Advanced Machine Learning Methods for Fraud Detection

Abrhaley Hailenchael

[1]Warsaw University of Technology, Warsaw, Poland
abrhaley_arefaine.hailenchael.stud@pw.edu.pl

Abstract. Detecting fraud in financial institutions is increasingly complex. This study leverages the IEEE-CIS Fraud Detection dataset and advanced machine learning techniques. Key innovations include MICE for missing data, hybrid SMOTE-ENN for class balancing, robust feature engineering, and PCA for dimensionality reduction. Feature selection combined correlation analysis with statistical tests like Chi-Square and Cramér's V. Among tested models, XGBoost excelled, showcasing the potential of advanced techniques in improving fraud detection accuracy for financial institutions.

Keywords: Fraud Detection · Machine Learning · Advanced Techniques · IEEE-CIS Dataset · Financial Fraud · Feature Engineering · Random Forest · SVM · Deep Learning · Ensemble Methods · Precision-Recall ·

1 Introduction

Financial fraud is increasingly complex, and traditional detection methods often face high false positives and limited adaptability. This thesis leverages the IEEE-CIS Fraud Detection dataset to enhance fraud detection through advanced preprocessing techniques like MICE for missing values, hybrid SMOTE-ENN for class balancing, robust feature engineering, and PCA for dimensionality reduction.

Machine learning models, including Naïve Bayes, Logistic Regression, SVM, Random Forest, XGBoost, and Neural Networks, are evaluated for detection accuracy. By optimizing these approaches, this research aims to provide scalable, practical solutions for financial institutions to combat fraud effectively [1,2,3,4.5].

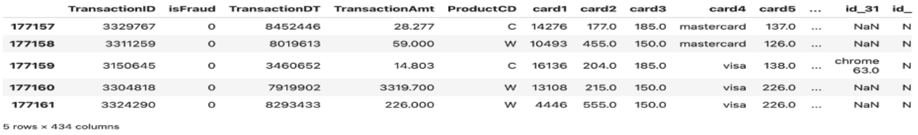
adfa, p. 1, 2011.

2 Methodology

2.1 Dataset Overview

The IEEE-CIS Fraud Detection Dataset, sourced from Vesta Corporation and Kaggle, includes 1,097,231 instances and 434 features across four files containing transaction and identity data, with some missing values and a severe class imbalance (3.5% fraudulent transactions), linked by TransactionID, posing challenges in missing data, sparsity, and class imbalance[3].

Table 1. Data overview



	TransactionID	isFraud	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	card4	card5	...	id_31	id_
177157	3329767	0	8452446	28.277	C	14276	177.0	185.0	mastercard	137.0	...	NaN	N
177158	3311259	0	8019613	59.000	W	10493	455.0	150.0	mastercard	126.0	...	NaN	N
177159	3150645	0	3460652	14.803	C	16136	204.0	185.0	visa	138.0	...	chrome 63.0	N
177160	3304818	0	7919902	3319.700	W	13108	215.0	150.0	visa	226.0	...	NaN	N
177161	3324290	0	8293433	226.000	W	4446	555.0	150.0	visa	226.0	...	NaN	N

5 rows × 434 columns

2.2 Data Preprocessing

EDA.

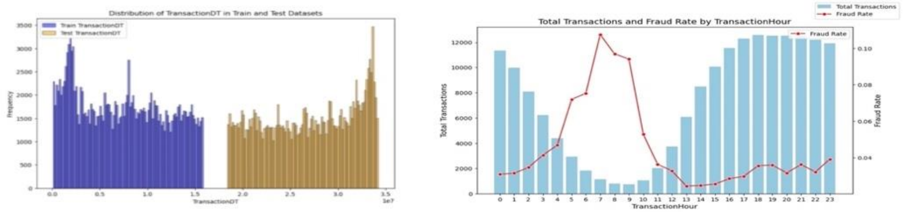


Fig. 1. Time based graphs of the data

Feature Engineering.

Feature engineering improves model performance by capturing key patterns in user behavior and transaction timing. In this study, aggregated features like transaction count, total amount, and average transaction value summarize spending habits. Time-based features such as transaction frequency, time since the last transaction, and day-of-week indicators identify patterns in transaction timing, helping the model detect irregular or fraudulent activities.

Advanced Missing Value and Outlier Handling.

