A Hybrid Approach for Fraud Detection in Online Transactions

Sanjith Senniangiri
Computer Science
ASU
Tempe, AZ, US
ssennian@asu.edu

Muraliprasath Ponnuswamy
Computer Science
ASU
Tempe, AZ, US
mponnus1@asu.edu

Radhika Pareek
Computer Science
ASU
Tempe, AZ, US
rpareek3@asu.edu

Manali Kishore Gawande
Computer Engineering
ASU
Tempe, AZ, US
mgawand1@asu.edu

Xiaaou Liu
Computer Science
ASU
Tempe, AZ, US
xiaaouli@asu.edu

ABSTRACT

As online transactions continue to grow, detecting fraud in real-time has become a major challenge for banks, financial institutions, and e-commerce platforms. Fraudulent transactions are rare events in a sea of legitimate activities, making them hard to spot and often resulting in biased models that favor the majority (non-fraud) class. This project explores a hybrid approach to fraud detection that combines anomaly detection with traditional classification techniques to improve our ability to catch these rare events.

In our approach, we start with anomaly detection models like Isolation Forest and One-Class SVM to filter out potential outliers, thereby reducing noise for the classification step. This pre-filtered data is then passed through supervised classifiers like Random Forest, Naive Bayes, and Decision Tree to make final fraud predictions. By blending these techniques, we aim to leverage the strengths of both approaches to better identify fraudulent transactions in a highly imbalanced dataset.

Using the IEEE-CIS Fraud Detection Dataset, which includes transaction records and features such as customer demographics and device information, we tested multiple hybrid combinations. Early results suggest that combining Isolation Forest with Random Forest is especially effective, providing balanced performance in metrics like accuracy, precision, recall, F1 score, and AUC-ROC. This hybrid method shows promise as a practical solution for enhancing fraud detection in online transactions.

KEYWORDS

Fraud Detection, Anomaly Detection, Supervised Classifiers

1 Introduction

The rise of online transactions has brought convenience but also a surge in fraudulent activities. Detecting fraud in this environment isn't easy—fraudulent transactions are rare compared to the sheer volume of legitimate ones, leading to a significant class imbalance. Traditional supervised machine learning models often fall short in this area, as they tend to favor the majority class, overlooking the rare instances of fraud. Moreover, the dataset for fraud detection typically includes high-dimensional features, from transaction details to customer demographics and device information, adding complexity to the problem.

To address these challenges, this project presents a hybrid approach that combines anomaly detection with supervised learning. Anomaly detection models, like Isolation Forest and One-Class SVM, help filter out potential outliers, effectively reducing noise and enhancing the data quality for the supervised classifiers. The filtered data is then processed through models such as Random Forest, Naive Bayes, and Decision Tree to predict fraud.

For this study, we used the IEEE-CIS Fraud Detection Dataset from Kaggle, which consists of around 590,000 transactions with over 20 features. We pre-processed the data by handling missing values and applied SMOTE (Synthetic Minority Over-sampling Technique) to counteract the class imbalance. Principal Component Analysis (PCA) and other feature engineering techniques helped reduce dimensionality, retaining only the most relevant information for fraud detection.

Our evaluation compares several model combinations, including Isolation Forest + Random Forest and Isolation Forest + Naive Bayes. We assess performance based on accuracy, precision, recall, F1 score, and AUC-ROC, aiming to find the combination that delivers the best balance across metrics. Initial results show that our hybrid approach significantly improves detection performance, especially in catching rare fraudulent transactions, making it a promising solution for real-world applications in fraud detection.

2 Related Work

Fraud detection in online transactions has been extensively studied using various machine learning approaches. Recent

research has explored both traditional algorithms and hybrid approaches to improve detection accuracy while handling inherent challenges like class imbalance.

