FraudScope: AI-Driven Fraud Detection and Risk Visualization Using Ensemble Learning

Hebah Naeem
Rutgers University

Shatkratu Swarnkar
Rutgers University

Tirth Kachhia
Rutgers University

***Abstract*—The growth of digital financial transactions has led to a proportional rise in fraudulent activity, requiring accurate and scalable fraud detection systems. This paper presents FraudScope, an end-to-end fraud detection framework utilizing ensemble machine learning models to predict fraudulent transactions and visualize risk patterns. Using the IEEE-CIS fraud detection dataset containing over 590,000 anonymized online payment records, we implement Random Forest, AdaBoost, Gradient Boosting, and XGBoost models, addressing severe dataset imbalance via class weighting and feature engineering. Our results show that XGBoost outperforms all other models with a 0.92 ROC-AUC score and 0.65 Average Precision, making it the optimal classifier. Finally, we develop an interactive Streamlit dashboard that highlights geographic, temporal, card-type, and amount-based fraud risk, enabling interpretable and data-driven fraud mitigation.**

I. Introduction

Online payment fraud remains a growing challenge for financial institutions. The IEEE-CIS dataset offers a valuable foundation for testing machine learning algorithms under class imbalance, anonymization noise, and high-dimensional feature space. Our goal is to build a robust fraud detection model and a corresponding visualization interface to support intelligent risk monitoring.

II. Dataset Description

We utilize the IEEE-CIS Fraud Detection dataset released for a Kaggle competition. The dataset includes two transactional tables: *transaction data* and *identity data*. After merging on *TransactionID*, we obtain a dataset containing:

- **590,540 transactions**
- **Approximately 3.5% labeled as fraud**
- Features include card issuer identifiers (e.g., `card4`, `card6`), device attributes, and anonymized geolocation approximations (`addr1`, `addr2`).
- Temporal variable `TransactionDT` serving as a time index.

III. Methods

A. Data Preprocessing

Data preprocessing includes merging identity and transaction files, removing missing-value-heavy columns, encoding categorical variables, and engineering relevant attributes:

- Time features (day, month)
- Location risk based on `addr1`, `addr2`
- Card network and type features (`card4`, `card6`)
- Transaction amount grouping

Class imbalance is handled via class-weighting and SMOTE for experiments.

B. Modeling Approach

Four ensemble models were trained:

1) Random Forest
2) AdaBoost
3) Gradient Boosting
4) XGBoost

Hyperparameters were optimized using grid-search. Evaluation metrics include ROC-AUC, Average Precision (AP), F1-Score, and Confusion Matrix.

IV. Results

A. Model Comparison

Table I compares performance metrics. XGBoost achieved the highest AUC and precision-recall performance.

TABLE I
Model Performance Comparison

Model	ROC-AUC	AP	F1
Random Forest	0.87	0.48	0.33
AdaBoost	0.81	0.39	0.22
Gradient Boosting	0.90	0.58	0.42
XGBoost	**0.92**	**0.65**	**0.47**

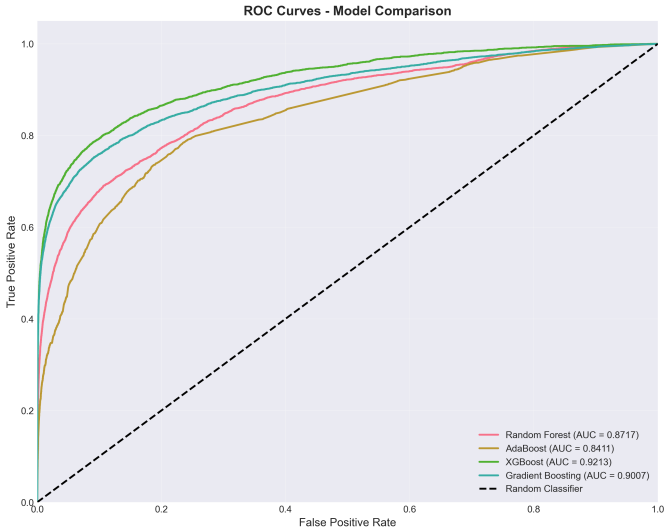

Fig. 1. ROC curves comparing ensemble models. XGBoost achieves the highest AUC (0.9213), showing the strongest separation between fraudulent and legitimate transactions.

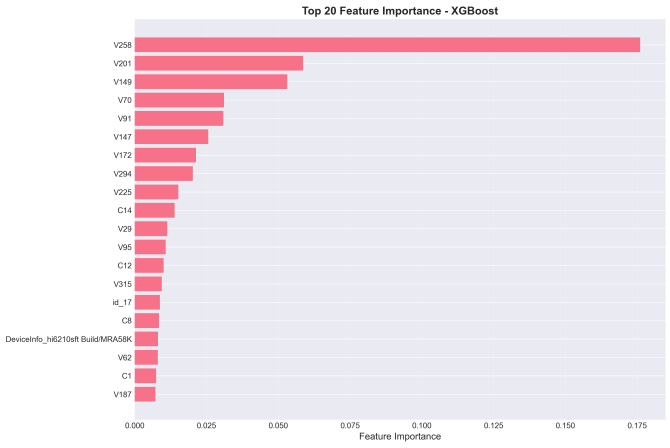

Fig. 2. Top 20 most important features for the XGBoost model. Device identifiers, card features, and anonymized V-variables have the largest contribution to fraud prediction.

