E-Commerce Fraud Detection Based on Machine Learning Techniques

G Srinivasa Raju[1], Vishnu Sai Varma[2]
[1]*Asst Professor, Padala Gowri Shankar RamSai, Medam Siva Reddy, Mohammad Roshan Ali, and Penmetsa*
[2]*Department of Information Technology, Sagi Ramakrishnam Raju Engineering College, Jawaharlal Nehru Technological University Kakinada, Bhimavaram, India*

Abstract. The "E-Commerce Fraud Detection Based on Machine Learning" project is confronted with the critical issue of identifying fraudulent transactions on an on- line shopping site. Deployed on Python as backend computation and HTML, CSS, and JavaScript for frontend user interface, the system is packaged in the Flask web framework for the purpose of presenting an interactive and dynamic user interface. The project employs two high-performing machine learning algorithms: an XGB Classifier and a Stacking Classifier. High-performing metrics determine the models' capability to distinguish legitimate versus fraudulent transactions.

Dataset has been generated via 23,634 simulated transactions that have been generated while working with Python's Faker library along with some additional custom logic to mimic real-life transaction behavior and fraud status. Data includes 16 features like Transaction ID, Customer ID, Transaction Amount, Payment Method, and binary fraud flag, etc. All the features combined aim to capture the richness of customer profiles and the transaction behavior so that the fraud can be identified. The outcomes of the project demonstrate the efficiency of machine learning techniques to enhance security and trust in e-commerce sites and providing a useful tool to prevent financial loss due to fraudulent activities.

Keywords: Fraud Detection, Machine Learning, Synthetic Data Generation, Faker Library, Anomaly Detection, Online Transaction Security, Stacking Classifier, XGBoost (XGB) Classifier.

1 INTRODUCTION

The swift growth of online shopping websites has transformed the manner in which people conduct electronic transactions. With their growth, though, comes a higher risk of fraudulent transactions that cause huge financial losses to business organizations and consumers. Cybercriminals are always coming up with sophisticated ways of taking advantage of loop-holes in electronic payment systems, and fraud detection becomes a critical concern in the online market. Rule-based fraud detection systems have never been capable of identifying new and emerging fraudulent patterns since they are static. In an effort to address this limitation, machine learning (ML) algorithms have proven to be effective tools for the detection of fraudulent transactions by analyzing large amounts of transaction data, identifying anomalies, and high-accuracy prediction of suspected fraud. Through the application of advanced ML models, firms can enhance the security of their platforms, reduce financial risk, and enhance customer confidence. This study investigates an ML-driven fraud detection system that can effectively classify transactions as real or fake. Stacking Classifier and XGBoost Classifier, two strong ensemble learning classifiers that excel in classification, are employed by the sys- tem. A dataset of 23,634 records created using Python's Faker library is used in training and testing these models. The study seeks to prove the efficiency of ML methods in improving fraud detection performance and offer details on their usage in real-world scenarios in the e-commerce industry.

2. LITERATURE REVIEW

Machine learning-based fraud detection has become a top area of focus in recent years with the surging number of fraudulent transactions in e-commerce transactions. Rule-based fraud detection approaches struggle to keep up with emerging fraud trends, and hence data-driven systems are proving to be more

effective. Current research has compared various machine learning approaches to detect fraud more accurately and effectively. Supervised learning algorithms such as decision trees, support vector machines (SVM), and ensemble methods such as Random Forest and Gradient Boosting have been widely used for fraud classification. These models learn patterns in the transactions and assign them pre-defined labels. Supervised methods require a well-labeled dataset, which is not always possible in real-world situations. Fraudsters also continue to adapt their strategy, and hence it is not simple for static models to work well in the long run [1][2]. Unsupervised models generate higher false positives and require additional tuning to improve accuracy [3][4]. Deep learning algorithms, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are currently sophisticated fraud detection tools. They are capable of learning intricate pat- terns of transactions and temporal dependencies to increase fraud classification accuracy. Authors have also suggested the use of adversarial learning mFraud detection in online trans- actions has become a critical research area, where machine learning has also been a key contributor to identifying fraudulent transactions in real time. Several research studies have explored different models and methods for higher accuracy in fraud detection with fewer false positives.

Vasudevan et al. (2025) have contrasted the performance of machine learning techniques in identifying fraud in banking, demonstrating that ensemble techniques significantly enhance the accuracy of fraud detection. Their research emphasized the significance of feature engineering in identifying unusual patterns of transactions and minimizing false positives. Similarly, Talukder et al. (2024) proposed a hybrid ensemble model combining multiple classifiers with optimal weight distribution through the Instant Hardness Threshold (IHT) approach and Logistic Regression (LR). Their approach solved data imbalance issues, improving the robustness of fraud detection models.