A hybrid approach combining MICE (Multiple Imputation by Chained Equations) with mean and median imputation was used for handling missing data, improving data completeness. Mode imputation was applied to categorical features, and numerical features were normalized while categorical variables were one-hot encoded. Outlier detection was performed using the Interquartile Range (IQR) method, which is robust to skewed data and does not assume a normal distribution, ensuring accurate identification of extreme values without distorting the data.

Categorical Encoding and Missing Value Handling.

To handle missing categorical values, mode imputation was used to fill entries with the most frequent category, ensuring consistency. Encoding techniques were selected based on the number of unique values and feature nature:

- **Binary Encoding**: Applied to features with two unique values (e.g., M1-M9, id_12, id_16, DeviceType), mapping categories to 0s and 1s.
- **One-Hot Encoding**: Used for features with a small number of categories (e.g., ProductCD, card4, card6), creating binary columns for each category.
- **Frequency Encoding**: Applied to features with moderate category counts (e.g., P_emaildomain, id_31), replacing categories with their frequency.
- **Hash Encoding**: Used for high-cardinality features (e.g DeviceInfo), transforming categories into fixed-length hashed values to reduce dimensionality and memory.

Normalization and Transformation Techniques.

For numerical features, standard scaling was applied to ensure a mean of 0 and a standard deviation of 1, making the data suitable for machine learning algorithms. Skewed data was handled using the following:

- **Log Transformation**: Applied to positively skewed data (e.g., TransactionAmt) to reduce skewness and stabilize variance.
- **Johnson Transformation**: Used for negatively skewed features (e.g., id_07, V41, V65) to make the distribution closer to normal.

Handling Class Imbalance.

To address the class imbalance (3.5% fraudulent transactions), I applied a hybrid SMOTE + ENN technique. SMOTE generates synthetic samples for the minority class, while ENN removes noisy data points, creating a balanced and cleaner dataset. This approach improves the accuracy and stability of the models.

Comprehensive Feature Selection.

I employ a combination of techniques for effective feature selection:

- For categorical variables, statistical tests such as Chi-Square, P-Value, Degrees of Freedom, and Cramér's V are used to retain the most impactful features.
- Principal Component Analysis (PCA) is applied to reduce dimensionality while preserving essential information, selecting at least 20 components.

This approach selects relevant features, improving efficiency and model performance beyond simpler methods in previous research.

3 Optimized Hyperparameter Tuning

To improve the performance and generalization of my machine learning models, I use advanced hyperparameter tuning techniques:

- **Randomized Search**: Efficiently explores a broader parameter space by randomly sampling combinations, ideal for high-dimensional datasets.

4 Model Selection

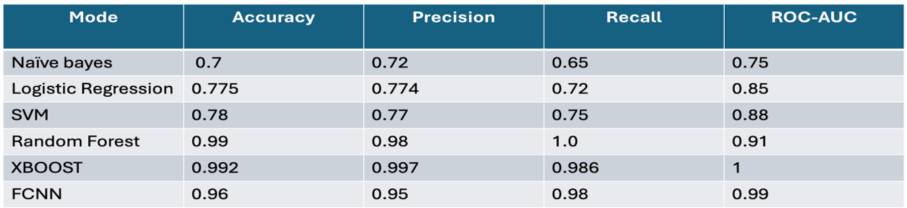
The models are categorized from more traditional to advanced methods based on their complexity and ability to capture intricate patterns in fraud detection:

1. **Traditional Models**:
   Naive Bayes: A simple probabilistic classifier based on Bayes' theorem, effective for categorical data and quick predictions, particularly in imbalanced datasets[5].
   Logistic Regression: A linear classifier that models the probability of class membership, offering interpretable results and efficient performance for binary classification tasks [1][3],[5].
2. **Discriminative Models**:
   Support Vector Machine (SVM): A robust model that finds the optimal decision boundary between classes, excelling in high-dimensional spaces and handling complex, non-linear patterns [1],[5].
3. **Ensemble Methods**:
   Random Forest: An ensemble method combining multiple decision trees to reduce overfitting and improve model accuracy, particularly effective for complex datasets with many features [1],[2],
   XGBoost: A gradient boosting algorithm that sequentially builds trees to minimize errors, known for its high performance and efficiency in classification tasks [2][4].
4. **Deep Learning**:
   Fully Connected Neural Network (FCNN): A deep learning model with multiple hidden layers capable of learning highly complex relationships and patterns in large datasets, making it powerful for fraud detection tasks requiring advanced pattern recognition.

