Nithya Shree N, 2026, 14:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Smart Commerce Fraud Detection Using Artificial Intelligence and Machine Learning

Nithya Shree N, Jnaneshwari M
Department of CSE (AI&ML), Don Bosco Institute of Technology, Bengaluru, India

Abstract- Smart commerce platforms increasingly depend on digital identities, online payments, user reviews, and high-volume order processing, creating multiple opportunities for fraudulent activity. Existing rule-based mechanisms are transparent and easy to implement, but they are static, require frequent manual updates, and can generate false positives when fraud behaviour changes. This paper proposes a smart-commerce fraud detection framework that combines transactional, behavioural, contextual, and review-related signals within a unified artificial intelligence and machine-learning pipeline. The proposed framework uses supervised learning models such as Random Forest and XGBoost for labelled transaction and account classification, Isolation Forest for anomaly detection, and an optional natural-language processing layer for suspicious review identification. Behavioural features include login frequency, transaction velocity, device usage, location consistency, order cancellation rate, address changes, payment method, and review activity. The system incorporates class-imbalance handling, feature engineering, real-time risk scoring, and an explainability layer based on feature importance and SHAP/LIME-style local explanations. The design also supports continuous feedback and future graph-based analysis for coordinated fraud networks. Its methodology covers data acquisition, inspection, preprocessing, feature engineering, model training, evaluation, model selection, prediction, explanation, and deployment. A risk score combines supervised transaction-fraud probability, anomaly score, behavioural risk, contextual inconsistency, and review/account risk. Events are assigned low-, medium-, or high-risk states to support normal processing, additional verification, or blocking/manual review. Evaluation is planned using precision, recall, F1-score, ROC-AUC, PR-AUC, confusion matrix, false-positive rate, and inference time. The paper focuses on a reproducible research framework and does not fabricate experimental performance values; numerical results are to be reported after implementation on the final project dataset. The proposed architecture is intended to improve early fraud detection, reduce false positives, strengthen transaction security, and increase trust in smart-commerce platforms.

Keywords- Smart Commerce Security, Fraud Detection, Artificial Intelligence, Machine Learning, XGBoost, Random Forest, Isolation Forest, Behavioral Analytics, Fake Review Detection, Explainable AI

I. INTRODUCTION

Smart commerce has transformed conventional retail by combining online marketplaces, digital payments, mobile applications, personalized services, and automated order processing. This growth improves convenience but also enlarges the attack surface for fraudulent accounts, unauthorized payments, fake orders, identity misuse, review manipulation, and abnormal purchasing behaviour. The project synopsis supplied for this work identifies fraudulent user accounts, fake reviews, fake payments, suspicious purchase activities, and behavioural anomalies as the principal security concerns. It also emphasizes the need for an intelligent automated system capable of detecting suspicious behaviour in real time.

Traditional fraud detection commonly depends on predefined rules such as transaction limits, blacklists, location checks, or manual review. Such rules remain useful as a first line of defence because their decisions are easy to understand. However, the reviewed literature shows that static rules can become ineffective when fraudsters change their strategies. Machine learning addresses this limitation by learning patterns from historical data and assigning risk to new observations. Recent e-commerce studies have explored Random Forest, XGBoost, stacking classifiers, LSTM models, anomaly detection, and real-time scoring.

© 2026 Nithya Shree N. his is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

The problem is not limited to deciding whether one transaction is fraudulent or genuine. A practical smart-commerce system must combine several sources of evidence, deal with severe class imbalance, operate with low latency, limit false positives, explain why an event was flagged, and adapt to changing fraud patterns. The proposed work addresses this requirement through a layered architecture covering data acquisition, preprocessing, feature engineering, model training, anomaly detection, risk scoring, explanation, and feedback.

The main contributions of this paper are: (1) a unified framework for payment, account, order, behaviour, and review-related fraud signals; (2) a hybrid supervised and unsupervised detection strategy; (3) explicit treatment of class imbalance and behavioural features; (4) a real-time decision and alerting workflow; and (5) an explainability and reproducibility plan that avoids unsupported performance claims.

II. LITERATURE REVIEW

Reddy et al. describe e-commerce fraud detection as a data-intensive problem in which machine learning and big-data analytics can identify patterns and anomalies that are difficult to capture through manual methods. Their review discusses supervised and unsupervised learning, user spending behaviour, sequential transaction information, anomaly features, and the need to address limitations of static rules.

Valanarasu proposes a layered AI-driven cybersecurity architecture integrating data ingestion, preprocessing, model training, anomaly detection, real-time monitoring, and decision making. The study considers purchase history, geolocation, device fingerprinting, payment methods, and login patterns, and discusses Logistic Regression, Random Forest, XGBoost, and neural networks. It also identifies privacy, explainability, adversarial behaviour, and evolving fraud tactics as important challenges.

