China and WTO Review (CWR) Journal
2026, Vol(1)
ISSN: 2384-4388
https://cwto.net

Research Article

Transaction-Gateway Security Monitoring Through Real-Time Communication Behavior Analysis

Daniel Thompson [1*], Olivia Mitchell [1*]

[1] S.Department of Computer Science, University of Toronto, Toronto, Canada
* **Corresponding Author**: o.mitchell@mcgill.ca

ARTICLE INFO

ABSTRACT

Online payment gateway systems process high-frequency communication among user devices, merchant servers, banking interfaces, fraud-control services, and third-party APIs. Abnormal network traffic in these systems may reflect credential stuffing, transaction replay, API abuse, bot-driven request bursts, or coordinated denial-of-service activities. This study proposes a SHAP-based ensemble anomaly detection framework for online payment gateway network traffic. The proposed model combines Balanced Random Forest, LightGBM, and AdaBoost using a cost-aware voting strategy to reduce missed detection of high-risk abnormal traffic. SHAP is used to explain how transaction-related network features and API communication patterns influence model decisions. Experiments are conducted on a payment gateway traffic dataset generated from a controlled transaction-processing environment with 1,200 merchant endpoints, 9 banking API interfaces, and 5 fraud-screening services. The dataset contains 7.18 million labeled request-flow records over 42 days, covering normal payment authorization, refund requests, chargeback queries, credential stuffing, replay attempts, abnormal API polling, and bot-based request flooding. After preprocessing, 55 features are retained, including request burst rate, failed authentication ratio, API endpoint switching frequency, transaction session duration, repeated token usage, response-code entropy, and source IP rotation density. The proposed model achieves 98.34% accuracy, 97.27% macro-F1, and 99.02% AUC. For replay and credential-stuffing traffic, recall reaches 95.86% and 96.44%, respectively. Compared with standalone LightGBM, the ensemble method reduces the false negative rate from 2.58% to 1.49% and improves detection of low-volume API abuse by 3.63%. SHAP analysis identifies repeated token usage, high failed authentication ratio, abnormal endpoint switching, and source IP rotation density as the most important features. The results indicate that explainable ensemble learning can strengthen real-time security monitoring for online payment gateway infrastructures.

Keywords: Payment Gateway Security, Network Traffic Anomaly Detection, API Abuse Detection, Explainable AI, SHAP, Ensemble Learning, Credential Stuffing, Transaction Network Security

INTRODUCTION

Online payment gateway systems have become a critical component of digital commerce because they connect user devices, merchant servers, banking APIs, fraud-control services, and third-party platforms within a short and highly time-sensitive transaction process (Wang et al., 2025; Boateng, 2025). This architecture improves payment efficiency and supports large-scale real-time financial services, but it also expands the attack surface of payment infrastructures (Xu et al., 2026; Mitchell et al., 2025. Abnormal request-flow behavior may indicate credential stuffing, transaction replay, API abuse, bot-driven request bursts, token misuse, unauthorized refund attempts, or denial-of-service activity. These threats are difficult to identify because malicious requests are often embedded in normal payment authorization, refund, chargeback, settlement, and fraud-screening processes. Recent work on policy-driven vulnerability risk quantification has also shown that large-scale digital infrastructure requires risk assessment methods that can connect security policies, system exposure, and operational data rather than relying only on isolated vulnerability indicators (Shao, 2026). This view is highly relevant to payment gateway security, where abnormal behavior may arise from both endpoint-level abuse and cross-service interaction risks (Segireddy, 2025; Huang, 2026).

