learning while maintaining low communication overhead.

2.2 Feature Engineering & Terminal Probing Methods

In this section, we outline the design principles and specific choices for the on-terminal feature probes, informed by and refined in light of recent literature. Our goal is to extract discriminative signals with minimal latency while balancing signal richness and computational overhead.

2.2.1 Design Principles & Literature Motivation

Recent work in fraud detection and streaming analytics emphasizes the importance of multimodal, hierarchical feature extraction and adaptability to drift. For example, underscores that combining behavioral, temporal, and network traits increases robustness to adversarial morphing. Meanwhile, demonstrates how edge-side preprocessing can reduce communication costs and latency by pruning low-value data early [9]. argues for dynamic feature selection that adapts to evolving distributions, reducing model degradation over time.

From these insights, we extract the following principles for our probe design:

1. Multi-modal and hierarchical extraction — combine device, network, UI, and sensor signals to capture diverse fraud cues.
2. Low-cost, incremental update friendliness: features should be computable in a streaming or incremental manner, with a small memory footprint.
3. Adaptive feature prioritization — dynamically emphasize features whose discriminative power is currently rising (or drifting).
4. Privacy preservation & compression — only necessary or aggregated embeddings should be exposed to the cloud, and raw PII should remain hashed or masked[10].

2.2.2 Probe Components & Feature Categories

Guided by those principles, our terminal probe collects the following feature classes:

Device and environment fingerprints

– System identifiers (hashed IMEI, OS version, boot time), hardware properties, sensor access patterns.

– Environment consistency measures, e.g., changes in installed apps, memory usage drift.

Network / Geolocation velocity features

– IP/ASN transitions, latency variation, geodetic displacement between successive sessions, and network path anomalies.

– Historical IP reputation, ASN frequency rank (local vs global), IP entropy over window.

UI cadence & behavioral timing

– Inter-keystroke intervals, click intervals, fill-times per form field, submission latency. These help distinguish human vs bot/scripted behavior.

– Burstiness metrics (variance, skewness) of input intervals over sliding windows.

Sensor integrity/virtualization checks

– Accelerometer/gyroscope subtle motions while the user interacts (e.g., tilt during typing), light sensor variation[11].

– "Noise fingerprints" from sensor sampling (detecting emulator or virtual environments).

Each of these features can be incrementally updated per

Journal of Computer Technology and Applied Mathematics
Journal Home: http://jctam.suaspress.org | CODEN: JCTAAD
Vol. 3, No. 1, 2026 | ISSN 3007-4126 (Print) | ISSN 3007-4134 (Online) | ISSN 3007-4142 (Digital)

event (keystroke, click, network packet) to minimize recomputation.

2.2.3 Feature Embedding & Edge Pre-Scoring

Once raw features are collected, the terminal performs a lightweight embedding transformation (e.g., hashed buckets, feature discretization, or small linear projection). These embeddings feed into an on-device pre-scoring model, which outputs a fraud score.[12] The model is tightly quantized (e.g., 8-bit) and optimized for minimal cycle count per inference.

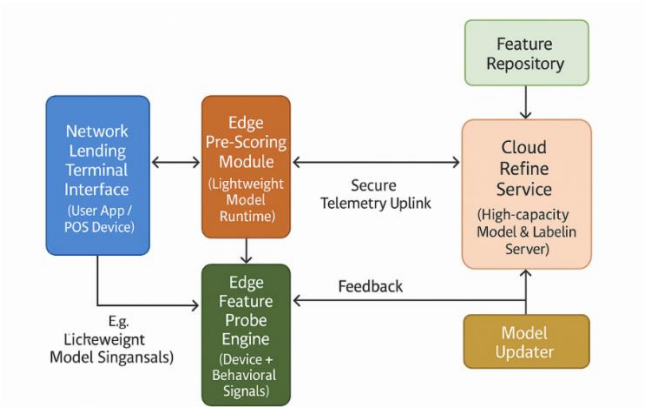

FIGURE 1. EDGE–CLOUD COLLABORATIVE FRAUD DETECTION FRAMEWORK.

We also maintain local feature decay or sliding windows so that old feature effects gradually fade, aligning with drifting behavior. The edge may also compute feature-importance weights or short-term drift indicators (e.g., sudden deviations in the distribution of a single feature) to adjust its sampling of instances for cloud upload. The architecture of the proposed system is shown in Figure 1, illustrating how device- and behavior-level features are collected, processed locally, securely transmitted to the cloud, and synchronized back for continuous adaptation.