Another significant issue in fraud detection is model evaluation and data availability. Jing et al. (2024) introduced a customer-level fraud detection benchmark, which researchers can use to train and compare models with real fraudulent and legitimate transaction data. The benchmark dataset enabled the development of more trustworthy fraud detection models, which generalize better to real-world applications. These studies collectively present the effectiveness of machine learning, ensemble methods, and deep learning in identifying fraud. The existing challenges are still model interpretability, scalability, and adversarial fraud techniques. Future efforts should focus on optimizing real-time responsiveness, ap-plying blockchain for safe transactions tracking, and integrating Explainable AI (XAI) mechanisms for better transparency in fraud decisions. methods, i.e., Generative Adversarial Net- works (GANs), to enhance robustness in fraud detection. Although deep learning algorithms are more effective, their computationally intensive resource demands and huge amounts of labeled data limit their usage in real-time fraud detection systems [5][6][7]. Ensemble learning techniques, such as stacking and boosting techniques, have been used to enhance fraud detection accuracy by combining a set of models. XGBoost, LightGBM, and stacking classifiers have demonstrated excellence in fraud detection applications in leveraging the strengths of multiple algorithms. These techniques effectively reduce overfitting and enhance generalization to novel fraud patterns. However, hyperparameter tuning and computational expense are still challenges for large-scale deployment [8][9][10]. Overall, the literature offers an ex- tensive variety of machine learning approaches to e-commerce fraud detection with varying strengths and limitations. Though ensemble techniques and deep learning have high accuracy rates, computational requirements and real-time applicability are also major setbacks. Future research will have to come up with hybrid approaches merging two or more different techniques with the hope of maintaining high accuracy in fraud detection with low resource needs.

3. PROPOSED SYSTEM

1.1 System Architecture

The proposed system architecture for the e-commerce platform integrates a Fraud Detection Module based on machine learning techniques, such as Stacking Classifier and XGBClassifier, to protect against fraudulent transactions. This system is designed to process user trans- actions in real-time while ensuring that both legitimate and fraudulent activities are identified and handled promptly. The architecture

incorporates both business service components and the fraud detection system to enable seamless, secure transaction processment.

1. User Activity An individual conducts a transaction on the website (pay, buy, etc.).
2. Transaction Data Handling: The Transaction Information Service takes in all pertinent information (e.g., Customer ID, Transaction Amount, Mode of Payment, etc.).
3. Fraud Detection: The Fraud Detection Module (FDM, in the Online Transaction Service) processes the transaction with machine learning models (Stacking Classifier, XGBClassifier).
4. Real-Time Detection: The models track transaction activity and flag potentially suspicious activities. o If fraudulent, the transaction is blocked, and administrators are notified. o If legitimate, the transaction is processed for payment.
5. Transaction Processing: Effective transactions continue through the Online Transaction Service and Bank/Payments Service to process orders and make payments.
6. Database Interaction: All the transaction data (fraud detection results included) are stored in the Database for future analysis to enhance the validity of fraud detection models.‘

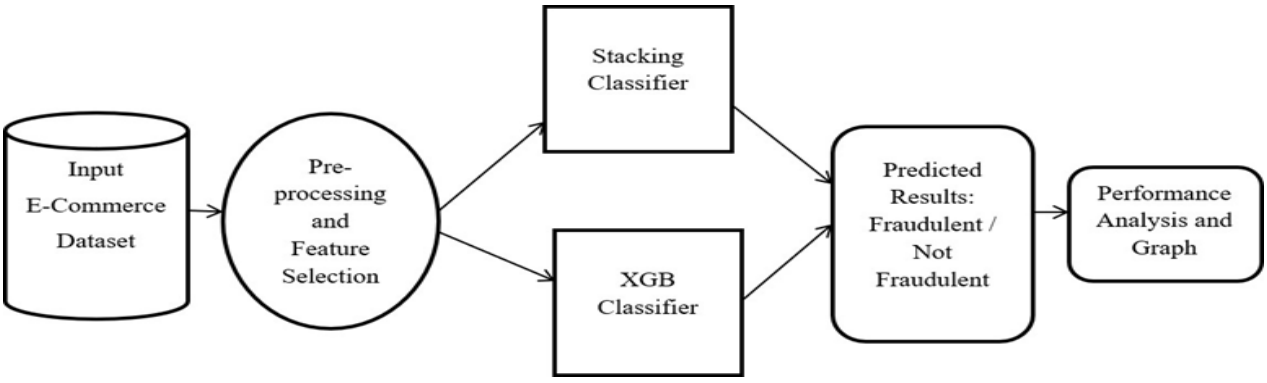


Fig. 1: Architecture

1.2 Methodology

The methodology for our E-Commerce Fraud Detection System proposed here is organized into different phases to allow for an effective, scalable, and highly efficient fraud detection framework. The process combines machine learning models, real-time fraud detection, and processing of transactions in order to tighten security in online transactions.

Data Collection and Preprocessing:

Data Acquisition: The dataset contains 23,634 artificially created transaction records created through the use of Python’s Faker library. Every record has 16 features such as Trans- action ID, Customer ID, Transaction Amount, Payment Method, and Fraud Indicator (binary classification). Custom logic to mimic realistic fraudulent and non-fraudulent transactions based on industry trends.