Existing fraud detection studies have mainly focused on transaction records, cardholder behavior, merchant profiles, and financial loss patterns (Fariha et al., 2025; Ma, 2026). These approaches are useful for identifying fraudulent payments after transaction information has been generated, but they are less effective for detecting gateway-level attacks before a transaction is completed (Pratama & Wahid, 2025; Zhu et al., 2025). Payment gateway security requires earlier analysis of request-flow behavior, API communication patterns, and service response characteristics (Penaganti, 2025; Zi et al., 2025). Features such as failed authentication ratio, repeated token usage, API endpoint switching, response-code entropy, request burst rate, source IP rotation density, and abnormal retry behavior can provide early signals of attack activity (Damarched, 2026; Yin et al., 2026). However, these indicators are often weak, sparse, and highly imbalanced in real traffic. Low-volume replay attempts, slow API probing, and distributed bot requests may not produce obvious transaction-level fraud signals, but they can still compromise gateway availability, user account security, and downstream banking interfaces (Hossain, 2025; Zhang et al., 2026). Machine learning has been widely used in fraud detection and network intrusion detection because it can learn nonlinear relations among traffic features, user behavior, and system responses (Ali et al., 2025; Qi & Qiao, 2026). Nevertheless, payment gateway traffic presents several challenges for conventional learning models. Abnormal requests usually account for only a small portion of the total traffic, and their patterns may vary across merchants, banking interfaces, and fraud-screening services (Owusu-Mensah et al., 2026; Su & Wu, 2026). A single classifier may therefore achieve high overall accuracy while still missing high-risk abnormal traffic. In addition, many black-box models provide limited explanation for security analysts, making it difficult to understand why a request flow is classified as abnormal. For real-time payment security monitoring, detection performance alone is not sufficient; the model should also provide interpretable evidence that can support alert review, rule adjustment, and operational response (Shakil et al., 2025; Gao et al., 2026).

To address these problems, this study proposes a SHAP-based ensemble anomaly detection framework for online payment gateway network traffic. The framework combines Balanced Random Forest, LightGBM, and AdaBoost through a cost-aware voting strategy to reduce missed detection of high-risk abnormal traffic under class imbalance. SHAP is introduced to explain how transaction-related network features, authentication behavior, token usage, API endpoint transitions, and response-code patterns influence model decisions. Experiments are conducted in a controlled transaction-processing environment involving 1,200 merchant endpoints, 9 banking API interfaces, and 5 fraud-screening services. The dataset contains 7.18 million labeled request-flow records collected over 42 days. The purpose of this study is to improve early anomaly detection at the payment gateway layer before suspicious transactions are finalized. Its significance lies in providing an interpretable and operationally useful detection framework that can support real-time security monitoring, reduce missed high-risk attacks, and strengthen the resilience of online payment infrastructures.

MATERIALS AND METHODS

Sample and Study Environment

This study used a controlled online payment gateway environment to represent high-frequency transaction traffic. The environment included 1,200 merchant endpoints, 9 banking API interfaces, and 5 fraud-screening services. The merchant endpoints represented retail platforms, subscription services, booking systems, and mobile payment clients. The banking interfaces handled authorization, settlement confirmation, refund verification, and chargeback queries. The fraud-screening services processed risk scoring, token validation, device checks, and suspicious request filtering. Traffic was collected for 42 consecutive days under normal and abnormal transaction conditions. In total, 7.18 million labeled request-flow records were obtained. Each record described request time, authentication status, API route, token behavior, response code, source address pattern, and transaction session information. Sensitive information, including user identifiers, card-related fields, merchant names, and token values, was anonymized before analysis.

Experimental Design and Control Settings

The experiment was designed to detect abnormal network traffic in online payment gateway systems under imbalanced and high-risk transaction conditions. The control group included normal payment authorization, refund requests, chargeback queries, fraud-screening requests, banking API responses, and routine merchant-server communication. The experimental group included credential stuffing, transaction replay, abnormal API polling, bot-based request flooding, and low-volume API abuse. These scenarios were selected because they reflect common threats to payment gateway systems. Credential stuffing represents repeated login or payment attempts with stolen credentials. Replay traffic reflects repeated use of old transaction tokens or request signatures. Abnormal API polling indicates excessive or irregular access to payment endpoints. Bot-based request flooding represents automated high-frequency access. Low-volume API abuse reflects slow and hidden misuse of gateway

interfaces. Balanced Random Forest, LightGBM, and AdaBoost were used as the main classifiers. Single models were used as baselines to test whether the cost-aware ensemble reduced missed detection of high-risk abnormal traffic.