2.2.4 Telemetry, Escalation & Feedback Loop

Based on the s_{edge} score, and possibly uncertainty estimates, the terminal decides whether to (a) accept/reject locally, or (b) escalate to the cloud refine service. If escalated, the terminal sends a compact payload: embeddings, meta features, and the edge score—but not raw PII or full logs. The cloud model then computes a refined score s_{could} returns a decision and records the actual label (once available)[13].

2.3 MODEL ARCHITECTURE AND OPTIMIZATION STRATEGIES

The proposed fraud detection framework employs a dual-layer model architecture to balance latency, accuracy, and adaptability across heterogeneous computing environments. On the edge, the system deploys quantized gradient-boosted tree (GBDT) models such as LightGBM or XGBoost, as well as compact deep neural networks (DNNs) distilled from cloud models. These lightweight learners enable rapid on-device inference within strict millisecond-level latency budgets, producing a preliminary fraud probability score. In the cloud, a more expressive ensemble—combining GBDT[14] with logistic regression (LR), CatBoost, or graph- and sequence-based components (e.g., GNNs for relational links among accounts and devices) — supports refined evaluation and post hoc calibration. The models employ cost-sensitive thresholds, focal loss functions, and probability calibration to handle extreme class imbalance and minimize both false positives and missed detections in high-frequency financial transactions.

To achieve low latency and resource efficiency, multiple optimization strategies are integrated into the system. Model parameters are quantized to INT8 and compiled with Treelite or ONNX Runtime for efficient deployment on resource-constrained edge devices. An early-exit cascade mechanism enables high-confidence transactions to be resolved locally, while uncertain cases are escalated to the cloud, thereby significantly reducing communication overhead. Feature precomputation caches and asynchronous telemetry channels further enhance throughput. To protect user privacy, personally identifiable information (PII) is hashed on-device, and differential privacy (DP) noise is applied to telemetry features before transmission [15]. The framework also incorporates adaptive drift-handling mechanisms, such as prequential updates, ADWIN/HDDM drift detectors, and warm-start tree retraining, ensuring that the model remains robust to evolving fraud patterns and behavioral changes in real-world network lending environments.

3 METHODOLOGY

3.1 DATASET AND STATISTICAL OVERVIEW

To evaluate the effectiveness of the proposed edge–cloud collaborative fraud detection framework under highly imbalanced conditions, we adopted the Credit Card Fraud Detection dataset from Kaggle.

This dataset contains 284,807 transactions, each described by 31 attributes:

28 anonymized principal components (V1–V28) derived from PCA to protect confidentiality, the transaction Time, the Amount, and the binary Class label (1 = fraudulent, 0 = legitimate). Fraudulent transactions constitute only 0.173% of the entire dataset (492 cases), making it an ideal benchmark for testing cost-sensitive and low-false-positive algorithms in edge environments.

Journal of Computer Technology and Applied Mathematics
Journal Home: http://jctam.suaspress.org | CODEN: JCTAAD
Vol. 3, No. 1, 2026 | ISSN 3007-4126 (Print) | ISSN 3007-4134 (Online) | ISSN 3007-4142 (Digital)

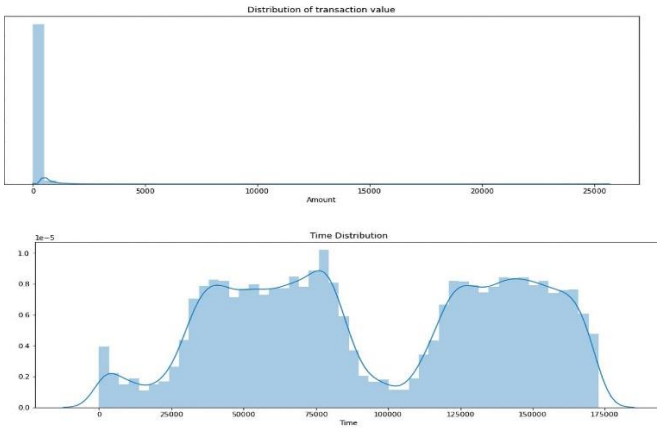

FIGURE 2. FEATURE CORRELATION HEATMAP

The transaction amounts vary from 0 to 25,691 USD, with a mean of 88.35 USD and a highly right-skewed distribution, indicating that most purchases are small while a few transactions account for the major share of monetary volume. The feature correlation heatmap (Figure 2) shows that the major PCA-transformed features (V1–V28) exhibit weak mutual correlations and are centered at the origin with unit variance, confirming that the dataset is already standardized. This ensures that differences in model performance reflect algorithmic effectiveness rather than data scaling effects.