Raju et al. present an e-commerce fraud detection implementation using XGBoost and a Stacking Classifier with a synthetic dataset containing transaction and customer features. The work demonstrates a practical transaction flow in which data is sent to a fraud detection module and the result determines whether processing continues or a warning is issued. The paper also highlights feature engineering, real-time detection, database logging, and continuous model updates.

Manasa and Ashish examine AI-powered fraud prevention for e-commerce and emphasize real-time analysis, behavioural analysis, identity verification, device fingerprinting, and predictive analytics. Their literature survey also identifies scalability, interpretability, robustness, privacy, and responsible AI as important requirements.

Jhansi presents a behaviour-based real-time framework combining LSTM for sequential behaviour and XGBoost for tabular transaction data. The study identifies a research gap around real-time streaming, session-level behavioural features, label scarcity, explainability, and hybrid approaches. These observations directly motivate the present framework.

Aruna et al. propose a smart fraud detection system for online payments using supervised learning, SMOTE-based class balancing, a Flask application, and automated email alerts. Revanth et al. similarly discuss behavioural and transactional indicators such as transaction time, location inconsistency, order counts, cancellation rates, payment preferences, and address changes, with Random Forest and XGBoost as candidate models.

Across these studies, a consistent research gap emerges: transaction-only models may miss coordinated or behavioural fraud, while complex deep-learning systems can be computationally expensive and difficult to explain. The proposed smart-commerce framework therefore combines tabular classification, anomaly detection, behavioural features, review analysis, and explainability while keeping the implementation modular.

III. RESEARCH GAP AND OBJECTIVES

Existing research has established the usefulness of machine learning for transaction fraud, but several issues remain. First, many systems concentrate on payment transactions and do not treat fraudulent accounts, fake reviews, and suspicious order behaviour as connected signals. Second, class imbalance can make accuracy misleading because legitimate events dominate the dataset. Third, real-time scoring requires efficient feature computation and a clear decision policy. Fourth, high-performing models may be difficult for users or administrators to interpret. Finally, fraud patterns evolve, so a static model can degrade after deployment.

The proposed research objectives are: (1) to develop a smart security framework using AI and ML; (2) to detect fraudulent user accounts from registration and behavioural patterns; (3) to identify suspicious reviews and ratings; (4) to detect fraudulent payments and fake orders; (5) to monitor customer activity using behavioural analytics; (6) to provide real-time risk scoring and alerts; and (7) to improve trust, security, and operational reliability.

IV. PROPOSED METHODOLOGY

The methodology follows a step-by-step workflow covering data acquisition, inspection, preprocessing, feature engineering, model training, evaluation, model selection, prediction, explanation, and deployment. The system is deliberately modular so that a change in one component does not silently change another component.

1. Data Acquisition

The framework can use public fraud datasets and, where necessary, a project-specific synthetic dataset. Relevant sources include transaction amount and time, customer/account identifiers, payment method, merchant or product information, IP/geolocation, device information, login history, order history, cancellation behaviour, delivery-address changes, review text/rating, and previous fraud outcomes. The exact dataset name, sample count, feature count, and class distribution should be recorded from the final dataset used in implementation rather than assumed in advance.

2. Preprocessing

Missing values and duplicate records are inspected before model training. Numerical variables are scaled when required, while categorical variables are encoded. Timestamp fields are transformed into useful temporal features such as hour, day, session interval, and time since the previous transaction. Training-only fitting is used for learned preprocessing operations to avoid data leakage. Class imbalance is examined using class distributions and appropriate techniques such as class weighting or SMOTE are evaluated only within the training process.

3. Feature Engineering

The proposed feature groups are transaction, account, behavioural, contextual, historical, and review features. Derived features include transaction velocity, average spend, deviation from historical spend, login frequency, time since last login, cancellation-to-order ratio, number of address changes, location mismatch, device reuse across accounts, and review frequency. These features are intended to capture relationships that are not visible from a single transaction value.

4. Model Layer

XGBoost and Random Forest are proposed as the main supervised classifiers because the reviewed literature repeatedly identifies ensemble tree methods as practical candidates for structured fraud data. Logistic Regression can be retained as a baseline. Isolation Forest is included as an unsupervised anomaly detector for unusual events that may not have labels. For review analysis, a lightweight NLP classifier can be evaluated to identify suspicious review patterns. A graph neural network component is considered as future work for relationships among users, devices, addresses, payment instruments, and products.

5. Risk Scoring

Instead of relying only on a binary prediction, the system produces a risk score. A conceptual risk function is defined as $R = w1P_t + w2A + w3B + w4C$

+ w5V, where P_t is the supervised transaction-fraud probability, A is the anomaly score, B is behavioural risk, C is contextual inconsistency, V is review/account risk, and the weights are selected during validation. This formula is a design representation; the final weights must be learned or tuned from the actual project dataset.