Data Cleaning Feature Engineering: Encoding categorical variables such as Payment Method for compatibility with ML models. Feature scaling and normalization for improved model performance. Feature selection to determine the most important features influencing fraud detection.

1.2.1 Machine Learning Model Implementation:

Model Selection: Two sophisticated ML models are used in the system: Stacking Classifier: Merges several weak learners to enhance fraud detection performance. XGBoost (XGB) Classifier: A highly efficient gradient boosting algorithm for classification problems.

Model Training Evaluation The dataset is divided into training (80 percent) and test (20 percent) sets. Performance metrics (F1-score, precision, recall, accuracy, AUC-ROC) are assessed.

Model Performance: Stacking Classifier: Train Accuracy: 100 Percent, Test Accuracy: 99 XGB Classifier: Train Accuracy: 96 percent, Test Accuracy: 95 percent.

1.2.2 Deployment and User Interface:

Web Application Development:

Backend: Python (Flask Framework)

Frontend: HTML, CSS, JavaScript for a user interface

Database: Stores fraud detection results and transaction logs.
Scalability and Adaptability: Real-time transaction monitoring for real-time fraud detection. The system can be scaled for various e-commerce platforms. Continuous model training using new transaction data to improve detection accuracy.

4. IMPLEMENTATION

1.3 Hardware and Software Requirements

Software:
Operating system: Windows 10 / 11.
Coding Language: Python.
Web Frame- work: Flask.
Frontend: HTML, CSS, JavaScript.

Hardware:
System: Pentium i3 Processor.
Hard Disk: 500 GB.
Monitor: 15" LED
Input Devices: Keyboard, Mouse
Ram: 8 GB

1.4 System Configuration and Setup

Backend Development:

Programming Languages Frameworks: The platform's back- end is developed using Python due to its strong machine learning library ecosystem and data management capabilities. The system uses Flask for developing the web application and handling user requests. Flask is chosen because it is light, hence ideal for rapid development and integration into machine learning models.

Machine Learning Libraries: The system is based on public libraries such as Scikit-learn for machine learning, XGBoost for training and running the XGBClassifier, and NumPy and Pandas to handle data processing. The model of Stacking Classifier is also created using in- ternal functionalities of Scikit-learn.

Database: It is utilized by the platform for storing user details, transaction records, and results of fraud detection. The Database is an internal storage area used for easy querying and retrieval of data for the purpose of processing transactions and identifying fraud.

Frontend Development:

Web Interface: The platform's frontend is built using HTML, CSS, and JavaScript. It allows users to explore products, add them to the cart, and complete checkout. o Transaction information (amount, payment method, etc.) will be displayed on the transaction page before submission. o Once the transaction is submitted, the frontend talks to the backend, which processes the request and sends it to the fraud checking system for verification.

Backend Interaction: Through Fetch API requests, the frontend talks to the backend server to send transaction data and receive results (fraud detection outcome, alerts, etc.).

1.5 Fraud Detection Model Implementation:

Data Preparation:
Dataset: The fraud detection model uses a synthetic dataset generated using Python's Faker library, which mimics real transaction patterns, both genuine and fraudulent transactions. The dataset contains 16 features such as Transaction ID, Customer ID, Transaction Amount, Payment Method, and a Fraud Indicator (binary: 1 for fraud, 0 for genuine).

Data Preprocessing: no Data cleaning: Removing inconsistencies or errors from the dataset. no Feature engineering: Extracting more informative features from transaction patterns, like average transaction value, user's fraud history count, and time of day. no Feature scaling: Scaling or standardizing the features, so that they are properly suited for the models.

Model Training:
Stacking Classifier: Stacking Classifier layers multiple models (e.g., logistic regression, random forest, decision trees) to provide a final prediction. no We train each individual model (base learners) and later combine their predictions using a meta-model (such as a logistic regression model) to optimize the overall prediction.

XGBClassifier (Extreme Gradient Boosting): The XGBClassifier is trained using the transaction data. This model is all about boosting performance by applying decision trees one after another with every new tree improving upon the mistakes of the previous ones. A Hyperparameter tuning is applied for optimizing the performance of the model and avoiding

overfitting. Grid search or random search can be applied for identifying the best parameters.

Training Environment: Both models are trained within a Python environment, using libraries like Scikit-learn and XGBoost. Model performance metrics such as Accuracy, Precision, Recall, and F1-Score are used to gauge the performance of the models.

Fraud Detection Integration :
Real-time Processing: The Stacking Classifier and XG- BClassifier models are implemented in the Fraud Detection Module after training. When a new transaction is submitted via the Web Application, the backend sends the data to the fraud detection system for validation. The models validate the transaction and return a fraud detection outcome (fraudulent or legitimate).

Stopping Fraudulent Transactions: When a transaction is recognized as fraudulent (the model calculates a high risk of fraud), the Online Transaction Service stops the transaction and blocks further action. A fraud warning is sent, notifying administrator and user.