Measurement Methods and Quality Control

Request-flow records were measured from gateway logs, merchant access logs, banking API logs, and fraud-control service logs. The basic fields included source endpoint, destination API, request time, response time, request method, response code, session duration, authentication result, token identifier, and source IP segment. Transaction-related network features were then calculated, including request burst rate, failed authentication ratio, API endpoint switching frequency, transaction session duration, repeated token usage, response-code entropy, and source IP rotation density. API behavior features were derived from repeated endpoint access, abnormal polling interval, route-switching pattern, and response failure sequence. Quality control was performed before feature extraction. Duplicate records, incomplete request logs, invalid timestamps, and records with impossible response times were removed. Logs from merchant endpoints, banking APIs, and fraud-screening services were aligned to the same time reference. Attack labels were checked by matching attack execution windows with the related request-flow records. After cleaning and feature selection, 55 features were retained for model training and evaluation.

Data Processing and Model Formulation

All records were cleaned, anonymized, and converted into request-flow samples. Continuous features were normalized using statistics from the training set. Categorical variables, including API route type, response-code group, request method, and merchant endpoint category, were encoded before model fitting. The dataset was divided into training, validation, and test sets by collection time to reduce data leakage. Class imbalance was handled by class-weight adjustment and risk-based sampling. The ensemble model combined the calibrated prediction scores of Balanced Random Forest, LightGBM, and AdaBoost through a cost-aware voting strategy. The final abnormal probability was calculated as (1):

$$P(y=1|x)=\sum_{i=1}^{3} c_i\, w_i P_i(y=1|x) \tag{1}$$

where $P_i(y=1|x)$ is the abnormal probability predicted by the *i-th* classifier, *Wi* is the model weight based on validation performance, and *Ci* is the cost factor used to increase sensitivity to high-risk abnormal traffic. Model performance was measured using accuracy, precision, recall, macro-F1, AUC, and false negative rate. The false negative rate was calculated as (2):

$$FNR=\frac{FN}{FN+TP} \tag{2}$$

where "FN" denotes abnormal request-flow records incorrectly classified as normal, and "TP" denotes abnormal records correctly classified as abnormal.

Model Explanation and Evaluation Procedure

SHAP analysis was used to explain how transaction-related network features and API communication patterns affected model decisions. Global SHAP values were calculated to identify the main indicators across all traffic categories. Local SHAP values were used to explain single abnormal alerts. Positive SHAP values increased the abnormal probability, while negative values supported a normal decision. The explanation results were checked with request context, including repeated token usage, failed authentication ratio, API endpoint switching, response-code entropy, request burst rate, and source IP rotation density. This step helped determine whether the model used meaningful payment-gateway risk signals rather than random request noise. The proposed ensemble model was compared with standalone Balanced Random Forest, LightGBM, and AdaBoost. The evaluation focused on macro-F1, AUC, false negative rate, replay detection, credential-stuffing detection, and low-volume API abuse recall. This design tested whether the method could improve payment gateway anomaly detection while keeping the decision process clear for fraud analysts and security engineers.

RESULTS AND DISCUSSION

Overall Detection Performance

The proposed SHAP-based ensemble model showed strong performance on the online payment gateway traffic dataset. It achieved 98.34% accuracy, 97.27% macro-F1, and 99.02% AUC. These results indicate that the model could separate normal payment communication from abnormal request-flow traffic with high stability. Compared with standalone LightGBM, the ensemble model reduced the false negative rate from 2.58% to 1.49%. This reduction is important because missed attacks may allow replay attempts, credential stuffing, or API abuse to continue within the payment process (Kolhar & Gundoor, 2026; Li & Liu, 2026). The improvement was related to the joint use of Balanced Random Forest, LightGBM, and AdaBoost. Balanced Random Forest reduced the bias toward normal traffic. LightGBM handled high-dimensional gateway features efficiently. AdaBoost improved the learning of difficult abnormal samples. Similar financial fraud studies have shown that ensemble learning and explainable models can improve detection under imbalanced transaction conditions, as shown in **Figure 1.**