Several studies have evaluated classical machine learning algorithms. Nayak et al. [1] (2021) compared Naive Bayesian, Support Vector Machine, and Random Forest algorithms, with Naive Bayesian achieving the highest accuracy (97.66%). Building on this, Farouk et al. [2] (2024) expanded the comparison to twelve algorithms, finding Gradient Boosting (96.7%) and Naive Bayes (96.2%) most effective, while introducing additional performance metrics beyond accuracy. A significant challenge in fraud detection is class imbalance. Tang et al. [3] (2024) addressed this through their "Decouple then Combine" framework, separately learning representations for fraudulent and non-fraudulent transactions. Similarly, Kewei et al. [4] (2021) developed a hybrid deep learning model combining Binary Cross Entropy and Focal Loss, demonstrating superior performance on the IEEE-CIS fraud dataset.

The literature suggests that leveraging hybrid and ensemble approaches, such as combining anomaly detection with robust classifiers like XGBoost, offers a promising pathway for improving fraud detection accuracy and generalizability. These approaches, coupled with advanced performance metrics and effective data preprocessing techniques, provide a foundation for developing scalable and efficient solutions for online transaction fraud detection.

3 Dataset and Methods

Dataset Description

For this project, we are using the IEEE-CIS Fraud Detection Dataset [5] from Kaggle. Because it accurately depicts e-commerce transaction data, this dataset is frequently utilized in studies on fraud detection. With more than 20 characteristics, such as transactional details, client demographics, and device data, it has about 590,000 transaction records. The 'isFraud' feature is the dataset's target variable; transactions with a label of 1 indicate fraudulent situations, whereas transactions with a label of 0 indicate genuine transactions.

[31]:

	TransactionID	id_01	id_02	id_03	id_04	id_05	id_06	id_07	id_08	id_09	...	id_31	id_32	id_33	id_34	id_35	id_36	id_37	id_38
0	2987004	0.0	70787.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	samsung browser 6.2	32.0	2220x1080	match_status:2	T	F	T	T
1	2987008	-5.0	98945.0	NaN	NaN	0.0	-5.0	NaN	NaN	NaN	...	mobile safari 11.0	32.0	1334x750	match_status:1	T	F	F	T
2	2987010	-5.0	191631.0	0.0	0.0	0.0	0.0	NaN	NaN	0.0	...	chrome 62.0	NaN	NaN	NaN	F	F	T	T
3	2987011	-5.0	221832.0	NaN	NaN	0.0	-6.0	NaN	NaN	NaN	...	chrome 62.0	NaN	NaN	NaN	F	F	T	T
4	2987016	0.0	7460.0	0.0	0.0	1.0	0.0	NaN	NaN	0.0	...	chrome 62.0	24.0	1280x800	match_status:2	T	F	T	T
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
144228	3577521	-15.0	145955.0	0.0	0.0	0.0	0.0	NaN	NaN	0.0	...	chrome 66.0 for android	NaN	NaN	NaN	F	F	T	F
144229	3577526	-5.0	172059.0	NaN	NaN	1.0	-5.0	NaN	NaN	NaN	...	chrome 55.0 for android	32.0	855x480	match_status:2	T	F	T	F
144230	3577529	-20.0	632381.0	NaN	NaN	-1.0	-36.0	NaN	NaN	NaN	...	chrome 65.0 for android	NaN	NaN	NaN	F	F	T	F
144231	3577531	-5.0	55528.0	0.0	0.0	0.0	-7.0	NaN	NaN	0.0	...	chrome 66.0	24.0	2560x1600	match_status:2	T	F	T	F
144232	3577534	-45.0	339406.0	NaN	NaN	-10.0	-100.0	NaN	NaN	NaN	...	chrome 66.0 for android	NaN	NaN	NaN	F	F	T	F

144233 rows × 41 columns

Figure 1. train_identity Dataset

[30]:

