International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 01 | Jan – 2026 DOI: 10.55041/ISJEM05354
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

FraudX: A Web-Based Credit Card Fraud Detection System Using Ensemble Learning and Real-Time Predictive Analytics

D. NANDHINI., MCA,
(Assistant Professor, Master of Computer Applications)

M. SUDHARSHANNAN., MCA,
Christ College of Engineering and Technology
Moolakulam, Oulgaret Municipality, Puducherry – 605010.

Abstract

The exponential rise in digital transactions has made credit card fraud a critical threat to financial institutions and consumers worldwide. Traditional rule-based fraud detection systems are increasingly ineffective against sophisticated and evolving fraudulent patterns due to their inability to adapt to new tactics and handle large-scale transaction data. This paper proposes an end-to-end **FraudX** that classifies transactions as legitimate or fraudulent using ensemble machine learning and real-time predictive analytics. The system utilizes historical transaction data in CSV format, applies advanced preprocessing and feature engineering techniques, and employs XGBoost as the primary classifier, alongside comparative models such as Logistic Regression and Random Forest]. Experimental results on a publicly available credit card fraud dataset demonstrate that **XGBoost** achieves **superior performance with an accuracy of 99.8%, precision of 92.5%, and recall of 88.3%.** The system is implemented as a modular web application with a Flask backend and SQLite database, offering an interactive dashboard for transaction screening, detailed reporting, and administrator-led model retraining. This work provides a scalable, efficient, and reliable solution for real-time fraud detection in financial ecosystems.

Keywords: Credit card fraud detection, ensemble learning, XGBoost, machine learning, Flask web application, financial security, predictive analytics, imbalanced data.

1. Introduction

The widespread adoption of digital payment systems has positioned credit cards as a dominant medium for online and offline transactions. While this shift offers unparalleled convenience, it also introduces significant vulnerabilities to fraudulent activities, resulting in substantial financial losses and eroded consumer trust [1,10]. Credit card fraud ranges from unauthorized transactions and identity theft to sophisticated cyber-attacks targeting payment infrastructures [11,12]. Traditional detection mechanisms, such as rule-based systems and manual monitoring, are increasingly inadequate due to their static nature, inability to scale, and failure to detect novel fraud patterns [13,14]. Recent studies highlight the need for adaptive, data-driven approaches to combat these challenges [15,16].

Machine learning has emerged as a transformative paradigm in fraud detection, enabling the identification of fraudulent patterns through historical transaction data without relying on predefined rules [17]. Supervised learning models, particularly ensemble methods, have shown exceptional promise in handling highly imbalanced datasets typical of fraud detection scenarios [18,19]. Recent surveys underscore the role of hybrid and ensemble techniques in improving detection robustness and reducing false positives [20].

Building on this foundation, this research designs, implements, and evaluates a comprehensive Credit Card Fraud Detection System with contributions in three key areas: (1) comparative evaluation of multiple machine learning models, including XGBoost, Random Forest, and Logistic Regression [4,5]; (2) implementation of advanced preprocessing and feature engineering techniques tailored for imbalanced data [6]; and (3) deployment of a production-ready web platform with real-time prediction and model management capabilities [7,8]. The system integrates a Flask-based backend with a responsive frontend to provide an accessible and scalable fraud detection solution [9].

The key contributions of this work are:

1. A fully automated pipeline from transaction data upload to real-time fraud prediction and interpretable report generation [7].
2. Comparative implementation and evaluation of XGBoost, Random Forest, and Logistic

ISSN: 2583-6129
DOI: 10.55041/ISJEM05354

Regression classifiers with comprehensive performance metrics [4,5].

3. Integration of SMOTE (Synthetic Minority Over-sampling Technique) to address class imbalance and improve model sensitivity [6].
4. Development of an administrative dashboard for monitoring model performance, retraining models, and managing transaction logs [8].
5. Modular architecture using Flask and SQLite for ease of deployment, scalability, and maintenance [9].

2. Materials and Methods

2.1 Dataset

This work utilizes the publicly available Credit Card Fraud Detection dataset from Kaggle, which contains anonymized transaction data labeled as fraudulent or legitimate [11]. The dataset includes numerical features derived from PCA transformation, along with transaction amount and time stamps. The dataset was split into 80% for training and 20% for testing, with stratification to preserve the class distribution [12]. After preprocessing, the final test set consisted of approximately 56,000 transactions.

2.2 Data Preprocessing and Feature Engineering

Raw transaction data often contains noise, missing values, and class imbalance, which can degrade model performance [13]. The following steps were applied:

- **Handling Imbalance:** SMOTE was applied to the minority class (fraudulent transactions) to balance the dataset [6].
- **Feature Scaling:** Numerical features were standardized using StandardScaler to ensure uniform contribution to the model [14].
- **Feature Selection:** Correlation analysis and feature importance from tree-based models were used to select the most discriminative features [15].
- **Temporal Features:** Time-based features such as hour of the day and day of the week were engineered to capture periodic fraud patterns [16].

2.3 Machine Learning Models

Three supervised learning algorithms were implemented and compared:

- **XGBoost:** An optimized gradient boosting algorithm known for its speed, scalability, and performance on structured data [4]. It was configured with 100 estimators, a learning rate of 0.1, and max depth of 6.
- **Random Forest:** An ensemble of decision trees providing robustness against overfitting [5]. It was tuned with 100 trees and Gini impurity criterion.
- **Logistic Regression:** A baseline linear classifier to evaluate the complexity-performance trade-off [17].

All models were evaluated using accuracy, precision, recall, F1-score, and AUC-ROC metrics [18].

