International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

E-Commerce Fraud Detection Using Machine Learning

Jyothi Krishna[1], Sindhu Daniel[2]

[1]Department of Computer Applications, Musaliar College of Engineering & Technology, Pathanamthitta, Kerala, India
Email: *jyothikrishna477[at]gmail.com*

[2]Professor, Department of Computer Applications, Musaliar College of Engineering & Technology, Pathanamthitta, Kerala, India

Abstract: ***The rapid expansion of e-commerce has simultaneously increased the sophistication and frequency of fraudulent activities. This paper presents a machine learning-based approach for detecting e-commerce fraud using ensemble models—Stacking Classifier and XGBoost. A synthetic dataset of 23,634 transactions was generated to simulate realistic user and fraud behavior. The proposed system leverages Python and the Flask framework, offering real-time detection capabilities. Evaluation metrics such as precision, recall, and F1-score were used due to the highly imbalanced nature of fraud data. The results indicate significant improvement in fraud detection performance compared to traditional models, particularly in reducing false positives and negatives. The system shows strong adaptability to evolving fraud patterns and can be effectively integrated into e-commerce platforms for live transaction monitoring.***

Keywords: E-commerce, Fraud Detection, Machine Learning, XGBoost, Stacking Classifier, Ensemble Learning, Synthetic Dataset

1. Introduction

The exponential growth of e-commerce platforms over the last decade has transformed the global economy by providing users with the convenience of online transactions. However, this digital transformation has also made online marketplaces more susceptible to fraud, resulting in significant financial losses for both businesses and consumers. As online transaction volumes continue to surge, fraudsters have developed increasingly sophisticated techniques to exploit vulnerabilities within these systems. Traditional rule-based fraud detection mechanisms, while once effective, are now struggling to keep pace with these dynamic threats, leading to a growing demand for more intelligent and adaptable solutions.

Machine learning (ML) has emerged as a promising approach for detecting fraudulent activities in real time by analyzing large volumes of transactional data. Unlike traditional systems that rely on predefined rules, ML algorithms can learn from patterns within historical data to identify anomalies that may indicate fraudulent behavior. Techniques such as decision trees, support vector machines, and ensemble models have shown considerable success in distinguishing between legitimate and fraudulent transactions. However, challenges such as data imbalance, evolving fraud tactics, and the lack of access to high-quality datasets continue to hinder the full potential of ML in this domain.

o address these challenges, this study proposes an ML-based fraud detection system that leverages ensemble methods—specifically, the Stacking Classifier and XGBoost. A synthetic dataset, created to replicate realistic e-commerce transaction behaviors, is used to train and evaluate the models. The goal is to enhance fraud detection accuracy while minimizing false positives and negatives. The proposed system is implemented using Python and deployed using a Flask-based web interface.

2. Literature Review

Taylor (2024) reinforced this by emphasizing the necessity of low-latency detection systems. Her work identified that fraud detection models must operate within milliseconds in high-frequency transaction environments. To meet this demand, she advocated for models like XGBoost, known for its computational efficiency and parallelized tree-building capability. Taylor's conclusions align with the implementation strategies in this study, where XGBoost and a Stacking Classifier are used to process a large volume of transactions effectively.

Sarah Lee (2024) explored fraud detection from a media and technology adoption standpoint. She noted that many e-commerce platforms fail to adopt robust ML-based detection due to either a lack of technical expertise or fear of degrading user experience. Her findings suggest that modern fraud detection systems must be lightweight, fast, and seamlessly integrable with front-end architectures—an insight reflected in our system design, which utilizes Python and Flask for real-time fraud flagging without disrupting the user flow.

Wilson (2024) introduced a framework combining fraud analytics with graph theory. His study showed how fraudulent users are often interconnected through patterns of shared devices, IP addresses, or behavior. Graph-based learning algorithms like Graph Convolutional Networks (GCNs) were shown to be particularly effective in these contexts. This research has laid the groundwork for integrating network-based fraud analysis into traditional machine learning pipelines, thus increasing detection accuracy for collaborative or syndicated fraud.

Browen (2024) explored fraud from a platform-centric lens, particularly focusing on the rise of peer-to-peer e-commerce platforms such as Facebook Marketplace and Etsy. His work identified how the lack of centralized transaction control increases the risk of scams and fraud. Browen proposed that

fraud detection models on such platforms must rely more heavily on user profiling, behavioral analytics, and cross-session transaction tracking—methods that are ideally suited for machine learning pipelines due to their scalability and data-driven nature.