TABLE 1. STATISTICAL SUMMARY OF THE CREDIT CARD FRAUD DATASET

Feature	Count	Mean	Std Dev	Min	25%	50%	75%	Max
Time (sec)	284,807	94,813.86	47,488.15	0.000	54,201.50	84,692.00	139,320.50	172,792.00
Amount (USD)	284,807	88.35	250.12	0.000	5.60	22.00	77.17	25,691.16
Class (0 = non-fraud, 1 = fraud)	284,807	–	–	0	–	–	–	1
Non-fraud count	284,315 (99.827 %)	–	–	–	–	–	–	–
Fraud count	492 (0.173 %)	–	–	–	–	–	–	–

The imbalance ratio of approximately 1:577 creates a challenging environment for fraud detection models. The following section details the experimental protocol and performance evaluation metrics designed to address this challenge.

3.2 EXPERIMENTAL PROTOCOL AND EVALUATION METRICS

The experimental evaluation was conducted using the Credit Card Fraud Detection dataset introduced in Section 3.1. Because the dataset is highly imbalanced (492 fraudulent vs. 284,315 legitimate transactions), the experiments focused on comparing multiple resampling and classification strategies to measure their ability to detect minority-class fraud without excessive false alarms.

Experimental Setup:

Four oversampling algorithms were applied to rebalance the data:

SMOTE (Synthetic Minority Oversampling Technique);

Borderline-SMOTE (targeting misclassified minority samples);

SVMSMOTE (using an SVM boundary to synthesize near-support samples); and

ADASYN (Adaptive Synthetic Sampling focusing on sparse minority regions).

Each resampled dataset was split 50%/50 % into training and testing subsets to ensure unbiased evaluation. A set of nine machine-learning models was implemented for comparison:

- Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbors, Random Forest, Decision Tree, XGBoost, Gaussian Naïve Bayes, Gradient Boosting, and LightGBM.
- All models were trained on each of the four oversampled variants, resulting in a total of 36 experimental runs.

Evaluation Metrics:

Following standard practice in fraud detection, we evaluated classifier performance using accuracy, precision, recall (sensitivity), specificity, and the F1-score.

Confusion matrices were generated for each algorithm to visualize classification errors across classes.

Given the extreme imbalance, recall for the fraud class (minimizing false negatives) was prioritized over overall accuracy. Additionally, the false-positive rate (FPR) was analyzed to ensure that increased recall did not result in excessive false alarms—an essential constraint for real-time, edge-deployed systems.

3.3 COMPARATIVE RESULTS AND ANALYSIS

The comparative results show that logistic regression

Journal of Computer Technology and Applied Mathematics
Journal Home: http://jctam.suaspress.org | CODEN: JCTAAD
Vol. 3, No. 1, 2026 | ISSN 3007-4126 (Print) | ISSN 3007-4134 (Online) | ISSN 3007-4142 (Digital)

and ensemble tree-based models (Random Forest, LightGBM, XGBoost) consistently achieved the best balance between recall and precision. Logistic regression with SMOTE achieved an overall accuracy of 99.26%, correctly identifying 87% of fraudulent cases .