6. Decision Policy

The scoring engine can classify events into three operational states: low risk, requiring normal processing; medium risk, requiring additional verification; and high risk, requiring blocking or manual review. Thresholds should be selected using validation data and business costs rather than arbitrary values.

V. SYSTEM ARCHITECTURE

The architecture is organized into seven main layers. The first layer collects transaction, account, behaviour, device, location, and review information. The second layer validates and preprocesses the data. The third layer creates derived behavioural and historical features. The fourth layer executes the supervised classifiers and the anomaly detector. The fifth layer combines the outputs into a risk score. The sixth layer provides an explanation, alert, and decision. The final layer stores feedback and confirmed outcomes for later model updates.

User Activity → Data Collection → Preprocessing →Feature Engineering → ML / Anomaly Models → Risk Scorin → Explainability & Alerts → Decision / Verification → Database & Feedback

Fig. 1. Proposed smart-commerce fraud detection architecture.

The transaction path is: User activity → Data collection → Preprocessing → Feature engineering → ML/anomaly models → Risk scoring → Explainability → Decision/alert → Database and feedback. A suspicious transaction can be routed to secondary verification while a legitimate transaction proceeds through normal processing.

The application can be implemented in Python using Pandas and NumPy for data processing, Scikit-learn for baseline models and evaluation, XGBoost for gradient boosting, TensorFlow where neural models are required, and Isolation Forest for unsupervised anomaly detection. A lightweight web interface can be implemented using Flask or Streamlit, while a relational or document database can store transaction logs, scores, decisions, and feedback.

Feature Design

Table I summarizes the principal feature categories proposed for the smart-commerce system. The source PDF provides the Table I caption but does not include the table entries; therefore, no unlisted feature-to-category mapping has been invented. The feature groups and derived features explicitly described in the methodology are transaction, account, behavioural, contextual, historical, and review features; transaction velocity, average spend, deviation from historical spend, login frequency, time since last login, cancellation-to-order ratio, number of address changes, location mismatch, device reuse across accounts, and review frequency.

Table I. Feature categories used by the proposed framework (caption as supplied; table entries were not present in the source PDF).

VI. EXPERIMENTAL DESIGN AND REPRODUCIBILITY

The final experiment should use a stratified train-test split or a time-aware split when transaction order is important. Cross-validation may be used on the training partition for model selection. Hyperparameter tuning must not use the held-out test set. The random seed, dataset version, preprocessing operations, feature definitions, model hyperparameters, library versions, and hardware configuration should be recorded.

Performance should be reported using precision, recall, F1-score, ROC-AUC, PR-AUC, confusion matrix, and false positive rate. Accuracy may be included but should not be treated as the primary metric when fraud is rare. Precision measures the fraction of flagged events that are actually fraudulent, while recall measures the fraction of

fraudulent events successfully detected. F1-score summarizes precision and recall using their harmonic mean.

Error analysis should separately examine false positives and false negatives. Particular attention should be given to legitimate high-value purchases, new users, unusual but genuine locations, shared devices, and legitimate behaviour that resembles fraud. This analysis is essential because the operational cost of incorrectly blocking a genuine customer can be significant.

For research integrity, numerical performance values from other papers must not be presented as results of this proposed system. Published results are used only for methodological comparison. The actual project results should be inserted after the final dataset has been processed and the models have been evaluated.

VII. EXPLAINABILITY, PRIVACY AND RESPONSIBLE AI

Explainability is incorporated so that administrators can understand why an event received a high risk score. Global feature importance can identify recurring drivers of fraud predictions, while SHAP or LIME-style local explanations can show which input features contributed to an individual decision. Explanations should describe model behaviour and should not be interpreted as causal proof.

Fraud detection may use sensitive behavioural and contextual information. Therefore, the implementation should minimize collection, anonymize identifiers where possible, protect stored logs, restrict access, and retain data only for legitimate research or operational purposes. The reviewed literature also highlights privacy, fairness, transparency, and adversarial attacks as important considerations for AI-based fraud systems.

Bias and drift should be monitored. A model may learn correlations from historical data that are not stable across customers, regions, devices, or time periods. Periodic validation and retraining are therefore required. Confirmed fraud outcomes can be fed back into the training pipeline after appropriate verification.

VIII. RESULTS AND DISCUSSION

At this stage of the project, the project-specific numerical results are intentionally not fabricated. The final implementation should compare Logistic Regression, Random Forest, XGBoost, Isolation Forest, and any selected hybrid configuration using the same dataset and evaluation protocol. A results table should then report precision, recall, F1-score, ROC-AUC/PR-AUC, false-positive rate, and inference time.