Evensen (2021) contributed from a financial systems perspective, analyzing the economics of digital distribution and fraud prevention. While not focused solely on ML, his work emphasized the balance between maintaining user experience and implementing secure transaction protocols. This intersection of business and security goals is crucial when deploying fraud detection systems that aim to be both effective and customer-friendly.

Patel (2021) presented a hybrid fraud detection system combining decision trees and deep learning models. The paper proposed that the interpretability of tree-based models complements the complexity-handling capabilities of neural networks. Patel's work also introduced the concept of dynamic model updating, where the system adapts over time to evolving fraud patterns—an increasingly necessary feature in fast-paced online transaction systems.

Wang (2021) conducted a large-scale evaluation of supervised learning models for fraud detection. His research assessed performance across metrics like recall, precision, and F1-score, using real-world e-commerce datasets. Wang's findings confirmed that ensemble models such as Random Forest and XGBoost consistently outperformed basic classifiers in terms of robustness and scalability. He also noted that overfitting is a recurring issue in fraud detection, which must be countered through cross-validation and hyperparameter tuning.

Zhang (2021) took a different angle by concentrating on anomaly detection, a type of unsupervised learning, to detect fraud where labels are scarce or inconsistent. His work focused on the use of clustering algorithms like DBSCAN and K-means, along with isolation forests and autoencoders. Zhang's findings supported the idea that unsupervised models are particularly effective in environments with high uncertainty or limited labeled data—scenarios common in real-world e-commerce fraud detection.

Ahmed (2020) explored the integration of feature engineering and ML techniques in fraud detection. His work highlighted that engineered features—such as transaction frequency, customer behavioral patterns, and device information—are critical in enhancing model performance. The paper also pointed out the limitations of traditional models when handling imbalanced datasets, suggesting that techniques like SMOTE (Synthetic Minority Over-sampling Technique) and cost-sensitive learning could offer more balanced prediction outputs.

Saeed (2020) provided one of the most foundational surveys on machine learning (ML) applications in e-commerce fraud detection. The study categorized detection methods into supervised and unsupervised learning, covering algorithms such as decision trees, support vector machines (SVM), and ensemble models like Random Forest and GBM. A key takeaway from Saeed's research was the emphasis on the interpretability of ML models, noting that while deep learning offers high accuracy, simpler models are often preferred in commercial applications where decision transparency is required.

Beale (2016) approached fraud detection from a systems-level perspective, identifying the underlying structural vulnerabilities in digital platforms that allow fraudulent behavior to thrive. The study proposed a framework for building resilience into e-commerce architectures, advocating for machine learning as a key component in adaptive fraud prevention systems. His work emphasizes that beyond algorithmic detection, a fraud-resistant platform design should include modular and scalable ML-driven services that evolve as new threats emerge.

O'Regan (2013) contributed a multidisciplinary perspective by highlighting the human element in fraud ecosystems. His study examined how fraudulent behaviors adapt over time in response to new technological defenses, suggesting that static rule-based systems will always be outpaced. This conclusion supports the growing academic and industrial trend of shifting toward self-learning systems powered by artificial intelligence and machine learning. O'Regan also stressed the importance of cross-functional collaboration between cybersecurity experts and data scientists to improve fraud detection strategies.

David Edgar (2004) offered early insights into the theoretical foundation of fraud systems and digital deception. While dated, his principles still resonate, particularly his emphasis on pattern recognition and anomaly detection, which today align well with unsupervised learning models. Edgar's arguments have inspired modern fraud detection systems that prioritize adaptability, pattern recognition, and probabilistic reasoning—principles now executed through ML algorithms like XGBoost and neural networks.

3. System Requirements

he proposed fraud detection system is developed using Python 3.10 with Flask as the backend framework and HTML, CSS, and JavaScript for the frontend. The application runs on Windows 10 or 11 and is developed using Visual Studio Code. Essential Python libraries include scikit-learn, xgboost, pandas, and matplotlib. The system requires a minimum hardware setup of an Intel Core i3 processor, 4 GB RAM, and 500 GB of storage. For optimal performance and scalability, an 8 GB RAM setup is recommended. The application is compatible with standard input devices and is designed to operate efficiently on screens with a resolution of 1366×768 or higher.

4. Feasibility Study

The proposed system is technically feasible, using widely available tools like Python and open-source libraries. Economically, development costs are low due to free software and minimal hardware requirements. Operationally, the system is easy to deploy and user-friendly, making it practical for real-time fraud detection in e-commerce platforms real-world challenges in road maintenance and safety.

4.1 Technical Feasibility

The system is technically feasible as it is built using reliable and well-supported technologies such as Python, Flask, and machine learning libraries like scikit-learn and xgboost. These tools are widely used and compatible with most modern operating systems. The implementation does not require high-end hardware, and all software components are open-source, making the system easy to develop, maintain, and scale as needed.