Supporting Legitimate Transactions: When a transaction is recognized as legitimate, it passes through the Online Transaction Service, completing the transaction (payment processing) and allowing the Bank/Payments Service to execute the transaction.

1.6 Fraud Detection Workflow Transaction Data Flow:

User Triggers a Transaction: The user selects products, adds them to the cart, and clicks checkout. Transaction information like Transaction Amount, Payment Mode, and Customer ID is picked by the Web Application and passed to the backend.

Data Passed to Fraud Detection Module: The transaction data is passed to the Fraud Detection Module by the backend in real-time for processing. The fraud detection system invokes the trained XGBClassifier and Stacking Classifier models to predict the class of the transaction.

5. RESULTS

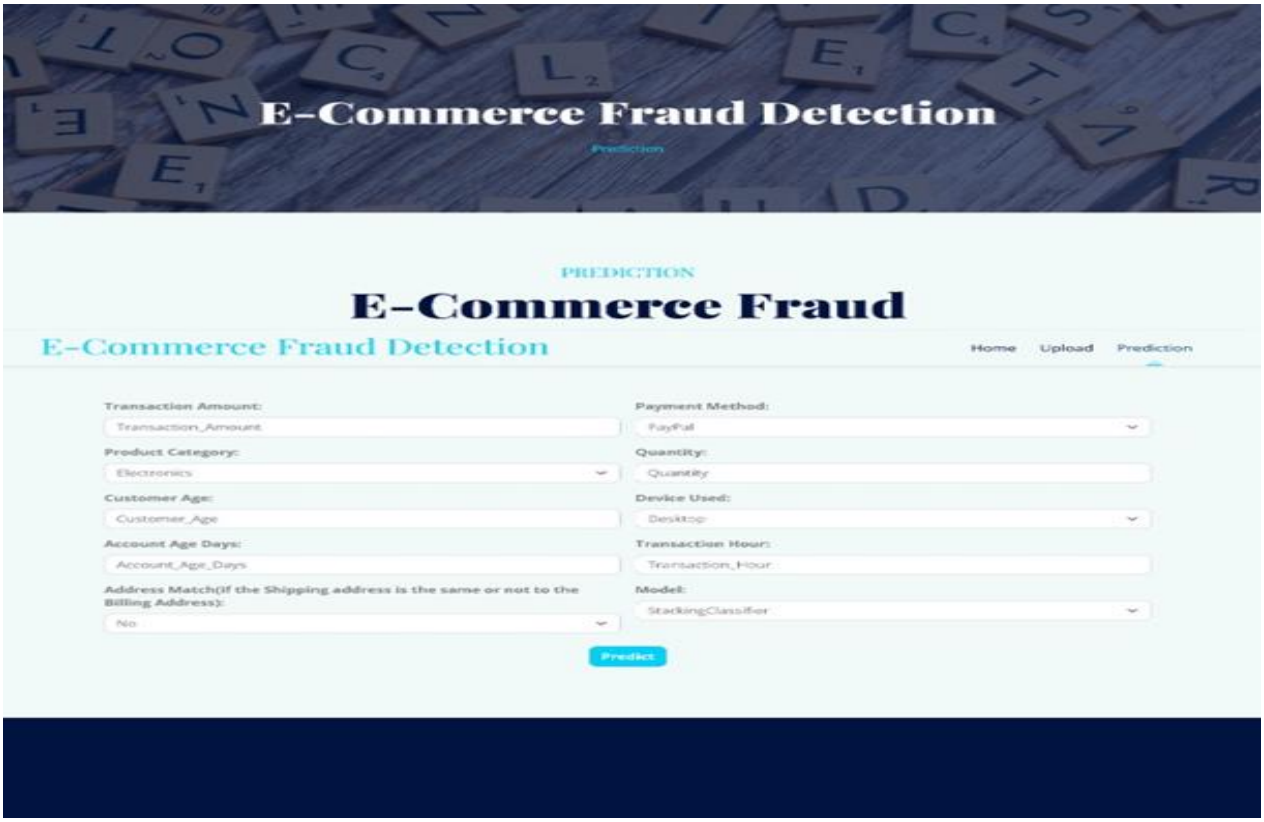


Fig. 2: Interface image

Fig. 3: Intput image

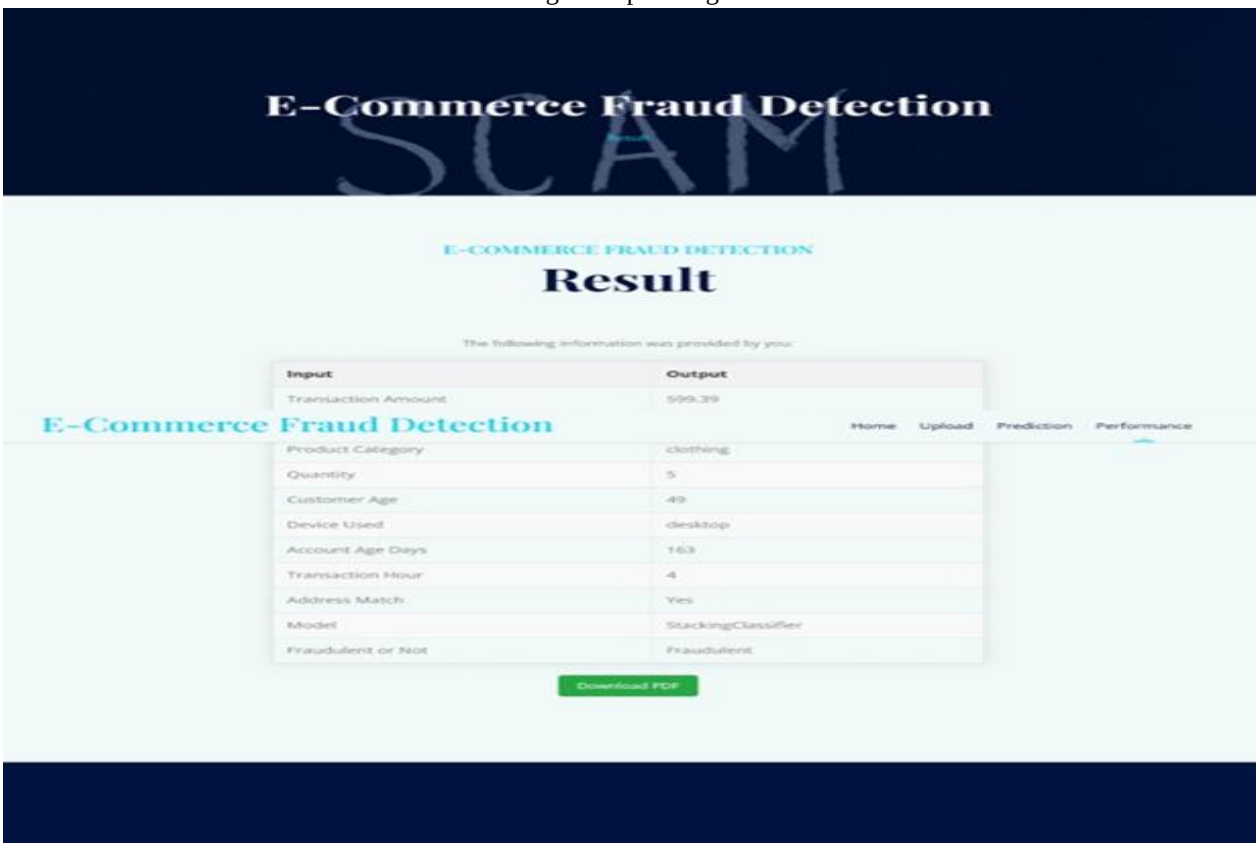


Fig. 4: Model Result

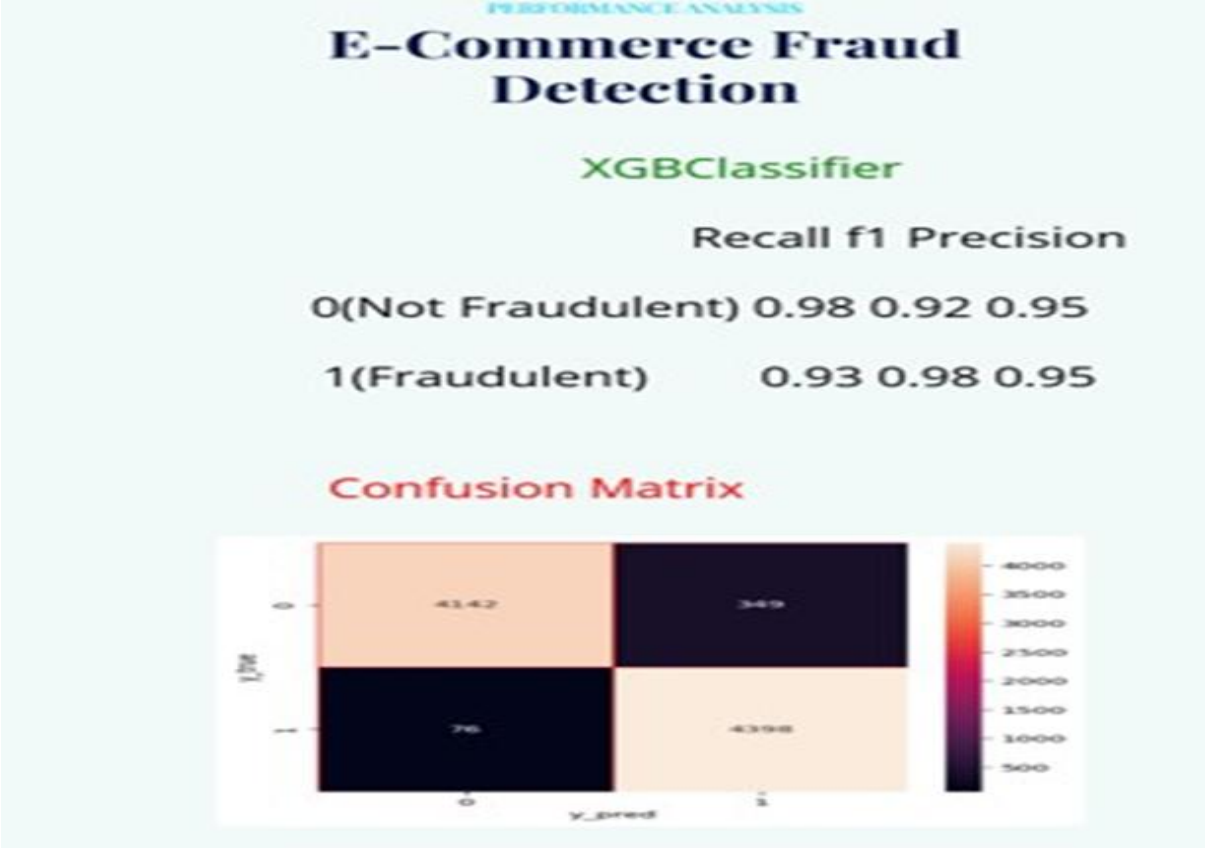