The literature provides useful evidence for the selection of candidate methods. The reviewed e-commerce studies report strong performance from ensemble models such as Random Forest, XGBoost, and stacking classifiers, while behaviour-based research motivates the use of temporal and session features. These published numbers cannot be directly transferred to the proposed system because datasets, class ratios, preprocessing, and experimental conditions differ.

The intended prototype should be able to produce a fraud probability/risk score, identifying suspicious accounts, flagging anomalous transactions, supporting fake-review analysis, and generating an alert or secondary-verification action. The final measured results will determine which model configuration provides the best trade-off between detection capability, false positives, interpretability, and response time.

Limitations and Future Work

The quality of the final system will depend strongly on how suitable and representative the selected dataset is. Fraud labels can be scarce, delayed, noisy, or unavailable for new attack patterns. Synthetic data can be useful for development but may not reproduce every real-world behaviour. Real-time deployment also introduces latency, scaling, privacy, and model-drift challenges.

Future work can explore graph-based fraud detection using relationships among accounts,

devices, addresses, payments, and products; sequence models for session behaviour; transformer-based review analysis; adversarial training; streaming data pipelines; federated learning for privacy-preserving collaboration; and continuous learning with verified feedback. These extensions can make the system more robust against coordinated and evolving fraud while preserving interpretability and operational efficiency.

IX. CONCLUSION

This paper presented a smart-commerce fraud detection framework that integrates transaction intelligence, behavioural analytics, anomaly detection, account and order signals, review analysis, real-time risk scoring, and explainability. The design is motivated by the limitations of static rule-based systems and by recent research showing the value of ensemble learning, behavioural features, class-imbalance handling, and real-time monitoring.

The proposed architecture uses XGBoost and Random Forest as primary supervised candidates, Isolation Forest for unlabelled anomalies, and an optional NLP component for review analysis. It also provides a structured workflow for data preprocessing, feature engineering, evaluation, deployment, feedback, and responsible AI. Most importantly, the paper separates literature-reported findings from project-specific results. Once the final dataset and implementation are available, the proposed evaluation protocol can be used to report reproducible numerical results and identify the most effective model for smart-commerce fraud prevention.

Acknowledgements
No acknowledgement text was provided in the source document.

REFERENCES

1. S. R. Byrapu Reddy, P. Kanagala, P. Ravichandran, R. Pulimamidi, P. V. Sivarambabu, and N. S. A. Polireddi, "Effective fraud detection in e-commerce: Leveraging machine learning and big data analytics," Measurement: Sensors, vol. 33, 101138, 2024, doi: 10.1016/j.measen.2024.101138.
2. D. P. Valanarasu, "Enhancing E-Commerce Fraud Detection Using AI-Driven Cybersecurity Systems," International Research Journal on Advanced Engineering and Management, vol. 03, no. 06, pp. 2341–2344, 2025, doi: 10.47392/IRJAEM.2025.0369.
3. G. Srinivasa Raju, Vishnu Sai Varma, et al., "E-Commerce Fraud Detection Based on Machine Learning Techniques," International Journal of Innovative Research in Technology, vol. 11, no. 11, pp. 5252–5261, 2025.
4. M. A. S. and A. L., "Securing E-Commerce: AI-Powered Fraud Detection and Prevention," International Journal of Novel Research and Development, vol. 9, no. 3, 2024.
5. R. Jhansi, "A Behavior-Based Framework for Real-Time Fraud Detection in E-Commerce," International Journal of Research Publication and Reviews, vol. 6, no. 11, pp. 2140–2143, 2025.
6. S. Aruna, K. Srinivasa Naik, S. Nair, M. Vasavi, G. S. Poornima, V. Kusuma, H. Sowmya, and S. S. Sri, "Smart Fraud Detection System for Online Payments Using AI and Real-Time Notifications," International Journal for Research Trends and Innovation, vol. 10, no. 4, 2025.
7. B. Revanth, T. Y. N. S. Kumar, S. S. P. Reddy, P. Anil, and P. L. S. Chandu, "Fraud Detection for E-commerce," Journal of Emerging Technologies and Innovative Research, vol. 12, no. 3, 2025.
8. Shreya Nehe, "AI-Based Fraud Detection System," International Journal on Science and Technology, vol. 16, no. 2, Apr.–Jun. 2025.
9. J. Prashanthi, A. Pallavi, N. Ramya, B. Shekar Reddy, and K. Hemanth, "Enhancing Fraud Detection in E-Commerce Transactions in Multi Perspective User Behaviour Analysis and Process Mining," cited in the reviewed 2025 literature.
10. Y. Ren, H. Zhu, J. Zhang, P. Dai, and L. Bo, "EnsemFDet: An Ensemble Approach to Fraud Detection Based on Bipartite Graph," cited in the reviewed literature.