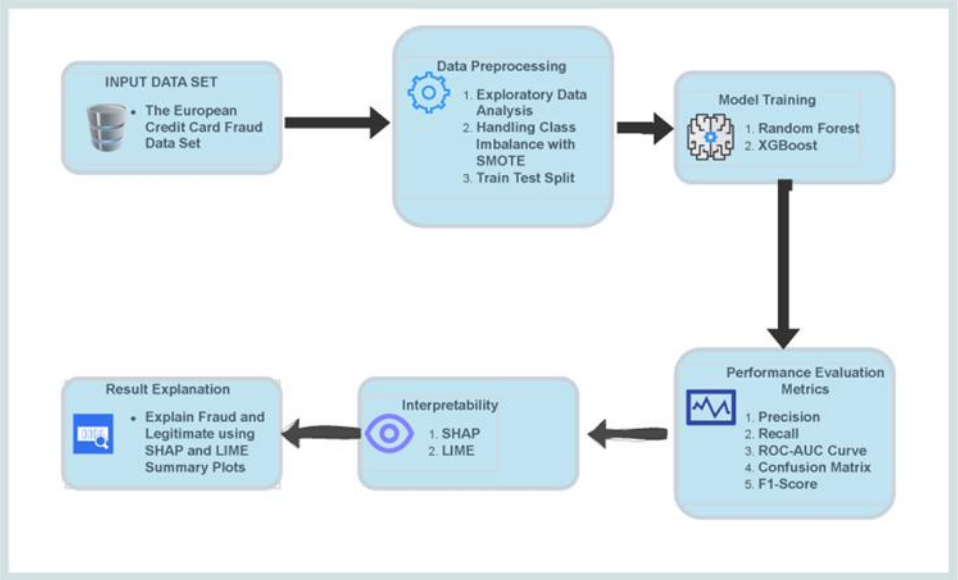

Figure 1. Ensemble Model Workflow For Payment Fraud Detection

Detection of Replay and Credential-Stuffing Traffic

The proposed model achieved 95.86% recall for replay traffic and 96.44% recall for credential-stuffing traffic. These results show that request-flow features can capture abnormal behavior before a fraudulent payment is completed. Replay traffic was mainly related to repeated token usage, abnormal session duration, and repeated request signatures. Credential-stuffing traffic was closely related to high failed authentication ratio, source IP rotation density, and short request bursts (Yang, 2026; Tallam, 2025). These findings suggest that payment gateway anomaly detection should not rely only on transaction amount or user profile features. API route changes, authentication failures, token reuse, and response-code patterns are also needed to identify network-layer payment threats.

Low-Volume API Abuse and False Negative Control

The ensemble model improved the detection of low-volume API abuse by 3.63% compared with standalone LightGBM. This result is meaningful because low-volume API abuse may not cause a clear traffic spike (Deng & Yang, 2026; Deressu et al., 2026). It may appear as slow endpoint polling, irregular route switching, or repeated access to sensitive API interfaces. The cost-aware voting strategy reduced missed detection by assigning higher decision weight to high-risk abnormal samples. This design is suitable for payment gateway systems, where missing a replay attempt or credential attack may cause greater loss than reviewing a small number of alerts. Recent studies on financial fraud detection also show that class imbalance and rare-event detection remain major challenges in real-world payment security, as shown in **Figure 2**.