4.2 Operational Feasibility

Operational feasibility examines whether the developed system can function effectively in a real-world setting. The proposed fraud detection system is designed to operate within the backend infrastructure of e-commerce platforms, providing real-time decision-making capabilities. The system interfaces seamlessly with transaction processing pipelines and provides a user-friendly dashboard for administrators to monitor flagged transactions. Its high adaptability, user experience, and minimal technical complexity for end-users affirm the project's operational viability.

Additionally, the use of Flask-based deployment ensures that the system is lightweight, modular, and easy to integrate with existing platforms. The continuous learning feature enhances long-term usability by adapting to evolving fraudulent behavior patterns, further validating its operational sustainability.

4.3 Economic Feasibility

The economic feasibility evaluates the cost-effectiveness of the system and whether the benefits outweigh the expenditures. This system has been developed using open-source technologies such as Python, Flask, and XGBoost, reducing software licensing costs significantly. Development and deployment were conducted using minimal hardware a standard PC with 4GB RAM and basic I/O peripherals minimizing initial setup costs.

4.4 Legal Feasibility

Legal feasibility assesses whether the proposed system adheres to all regulatory and legal requirements. The system complies with data protection norms such as the General Data Protection Regulation (GDPR) and India's Personal Data Protection Bill (PDPB). It does not store personally identifiable information (PII) without encryption and ensures user anonymity wherever possible. Moreover, synthetic datasets have been used for development and testing, which prevents the misuse of real customer data and aligns with ethical AI principles.

4.5 Schedule Feasibility

Schedule feasibility refers to the ability to complete the project within an acceptable timeframe. The project was divided into well-defined stages requirement gathering, data generation, model development, training, testing, and deployment each with specific deliverables and timelines. The use of Agile development methods enabled iterative improvements and milestone tracking.

The entire development cycle was completed within the academic semester schedule, demonstrating that similar systems can be built and deployed in real business scenarios within a 3–6 month timeframe. The use of pre-built Python libraries and machine learning frameworks also accelerated the timeline, reinforcing the project's schedule feasibility.

5. Methodology

5.1. Dataset

- The final dataset consists of 23,634 records and includes 16 attributes
- E-commerce Fraud Detection Uploading Dataset
- E-commerce Fraud Detection Prediction Dataset

5.2 Preprocessing and Augmentation

To ensure high-quality input for the machine learning models, the dataset underwent several preprocessing and augmentation steps. Initially, missing values were addressed using statistical imputation techniques mean for numerical fields and mode for categorical ones.

5.3 Model Architecture

XGBoost (Extreme Gradient Boosting) is a decision-tree-based ensemble model known for its robustness and high performance. It uses gradient boosting with regularization to prevent overfitting. Its inbuilt handling of missing data and fast execution made it ideal for this application.

Stacking involves combining multiple classifiers (base learners) whose outputs are fed into a meta-learner for final prediction.

5.4 Training and Evaluation

Models were trained using an 80-20 train-test split. During training, cross-validation (5-fold) was used to assess generalization and avoid overfitting. Hyperparameter tuning was performed using GridSearchCV to optimize model settings.

The stacking model outperformed standalone models in both precision and recall, making it the preferred solution for deployment.

6. Development Tools

Python was selected as the primary development language for building the e-commerce fraud detection system due to its simplicity, versatility, and strong ecosystem of data science libraries. Its readable syntax and extensive documentation make it accessible for both novice and experienced developers. The availability of high-level libraries such as NumPy, Pandas, and Scikit-learn enabled efficient handling of data preprocessing, transformation, and statistical analysis—critical steps in preparing transactional data for machine learning models. Python's modular

structure also allowed for clean separation between the data pipeline, model training, and deployment components.

Furthermore, Python's strong support for machine learning and deep learning was instrumental in this project. Libraries such as XGBoost and the mlxtend package were used for building powerful ensemble models, including the stacking classifier. These libraries offer optimized performance and ease of integration with existing Python data workflows. Additionally, visualization tools like Matplotlib and Seaborn were employed to explore data distributions, correlation matrices, and model evaluation results such as ROC curves and confusion matrices, aiding in better decision-making and model tuning.

On the deployment side, Python's Flask microframework provided a lightweight yet robust solution for serving the trained model in a web-based environment. Flask enabled the development of APIs that process incoming transactions in real time and return fraud probability predictions to the front-end interface.