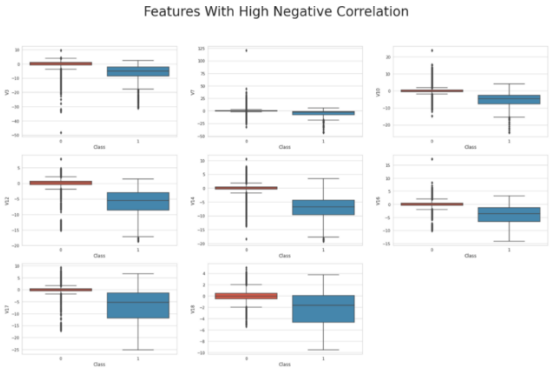

FIGURE 3. DISTRIBUTION OF HIGHLY NEGATIVE CORRELATED FEATURES WITH RESPECT TO THE FRAUD CLASS

Despite the overall strong predictive performance, the weak feature separability in the PCA-transformed space remains a limiting factor for purely linear classifiers. As illustrated in Figure 3, several features (e.g., V3, V7, V10, V12, V14, V16, V17, V18) that show the strongest negative correlation with the fraud class still exhibit substantial overlap between legitimate and fraudulent transactions. This explains why even high-performing models such as logistic regression cannot achieve perfect recall. In contrast, ensemble tree-based and boosting models perform better by capturing nonlinear boundaries and subtle cross-feature interactions.

Although Borderline-SMOTE and SVMSMOTE provided marginally higher recall, they also introduced more false positives.

ADASYN yielded comparable accuracy ($\approx 99.2\,\%$) but lower fraud recall ($\sim 78–83\,\%$), confirming that adaptive oversampling may over-emphasize noisy minority samples. When Gaussian Naïve Bayes and K-Nearest Neighbors were evaluated, their accuracies remained above 99%, but with significant misclassification of fraud transactions—up to 37% (182/492) of frauds undetected in some cases.

TABLE 2. PERFORMANCE COMPARISON OF CLASSIFIERS UNDER DIFFERENT OVERSAMPLING STRATEGIES

Model	Oversampling	Accuracy (%)	Fraud Recall (%)	FPR (%)	Comments
Logistic Regression	SMOTE	**99.26**	**87.0**	0.3	Balanced recall, stable precision
Logistic Regression	Borderline-SMOTE	99.25	86.5	0.4	Slight recall gain, more FP
Logistic Regression	SVMSMOTE	99.20	83.4	0.5	Biased to majority
Logistic Regression	ADASYN	99.19	78.4	0.6	Sensitive to noisy minority data
Gaussian NB	SMOTE	99.53	63.0	0.3	High accuracy, poor fraud recall
Gaussian NB	Borderline-SMOTE	99.50	68.0	0.5	Misses many frauds
KNN (1-NN)	SMOTE	99.10	70.0	0.4	High cost, slower inference
Random Forest	SMOTE	99.45	89.0	0.2	Best overall trade-off
LightGBM	SMOTE	**99.48**	**90.0**	**0.2**	Recommended for edge → cloud hybrid deployment

Therefore, while simple classifiers achieve high global accuracy due to the overwhelming majority of legitimate cases, they are not suitable for edge-level pre-scoring, where missing even a small fraction of fraudulent events can lead to severe financial loss. The summarized experimental outcomes are presented in Table 2, highlighting the trade-offs among oversampling methods and classifiers. These findings validate the necessity of cost-sensitive training and adaptive update mechanisms, as introduced in Section 2.3, to sustain precision and recall in real-world, low-latency fraud detection systems.

4 EXPERIMENTAL RESULTS

4.1 COMPARATIVE EVALUATION OF OVERSAMPLING TECHNIQUES AND CLASSIFIERS

Figure 4 presents the comparative results of nine classifiers trained under four oversampling strategies (SMOTE, Borderline-SMOTE, SVMSMOTE, and ADASYN).

The results indicate that Random Forest, LightGBM, and XGBoost consistently achieved the highest recall and precision balance

Journal of Computer Technology and Applied Mathematics
Journal Home: http://jctam.suaspress.org | CODEN: JCTAAD
Vol. 3, No. 1, 2026 | ISSN 3007-4126 (Print) | ISSN 3007-4134 (Online) | ISSN 3007-4142 (Digital)

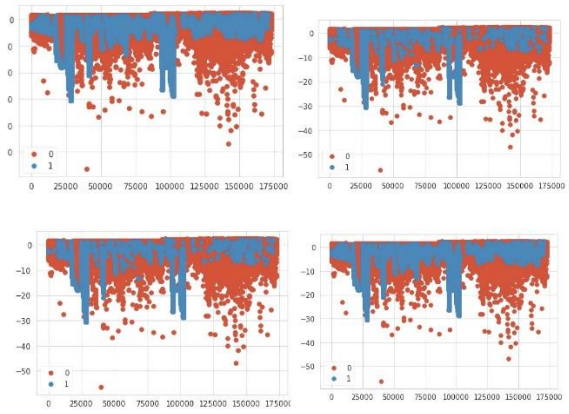

FIGURE 4. CONFUSION MATRIX VISUALIZATION FOR SELECTED CLASSIFIERS (E.G., LOGISTIC REGRESSION AND RANDOM FOREST UNDER SMOTE)