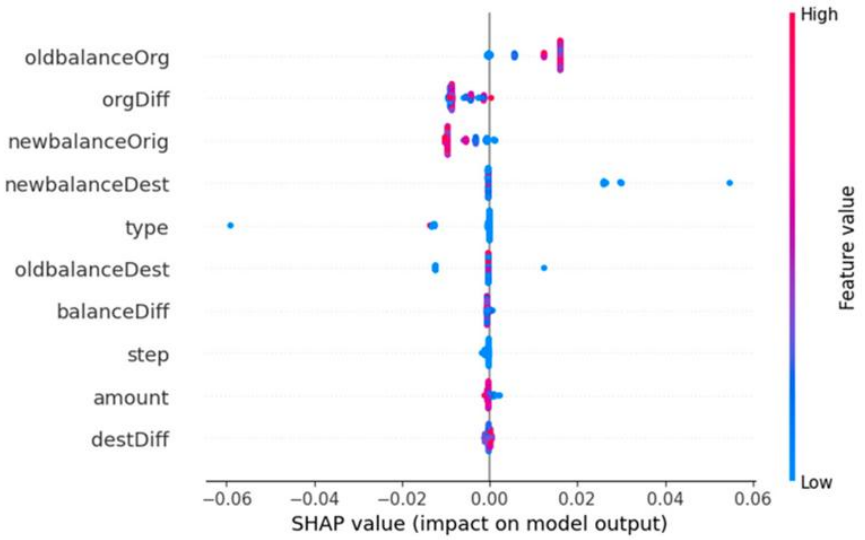

Figure 2. Ensemble Model Workflow For Payment Fraud Detection

SHAP Interpretation and Practical Security Meaning

SHAP analysis showed that repeated token usage, high failed authentication ratio, abnormal API endpoint switching, and source IP rotation density were the most important indicators of abnormal payment gateway traffic. These features have clear security meaning. Repeated token usage may indicate replay attempts. A high failed authentication ratio may suggest credential stuffing. Abnormal endpoint switching may reflect API abuse or probing (Zhang, 2026). Dense source IP rotation may indicate bot-based request activity. Local SHAP results also made single alerts easier to review because they showed which features increased the abnormal score. Compared with black-box models, the proposed method provides both a detection result and a feature-level explanation. However, this study still has limitations. The dataset was generated in a controlled transaction-processing environment. Real payment gateways may include more merchant types, API versions, traffic peaks, fraud-control rules, and regional routing patterns. Future work should test the model on longer real-world request-flow records and include more payment scenarios.

CONCLUSION

This study presented a SHAP-based ensemble method for detecting abnormal network traffic in online payment gateway systems. The model combined Balanced Random Forest, LightGBM, and AdaBoost through cost-aware voting to improve the detection of high-risk request-flow anomalies. Tests on a controlled payment gateway dataset showed strong performance, with 98.34% accuracy, 97.27% macro-F1, and 99.02% AUC. The model also reached 95.86% recall for replay traffic and 96.44% recall for credential-stuffing traffic. Compared with standalone LightGBM, the false negative rate decreased from 2.58% to 1.49%, and the detection of low-volume API abuse improved by 3.63%. These results indicate that the ensemble structure reduced missed detections in imbalanced payment traffic. SHAP analysis showed that repeated token usage, high failed authentication ratio, abnormal API endpoint switching, and source IP rotation density were key signs of abnormal gateway behavior. The main value of this study lies in combining accurate anomaly detection with clear feature-level explanation. This design can support real-time payment security monitoring, fraud-control systems, and API risk management. However, the dataset was generated in a controlled transaction-processing environment. Real payment gateways may include more merchant types, API versions, regional routing patterns, traffic peaks, and fraud-control rules. Future work should test the method on longer real-world request-flow records and cover more payment scenarios.

REFERENCES

Ali, M. L., Thakur, K., Schmeelk, S., Debello, J., & Dragos, D. (2025). Deep learning vs. machine learning for intrusion detection in computer networks: A comparative study. *Applied Sciences, 15*(4), 1903.

Boateng, B. (2025). *Assessment of DPI and Stakeholder Willingness to Pay* (Doctoral dissertation, Worcester Polytechnic Institute).

Damarched, M. K. (2026). A HIPAA-Aware Agentic AI Co-Pilot Framework: Orchestrating Secure Multi-Step EHR Workflows for Clinical Burden Reduction in US Hospital Systems. *Journal of Drug Delivery & Therapeutics, 16*(3), 71-93.

Deng, X., & Yang, J. (2026). Design of a Quantitative Futures Trading Model Incorporating Multimodal Feature Fusion and Tail Risk Control. Available at SSRN 6702398.

Deressu, T. F., Beyene, A. M., & Gemechu, A. Y. (2026). A Survey of Application Layer Distributed Denial of Service Detection: Approaches, Models, and Datasets. *Artificial Intelligence Review, 59*(6), 139.

Fariha, N., Khan, M. N. M., Hossain, M. I., Reza, S. A., Bortty, J. C., Sultana, K. S., ... & Begum, M. (2025). Advanced fraud detection using machine learning models: Enhancing financial transaction security. *arXiv preprint arXiv:2506*.10842.

Gao, G., Gao, R., Gao, R., Zhou, H., & Lu, C. (2026, March). Performance Study of Enterprise SMS Communication Scheduling Mechanisms in Triple-Play Convergence Environments. In *2026 IEEE 8th International Conference on Communications, Information System and Computer Engineering (CISCE)* (pp. 82-86). IEEE.

Hossain, M. A. (2025). Investigating the cybersecurity implications of open banking and application programming interfaces (APIs) in the financial sector. Available at SSRN 5207072.

Huang, R. (2026). ConfSched: Confidence-Threshold Routing for Efficient Edge-Cloud Activity Recognition. *World Journal of Engineering and Technology, 14*(2), 471-486.