7. Module Description

7.1 Data Upload Module

This module allows administrators or system operators to upload transaction data for analysis. It supports structured files such as CSV or Excel formats containing fields like transaction ID, amount, customer age, payment method, and device type. The module validates data formats, handles missing values, and initiates preprocessing steps automatically upon upload. This ensures that only clean and consistent data is passed on to the machine learning pipeline.

7.2 Prediction Module

The prediction module is the core component of the system. It takes processed input features and passes them through the trained machine learning models—namely, the XGBoost Classifier and the Stacking Classifier. These models analyze the transaction attributes and output a probability score indicating the likelihood of fraud. Based on a predefined threshold, the system classifies each transaction as either legitimate or fraudulent. This module also supports individual and batch prediction functionalities, providing flexibility for different use cases.

8. System Architecture

The proposed system architecture includes:

- Data Collection: This layer handles the input of transaction data
- Machine Learning: Core fraud detection is performed in this layer using trained models such as XGBoost and Stacked Classifier.
- Application Backend: Built using Python and Flask, this layer manages data routing
- Frontend and Visualization: This web-based interface, developed using HTML, CSS, and JavaScript

9. System Design and Implementation

The design and implementation of the proposed e-commerce fraud detection system follow a structured, modular approach to ensure efficiency, accuracy, and real-time usability. The system is divided into multiple components that work together to collect data, process it, make predictions, and present results in a user-friendly format.

The system is implemented using Python as the core development language, with Flask used to develop the backend server. The machine learning models (XGBoost and Stacked Classifier) are trained using preprocessed transaction data, and they are integrated into the backend to allow real-time predictions. Frontend development is done using HTML, CSS, and JavaScript. offering an interactive user interface for uploading data, viewing prediction results, and monitoring model performance.

10. Results and Discussion

The integration of ensemble techniques showed promising results in handling high-dimensional, imbalanced data. However, challenges such as generalization across different platforms and real-world dataset limitations remain:

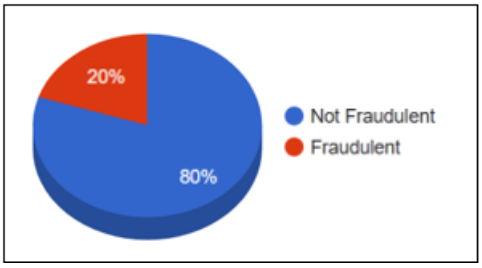

Figure 1: Accuracy

11. Conclusion

The rapid expansion of e-commerce has introduced significant vulnerabilities to online platforms, with fraudulent activities becoming increasingly sophisticated and prevalent. Traditional fraud detection methods are no longer sufficient to counteract the dynamic nature of these threats. This study presents a hybrid machine learning model leveraging Stacking and XGBoost classifiers to enhance the detection of fraudulent transactions in e-commerce environments. Through the generation of a synthetic dataset and the application of advanced preprocessing and classification techniques, the proposed system demonstrates improved accuracy and adaptability in identifying suspicious behaviors. The ensemble learning approach capitalizes on the strengths of individual models, offering a balanced solution that minimizes both false positives and false negatives.

Moreover, the system's ability to learn continuously from new data allows it to stay responsive to emerging fraud trends. The integration of real-time detection mechanisms ensures timely interventions, thereby safeguarding users and merchants against financial loss. Overall, the findings of this research highlight the potential of hybrid machine learning

frameworks in strengthening e-commerce security and pave the way for further innovation in fraud detection systems.

12. Future Work

Looking ahead, several potential enhancements can be made to strengthen the performance and applicability of the proposed e-commerce fraud detection system. One of the key areas for future research is the incorporation of real-world transaction datasets. While synthetic data has proven useful for initial testing, real transactional data—appropriately anonymized—would offer more realistic challenges and improve the system's ability to generalize across different e-commerce environments.

Another promising direction is the use of behavioral biometrics and user profiling. Incorporating features like user navigation patterns, device interactions, and time-of-day behaviors can significantly improve the system's capability to detect anomalies and identity theft in real-time. This would provide a more comprehensive security layer beyond traditional transactional data.

References

[1] Taylor, E. (2024). E-commerce: Using Fraud Detection in ML. *Journal of E-commerce*, 3(3), 12-29.
[2] Saeed, S. (2020). E-commerce Fraud Detection Using Machine Learning: A Survey.
[3] Ahmed, M. (2020). Machine Learning Techniques for Fraud Detection in E-commerce.
[4] Zhang, L. (2021). Anomaly Detection in E-commerce Transactions.
[5] Wang, Y. (2021). Fraud Detection in E-commerce Using Supervised Learning.
[6] Patel, S. (2021). Predicting Fraud in E-commerce Platforms Using Machine Learning.