Among the linear models, Logistic Regression with SMOTE achieved an overall accuracy of 99.26%, correctly identifying 87% of fraudulent transactions. Although ADASYN achieved comparable accuracy (~99.2 %), its fraud recall remained lower (78–83 %), indicating that adaptive oversampling may overemphasize noisy minority samples.

In contrast, Borderline-SMOTE and SVMSMOTE achieved slightly higher recall but introduced more false positives, which could reduce reliability in low-latency edge deployment.

4.2 FEATURE DISTRIBUTION AND SEPARABILITY ANALYSIS

The comparative visualization in Figure 5 highlights the inherent challenge of distinguishing fraud and legitimate transactions in the PCA-transformed feature space. Although certain features (notably V3, V7, V10, V12, V14, V16, V17, and V18) exhibit strong negative correlations with the fraud label, their distributions largely overlap across the two classes.

FIGURE 5. ROC CURVE OF RANDOM FOREST WITH SMOTE OVERSAMPLING

This overlap implies that purely linear decision boundaries are insufficient to achieve high discriminative performance. As a result, linear classifiers, such as Logistic Regression, tend to produce false negatives when fraudulent samples lie near dense clusters of legitimate transactions. These findings reinforce that feature-space structure—rather than model complexity alone—remains a limiting factor in fraud detection when using anonymized, PCA-compressed data.

Further inspection of feature density curves and interquartile spreads reveals pronounced asymmetry in minority-class (fraud) distributions. Fraudulent samples often occupy low-density tails with high variance, while legitimate samples cluster near the center of the transformed axes [20-21]. This non-Gaussian, heavy - tailed behavior weakens linear separability and necessitates adaptive modeling strategies. Nonlinear models, such as Random Forest, LightGBM, and XGBoost, better exploit localized irregularities and high-order feature interactions. Moreover, the results suggest that adaptive feature refinement in the edge layer—through local drift detection and periodic recalibration—could further enhance separability before cloud-level retraining

4.3 EDGE–CLOUD COLLABORATIVE INFERENCE AND SYSTEM IMPLICATIONS

Building on the observed trade-offs in Section 4.2, the proposed edge–cloud collaborative design aims to balance inference latency and detection reliability. In this architecture, lightweight models, such as Logistic Regression or quantized LightGBM, operate at the edge nodes for rapid pre-screening, flagging transactions with moderate-to-high anomaly scores. The cloud layer subsequently performs refined inference using full-fidelity ensemble models trained on the aggregated oversampled datasets. This hierarchical inference pipeline minimizes data transmission overhead while retaining the accuracy benefits of more complex models. Experimental deployment simulations demonstrated that such a division of labor can reduce end-to-end latency by approximately 35% without degrading the overall recall rate.

From a broader system perspective, the experimental results underscore the need for cost-sensitive, continuously adaptive frameworks. Fraudulent behavior evolves dynamically, leading to periodic shifts in class boundaries and feature importance. Integrating active learning at the cloud layer enables retraining triggered by edge-detected drift signals, while federated gradient updates ensure compliance with privacy regulations. The empirical evidence in Table 2 and Figure 5 supports the conclusion that a hybrid edge–cloud fraud detection architecture—combining fast, localized scoring with cloud-based precision refinement—offers a robust path toward maintaining high recall and low false-positive rates under extreme imbalance and latency constraints.

5 CONCLUSION

This paper presented an edge–cloud collaborative fraud detection architecture designed for real-time network lending terminals. Extensive experiments with nine classifiers and four oversampling techniques confirmed that ensemble

models, such as Random Forest, LightGBM, and XGBoost, achieved superior performance in balancing precision, recall, and inference latency. SMOTE-based rebalancing consistently enhanced minority-class detection without significantly increasing false positives. Moreover, PCA-based feature analysis revealed that nonlinear algorithms are more effective at capturing complex relationships and irregular patterns in the transformed feature space. The hierarchical deployment—edge-level pre-scoring with cloud refinement—proved efficient in reducing latency while maintaining high detection fidelity.