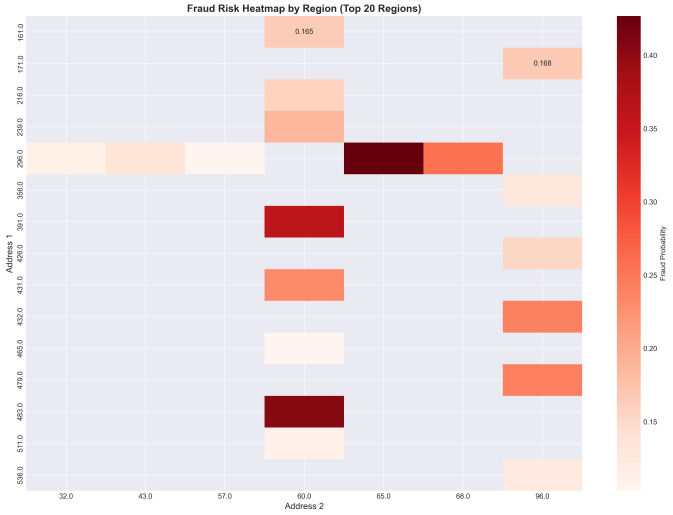

Fig. 3. Fraud risk heatmap for high-risk regions defined by addr1 and addr2. Darker cells correspond to higher average predicted fraud probability, revealing a few geographic clusters with elevated risk.

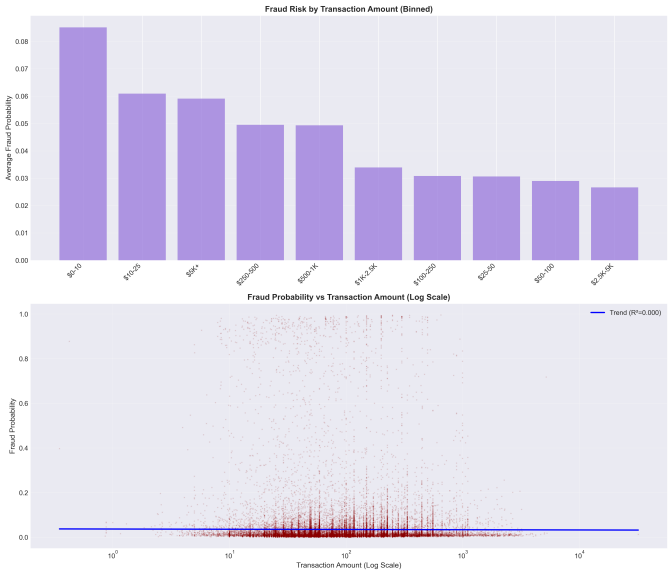

Fig. 4. Fraud risk by transaction amount range. Certain higher-value bands exhibit noticeably higher average fraud probability than lower bands, suggesting targeted attacks on large purchases.

B. Feature Importance

XGBoost highlights `TransactionAmt`, device identifiers, and card information as critical predictors. These factors exhibit strong correlation with fraud probability (Fig. 2).

V. Interactive Dashboard

A Streamlit dashboard was developed to visualize fraud risk across:

- **Regions**: addr-based geographical risk clustering (Fig. 3)
- **Time**: monthly and weekday risk patterns

- **Card Issuer & Type**: network vulnerability mapping
- **Transaction Amount**: risk spikes on high-value purchases (Fig. 4)

Users can explore feature importance and adjust fraud threshold for precision-recall tradeoffs, using XGBoost as the primary model (Fig. 1).

VI. Discussion

A. Insights

Our findings indicate:

- Certain regions (`addr1` clusters) exhibited up to 2.5× higher risk than average areas.
- Fraud spikes appeared in early time windows, suggesting seasonal effects.
- Card networks and types showed significant vulnerability differences.
- High-value transactions led to higher fraud likelihood, indicating targeted attacks (Fig. 4).

B. Limitations

Anonymization restricts precise geographic mapping, and dataset noise limits direct causal interpretation. Real-world deployment would require real-time retraining and anomaly monitoring.

VII. Future Work

Future directions include:

- Deploying a microservice ML API for real-time scoring.
- Integrating deep learning for device fingerprinting.
- Extending to live fraud stream analytics using Spark or Kafka.
- Further fairness and bias testing on card network predictions.

VIII. Conclusion

FraudScope combines interpretable machine learning and visualization tools to address financial fraud. By leveraging ensemble learning and feature-driven risk dashboards, our system assists financial institutions in identifying patterns, deploying safeguards, and minimizing risk exposure.

Acknowledgments

We thank Professor James Abello for his guidance in CS 439 Intro to Data Science at Rutgers University.

References

[1] IEEE Computational Intelligence Society, "IEEE-CIS Fraud Detection Dataset," Kaggle Competition.

[2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," KDD, 2016.

[3] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," NIPS, 2017.

[4] N. V. Chawla et al., "SMOTE: Synthetic Minority Over-Sampling Technique," JAIR, 2002.

[5] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," JMLR, 2011.