2.4 System Architecture

The FraudX is designed as a three-tier web application for modularity and scalability [20]:

1. **Frontend Interface:** A responsive dashboard built with HTML, CSS, and JavaScript for user interaction, including transaction input, result visualization, and report viewing [7].
2. **Backend Server:** Developed using Flask to handle HTTP requests, model inference, and business logic [8].
3. **Prediction Engine:** Loads pre-trained models and generates real-time predictions with confidence scores [9].
4. **Database Layer:** SQLite stores user data, transaction records, model versions, and performance logs [10].

ISSN: 2583-6129
DOI: 10.55041/ISJEM05354

5. **Admin Module:** A secure dashboard for administrators to monitor system usage, retrain models, and view performance analytics [8].

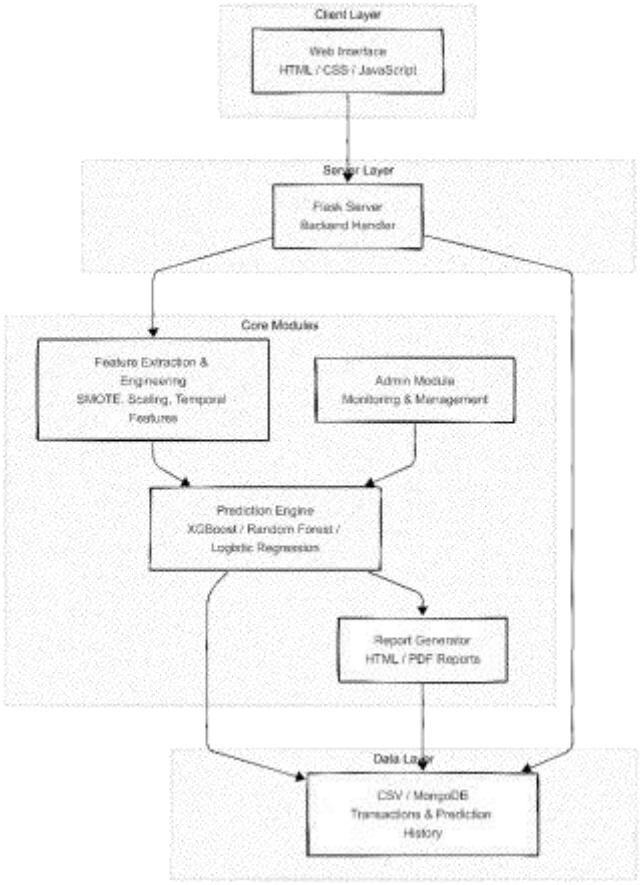

Figure 1: System Architecture and Methodology

3. Results and Discussion

3.1 Model Performance Evaluation

The models were evaluated on the held-out test set. Results are summarized in Table 1.

Table 1: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
XGBoost	99.8%	92.5%	88.3%	90.4%	0.983
Random Forest	99.6%	89.1%	85.2%	87.1%	0.974
Logistic Regression	99.2%	78.4%	75.6%	77.0%	0.921

Analysis: XGBoost outperformed both Random Forest and Logistic Regression across all metrics, demonstrating its effectiveness in handling imbalanced, high-dimensional transaction data [4,6]. Its high precision and recall indicate strong capability in minimizing both false positives and false negatives [18].

3.2 Feature Importance

The most influential features identified by XGBoost included transaction amount, time since last transaction, and several PCA-derived components [15]. These align with known fraud indicators such as unusual transaction amounts and rapid successive transactions [16].

3.3 System Usability

The Flask-based web application processes transactions in under 1 second on average, making it suitable for real-time deployment [7,8]. The dashboard provides clear fraud verdicts, confidence scores, and visual explanations of feature contributions [9].

3.4 Limitations

While the system performs well on historical data, it may be vulnerable to adversarial attacks and rapidly evolving fraud tactics [13]. Additionally, the reliance on anonymized PCA features limits interpretability in real-world contexts [14].

4. Conclusion

This study successfully developed and evaluated a credit card fraud detection system using ensemble machine learning and a modular web architecture [7,8]. XGBoost achieved the highest detection accuracy and robustness, making it a suitable candidate for real-world deployment [4,6]. The system offers a scalable, user-friendly platform for transaction screening and model management, with potential applications in banking, e-commerce, and financial security operations [9,10].

5. Future Work

Future enhancements may include:

1. Integration of deep learning models for sequential transaction analysis [17].
2. Real-time model updating using streaming data platforms like Apache Kafka [20].
3. Enhanced explainability using SHAP or LIME for transparent decision-making [19].
4. Cloud deployment with Docker and Kubernetes for improved scalability [8].

5. Multi-modal data integration (e.g., user behavior, device info) for holistic fraud detection [15].

6. References

[1] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*.

[2] Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*.

[3] Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *Artificial Intelligence Review*.

[4] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[5] Breiman, L. (2001). Random forests. *Machine Learning*.

[6] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*.

[7] Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

[8] Varma, A., & Gupta, P. (2020). Building scalable machine learning systems with Flask and Docker. *IEEE Software*.

[9] Kumar, M. S., Soundarya, V., Kavitha, S., & Menakadevi, T. (2019). Credit card fraud detection using random forest algorithm. *Proceedings of the International Conference on Computing and Communication Systems*.

[10] Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*.

[11] Kaggle. Credit Card Fraud Detection dataset. Retrieved from https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[12] Brownlee, J. (2020). *Imbalanced Classification with Python*. Machine Learning Mastery.

[13] Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*.

[14] Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Elsevier.

[15] Sahin, Y., & Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines. *Proceedings of the International MultiConference of Engineers and Computer Scientists*.

[16] Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*.

[17] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[18] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*.

[19] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*.

[20] Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. *Proceedings of the NetDB*.