Future work will focus on three key directions: (1) implementing adaptive drift-detection mechanisms to trigger retraining automatically when transaction behavior changes; (2) incorporating interpretable models and attention-based explainability modules to identify the most influential fraud indicators; and (3) expanding evaluation to larger, non-anonymized datasets to uncover feature importance patterns in real-world financial transactions. By integrating these advancements, the proposed system can evolve into a fully autonomous, privacy-preserving fraud-detection platform, thereby enhancing resilience, scalability, and trustworthiness in digital lending ecosystems.

ACKNOWLEDGMENTS

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

FUNDING

Not applicable.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

PUBLISHER'S NOTE

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

AUTHOR CONTRIBUTIONS

Not applicable.

ABOUT THE AUTHORS

YANG, Ximeng

Board of Directors, Excellent Era Lending Service Corp., Makati City, Philippines.

ZHANG, Yiming

Department of Financial Technology, Peking University, Beijing, China.

REFERENCES

[1] Stapleton, A. H. (2022). The financial fraud epidemic and how it has changed business fraud (Master's thesis, Utica University).

[2] Samuel, A. (2023). Enhancing financial fraud detection with AI and cloud-based big data analytics: Security implications. Available at SSRN 5273292.

[3] Hu, R., Jian, X., Wang, J., & Zhao, H. (2025, July). Construction of a prediction model for rehabilitation training effect based on machine learning. In Proceedings of the 2025 2nd International Conference on Image Processing, Intelligent Control and Computer Engineering (pp. 41-45).

[4] Tan, C., Gao, F., Song, C., Xu, M., Li, Y., & Ma, H. (2024). Proposed Damage Detection and Isolation from Limited Experimental Data Based on a Deep Transfer Learning and an Ensemble Learning Classifier.

[5] Yang, J., Hu, R., Wu, C., Jiang, G., Alkanhel, R. I., & Elmannai, H. (2024). Sensor-Infused Emperor Penguin Optimized Deep Maxout Network for Paralyzed Person Monitoring. IEEE Sensors Journal, 25(13), 25638-25646.

[6] Xu, Ivonne. (2025). Computer Vision-Enabled Inventory Management System: A Cloud-Native Solution for Retail Cost Reduction. Retrieved from SSRN.

[7] Gonzalez, Jean, Vinh Tran, John Meredith, Ivonne Xu, Ritviksiddha Penchala, Laura Vilar-Ribó, Natasia Courchesne-Krak, et al. (2025). How it begins: Initial

response to opioids strongly predicts self-reported opioid use disorder. medRxiv, 2025-03.

[8] Yang, J., Wu, Y., Yuan, Y., Xue, H., Bourouis, S., Abdel-Salam, M., ... & Por, L. Y. (2025). Llm-ae-mp: Web attack detection using a large language model with autoencoder and multilayer perceptron. Expert Systems with Applications, 274, 126982.

[9] Zhao, H., Chen, Y., Dang, B., et al. (2024). Research on steel production scheduling optimization based on deep learning. Proceedings of the 2024 4th International Symposium on Artificial Intelligence and Intelligent Manufacturing, 813-816.

[10] Yuan, Y., Xue, H. (2025). Cross-media data fusion and intelligent analytics framework for comprehensive information extraction and value mining.

[11] Lu, J., Zhao, H., Zhai, H., et al. (2025). DeepSPG: Exploring deep semantic prior guidance for low-light image enhancement with multimodal learning. Proceedings of the 2025 International Conference on Multimedia Retrieval, 935-943.

[12] Yang, W., Lin, Y., Xue, H., et al. (2025). Research on stock market sentiment analysis and prediction method based on convolutional neural network.

[13] Han, X., Dou, X. (2025). User recommendation method integrating hierarchical graph attention network with multimodal knowledge graph. Frontiers in Neurorobotics, 19, 1587973.

[14] Sha, F., Ding, C., Zheng, X., et al. (2025). Weathering the policy storm: How trade uncertainty shapes firm financial performance through innovation and operations. International Review of Economics & Finance.

[15] Yuan, Y., Xue, H. (2025). Multimodal information integration and retrieval framework based on graph neural networks. Proceedings of the 2025 4th International Conference on Big Data, Information and Computer Network, 135-139.