Kolhar, A., & Gundoor, T. K. (2026). API Security Testing and Exploitation Techniques. In *Advanced Cybersecurity for Threats Exploitation and Digital Risk* (pp. 87-116). IGI Global Scientific Publishing.

Li, Y., & Liu, S. (2026, May). A Study on Dynamic Optimization of Alerting Policies and Multi-Agent Decision-Making Mechanisms in Cloud Environments. In *2026 7th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)* (pp. 703-706). IEEE.

Ma, Y. (2026). Industrial Financial Risk Prediction Model Based on Graph Neural Network and Knowledge Graph Inference. J*ournal of Circuits, Systems and Computers, 35*(16), 2650103.

Mitchell, E., Reed, I., & Ok, E. (2025). Intelligent Defense Mechanisms in Financial Technology: Combating Evolving Threats in FinTech and Cloud Infrastructure.

Owusu-Mensah, K., Ansong, E. D., Adu-Manu, K. S., & Yaokumah, W. (2026). A Systematic Review of Frameworks for the Detection and Prevention of Card-Not-Present (CNP) Fraud.

Penaganti, R. (2025). Building a Universal Payment Platform: Breaking Free from Payment Gateway Lock-in. *Journal of International Crisis & Risk Communication Research (JICRCR)*, 8.

Pratama, S. F., & Wahid, A. M. A. (2025). Fraudulent transaction detection in online systems using random forest and gradient boosting. *Journal of Cyber Law, 1*(1), 88-115.

Qi, C., & Qiao, X. (2026). Using AI to Monitor AI: Automated Operations Through Log-Driven Intelligence. Available at SSRN 6795840.

Segireddy, A. R. (2025). Generative Ai For Secure Release Engineering In Global Payment Network. *Lex Localis: Journal of Local Self-Government, 23*(S6), 8166.

Shakil, S. M., Hossain, A., & Rahman, M. M. (2025). An empirical evaluation of anomaly detection techniques in smart grid systems using real-time operational data. *American Journal of Advanced Technology and Engineering Solutions, 1*(02), 95-134.

Shao, W. (2026). Policy-Driven Vulnerability Risk Quantification framework for Large-Scale Cloud Infrastructure Data Security. *arXiv preprint arXiv:2604*.06252.

Su, D., & Wu, Y. (2026). Multilevel Growth and Social Network Modeling for Functional Communication Enhancement in Students with Intellectual Disabilities.

Tallam, K. (2025). CyberSentinel: An emergent threat detection system for AI security. *arXiv preprint arXiv:2502*.14966.

Wang, Y., Wen, Y., Wu, X., Wang, L., & Cai, H. (2025). Assessing the Role of Adaptive Digital Platforms in Personalized Nutrition and Chronic Disease Management.

Xu, S., Ma, Q., Liu, H., & Yue, L. (2026). Continuous Reorganization and Performance Preservation of Agent Memory Structure Under Distributed Change Environments.

Yang, J. (2026). Stage-Coupled Computational Framework for Stratified Accessibility and Equity Analysis in Community-Based Elderly Care Services.

Yin, J., Huang, Y., & Rao, H. (2026). A Study on User Behavior Signal-Driven Predictive Models for Digital Platform Consumption Trends. Available at SSRN 6607298.

Zhang, Z. (2026). A Study on Multimodal Product Understanding and Assembly Guidance Optimization in e-commerce for Complex Consumer Goods. Available at SSRN 6734559.

Zhang, Z., Gao, Y., & Tong, Y. (2026). Semantic-Temporal Graph Learning for Demand Forecasting and Resource Allocation in Public Information Systems. Available at SSRN 6792279.

Zhu, W., Yao, Y., & Yang, J. (2025). Optimizing Financial Risk Control for Multinational Projects: A Joint Framework Based on CVaR-Robust Optimization and Panel Quantile Regression.

Zi, Y., Gong, M., Xue, Z., Zou, Y., Qi, N., & Deng, Y. (2025, August). Graph neural network and transformer integration for unsupervised system anomaly discovery. In *2025 5th International Conference on Computer Science and Blockchain (CCSB)* (pp. 158-162). IEEE.

ETHICAL DECLARATION

Conflict of interest: No declaration required. **Financing:** No reporting required. **Peer review:** Double anonymous peer review.