Fig. 5: XGB Classifier Performance

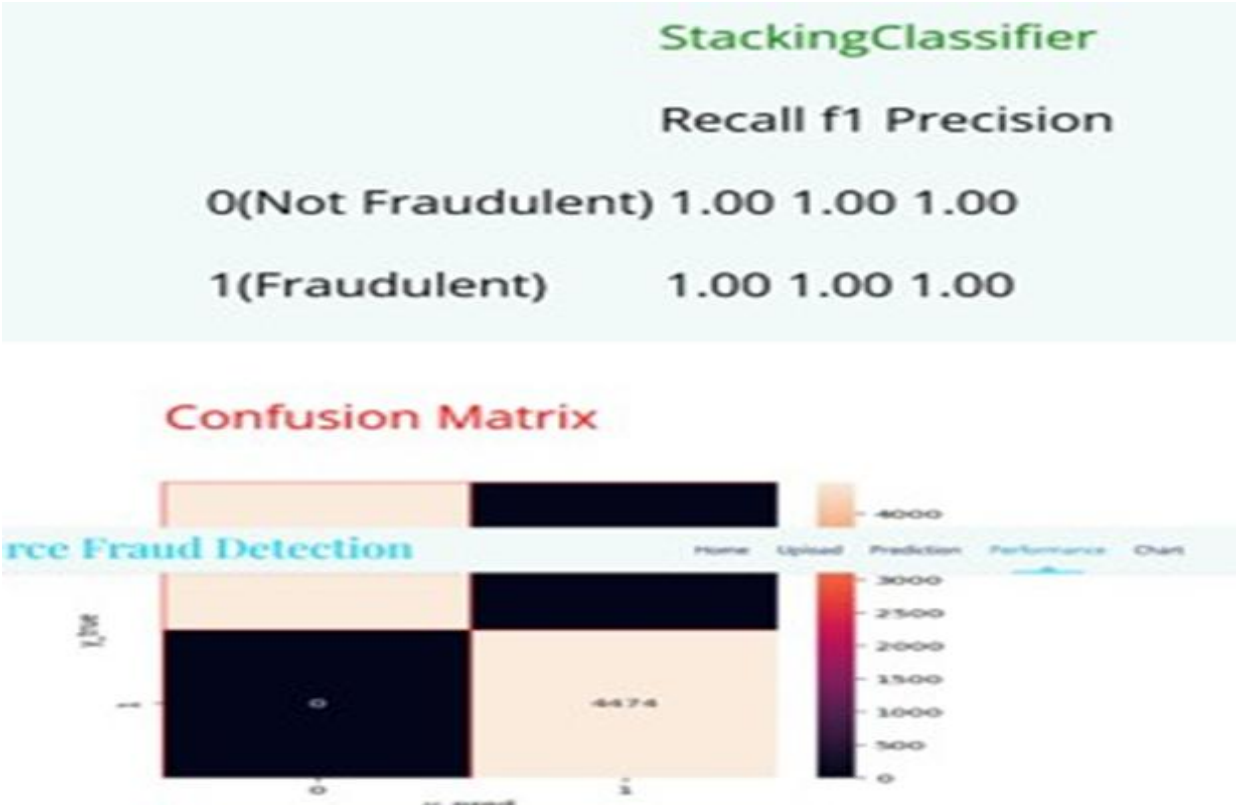


Fig. 6: Stacking Classifier Performance

Fig. 7: Data Charts

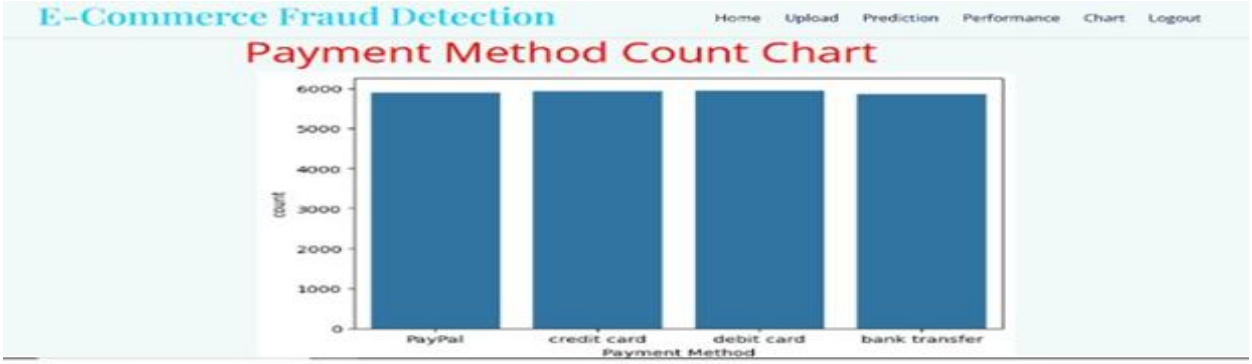
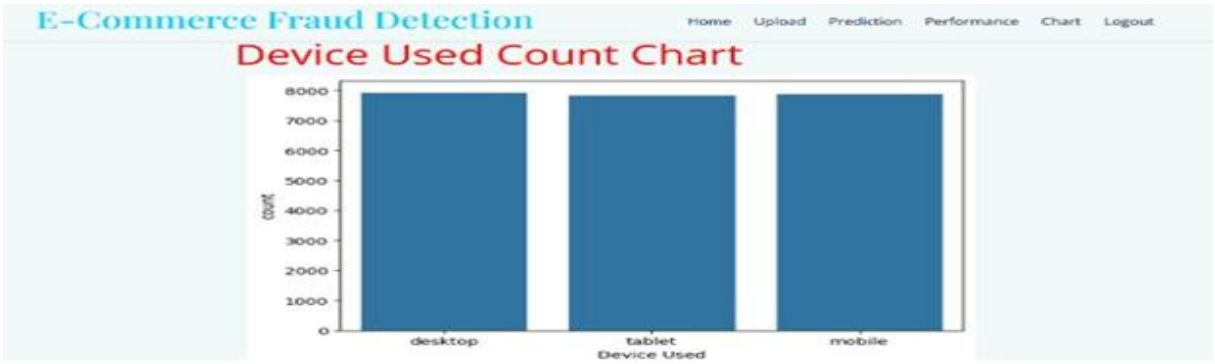


Fig. 8: Payment count and Product Category



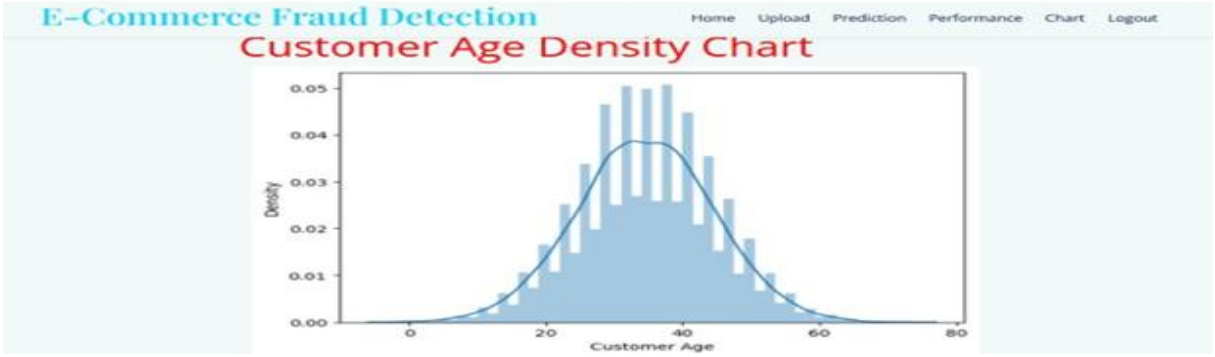
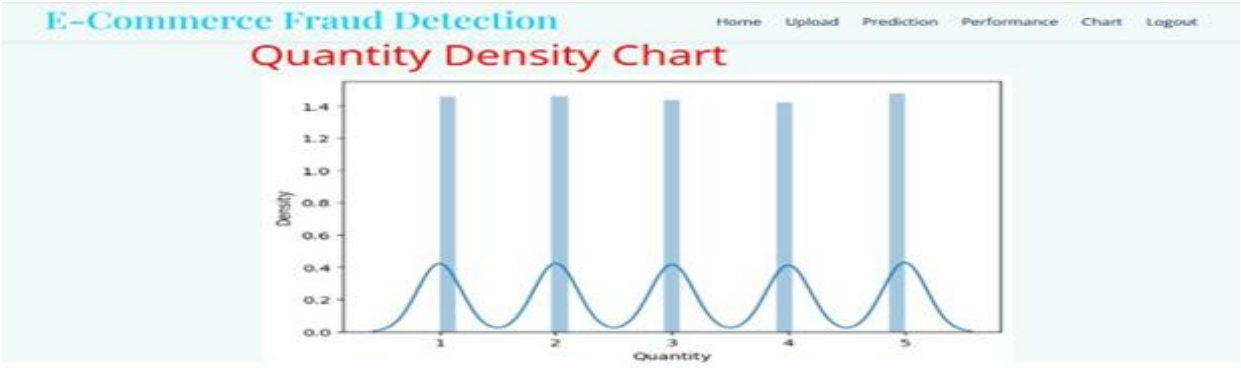