This progression from traditional to advanced models enables a thorough exploration of different classifiers, aiming to find the best one for fraud detection by identifying complex patterns and anomalies.

5 Experimental Results

Table 2. experimental results



Mode	Accuracy	Precision	Recall	ROC-AUC
Naïve bayes	0.7	0.72	0.65	0.75
Logistic Regression	0.775	0.774	0.72	0.85
SVM	0.78	0.77	0.75	0.88
Random Forest	0.99	0.98	1.0	0.91
XBOOST	0.992	0.997	0.986	1
FCNN	0.96	0.95	0.98	0.99

Naïve Bayes performed modestly with 70% accuracy and an ROC-AUC of 0.75, struggling with the dataset's complexity. Logistic Regression improved upon this with 77.5% accuracy and an ROC-AUC of 0.85, offering a reliable baseline. SVM showed balanced results (78% accuracy, 75% recall, ROC-AUC 0.88), handling non-linear patterns better.

Random Forest achieved near-perfect recall (100%) and 99% accuracy, though its ROC-AUC of 0.91 indicated minor room for improvement. XGBoost was the best performer, with 99.2% accuracy, 98.6% recall, and a perfect ROC-AUC of 1, demonstrating superior handling of imbalanced data. FCNN also performed well (96% accuracy, 98% recall, ROC-AUC 0.99) but slightly lagged behind XGBoost.

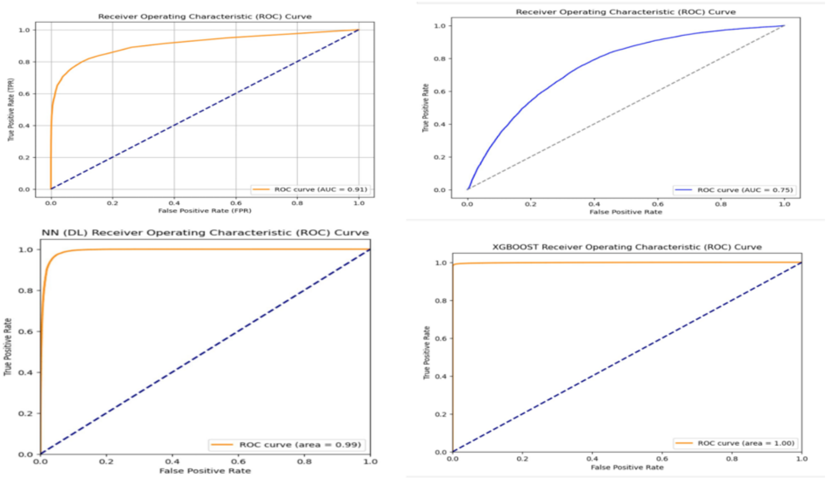


Fig. 2. ROC-AUC graph of top 4 models

6 Best Metrics for Imbalanced Datasets

AUC-ROC: A robust metric for imbalanced datasets, assessing the model's ability to distinguish classes across thresholds. It provides a clearer evaluation of performance for both fraudulent and non-fraudulent transactions.

F1-Score: Balances precision and recall, crucial for fraud detection, ensuring minimal false positives and negatives while effectively identifying rare events.

7 Conclusion and Future Work

This study demonstrates the effectiveness of advanced machine learning models, particularly ensemble methods like XGBoost and deep learning using FCNN, in detecting fraudulent transactions. Feature engineering significantly improved model performance, highlighting the importance of handling imbalanced data and implementing effective preprocessing strategies.

Future work will focus on enhancing model interpretability through Explainable AI (XAI) techniques, such as SHAP values and LIME, to improve transparency and build trust with stakeholders in fraud detection systems.

8 Reference

1. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, **50**(3), 602–613.
2. Breiman, L. (2001). Random forests. *Machine Learning*, **45**(1), 5–32.
3. Howard, A., Bouchon-Meunier, B., Lei, J., Vesta, M. (2019). IEEE-CIS Fraud Detection. Kaggle. Available at: https://kaggle.com/competitions/ieee-fraud-detection.
4. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.
5. Ding, X., Liu, B., Yu, P. S., & Ishwaran, H. (2018). Boosted decision tree approaches for insurance fraud detection. *Journal of Financial Crime*, **25**(2), 403–419.