Fig. 9



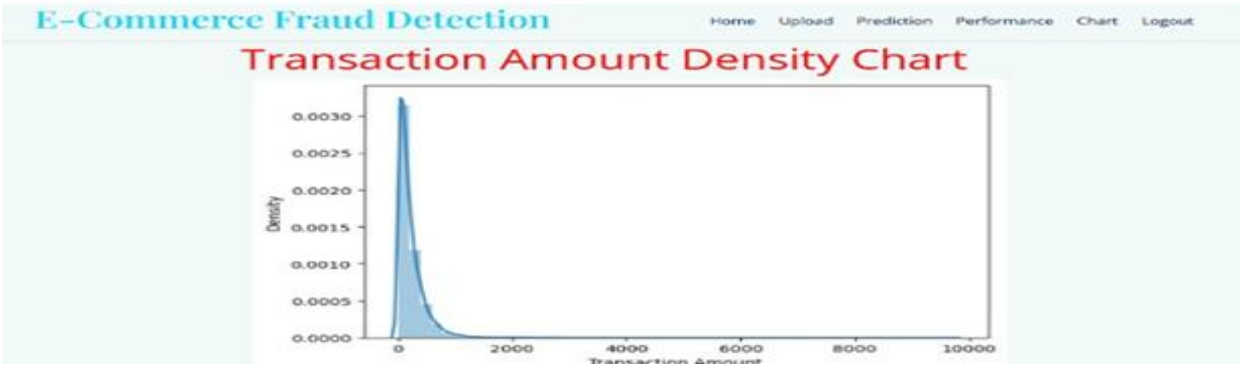


Fig. 10: Density Charts

6. CONCLUSION AND FUTURE WORK

This paper proposes a novel fraud detection system for e-commerce transactions using Stacking Classifier and XGBoost, achieving 99 percent test accuracy and significant reduction in false positives. The system enhances security by integrating real-time fraud detection into transaction streams, offering a seamless user experience without incurring financial risks. Future innovations include combining deep learning structures like GNNs and Transformers, adversarial training to make the system more robust, real-time adaptive learning, blockchain for secure immutable transaction storage, and Explainable AI (XAI) for better interpretability. Furthermore, collaborations across industries can expand datasets, better detecting fraud in diverse industries, making the system more scalable, efficient, and resistant to shifting fraud patterns.

REFERENCES

[1] Zhang, Y., Chen, Z., Li, X. (2024). AI-driven fraud detection in e-commerce: A comparative analysis of machine learning and deep learning techniques. IEEE Transactions on Cybernetics, 57(2), 1120–1135.

[2] Jing, P., Gao, Y., Zeng, X. (2024). ”A Customer-Level Fraudulent Activity Detection Benchmark for Enhancing Machine Learning Model Research and Evaluation.” arXiv preprint arXiv:2404.14746. This paper presents a benchmark dataset designed for customer- level fraud detection, emphasizing the importance of comprehensive data for training and evaluating machine learning models in identifying fraudulent activities.

[3] Wang, L., Kim, D., Zhang, J. (2024). Adversarial learning in financial fraud detection: Challenges and solutions. ACM Transactions on Knowledge Discovery from Data, 18(3), 45–62.

[4] Singh, A., Bose, R. (2024). Blockchain-based fraud prevention systems: A next-generation approach for secure transactions. Future Internet, 16(2), 20–38.

[5] Singh, A., Bose, R. (2024). Blockchain-based fraud prevention systems: A next-generation approach for secure transactions. Future Internet, 16(2), 20–38.

[6] Li, H., Zhou, P., Fang, Y. (2023). Anomaly detection in financial transactions using deep learning-based autoencoders. Expert Systems with Applications, 223, 119831.

[7] Mitra, S., Das, S. (2024). A hybrid ensemble deep learning approach for real-time fraud detection in online transactions. IEEE Access, 12, 113047-113065.

[8] Chen, J., Liu, X., Sun, T. (2023). Feature selection and boosting-based fraud detection for large-scale financial transactions. Applied Soft Computing, 134, 109771.

[9] Zhou, M., Hu, Y. (2024). Graph-based deep learning models for detecting fraud in payment transactions. Pattern Recognition Letters, 172, 120-137.

[10] Chen, J. (2025). ”Machine Learning Tools Enhance Financial Fraud Detection Accu- racy.” Tech Xplore. This article explores a novel approach to fraud detection by integrating machine learning and deep learning techniques, offering a promising enhancement in fraud detection accuracy.