	TransactionID	isFraud	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	card4	card5	...	V330	V331	V332	V333	V334	V335	V336
0	2987000	0	86400	68.50	W	13926	NaN	150.0	discover	142.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2987001	0	86401	29.00	W	2755	404.0	150.0	mastercard	102.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2987002	0	86469	59.00	W	4663	490.0	150.0	visa	166.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	2987003	0	86499	50.00	W	18132	567.0	150.0	mastercard	117.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	2987004	0	86506	50.00	H	4497	514.0	150.0	mastercard	102.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
590535	3577535	0	15811047	49.00	W	6550	NaN	150.0	visa	226.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
590536	3577536	0	15811049	39.50	W	10444	225.0	150.0	mastercard	224.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
590537	3577537	0	15811079	30.95	W	12037	595.0	150.0	mastercard	224.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
590538	3577538	0	15811088	117.00	W	7826	481.0	150.0	mastercard	224.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
590539	3577539	0	15811131	279.95	W	15066	170.0	150.0	mastercard	102.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN

590540 rows × 394 columns

Figure 2. train_transaction Dataset

Class Distribution

This dataset's notable class imbalance—roughly 96.5% of transactions are not fraudulent, while only 3.5% are—is one of its main problems. Because machine learning algorithms may be skewed toward forecasting the majority (non-fraudulent) class, this discrepancy poses a special issue.

Dataset Splitting

The following action is taken to get the data ready for modeling:

1. **Data splitting**: To guarantee that the models are trained efficiently and evaluated reliably, the dataset is split into 80% training data and 20% test data. The split aids in testing the model with unknown data to gauge the accuracy of generalization.

Data Preprocessing

Managing Missing Values: If not managed appropriately, the dataset's numerous missing value characteristics can impair model performance. We use methods like:

1. **Imputation**: The mean or median value can be used to fill in missing values for numerical columns, while the most common value or a distinct category can be used to fill in categorical features.
2. **Removal**: We might think about removing features that don't significantly provide value or that have many missing values.

[10]:

	TransactionID_x	isFraud	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	card4	card5	...	id_31	id_32	id_33	id_34	id_35
0	2987000	0	86400	68.5	W	13926	321.0	150.0	discover	142.0	...	samsung browser 6.2	32.0	2220x1080	match_status:2	T
1	2987001	0	86401	29.0	W	2755	404.0	150.0	mastercard	102.0	...	mobile safari 11.0	32.0	1334x750	match_status:1	T
2	2987002	0	86469	59.0	W	4663	490.0	150.0	visa	166.0	...	chrome 62.0	24.0	1920x1080	match_status:2	F
3	2987003	0	86499	50.0	W	18132	567.0	150.0	mastercard	117.0	...	chrome 62.0	24.0	1920x1080	match_status:2	F
4	2987004	0	86506	50.0	H	4497	514.0	150.0	mastercard	102.0	...	chrome 62.0	24.0	1280x800	match_status:2	T

5 rows × 435 columns

Figure 3. Replacing null values with mode

Resampling using SMOTE: Considering the class imbalance, we will oversample the minority class (fraudulent transactions) using the Synthetic Minority Over-sampling Technique (SMOTE). To assist the algorithm in learning patterns from the minority class without overfitting to redundant data, SMOTE creates synthetic instances.

Feature Selection and Dimensionality Reduction: Principal Component Analysis (PCA) and correlation analysis are used to keep only the most pertinent features to minimize training time

and model complexity. Additionally, feature selection lessens overfitting by eliminating unnecessary or redundant data.

Data Formatting and Standardization: When merging features from several domains (transaction, customer, and device information), it is crucial to make sure that the data formatting is consistent. Support Vector Machines and other models that are sensitive to feature scaling benefit from the standardization of features (z-score normalization), which brings all features to a scale.

Feature engineering: To improve the dataset's capacity for prediction, new features might be developed. To further understand user behavior, we can compute average transaction amounts, transaction frequency, or recent high-value transactions.

4 Model Training (TBD)

This project's main goal is to create a hybrid strategy that blends supervised classification and anomaly detection. This technique makes use of the advantages of both approaches: supervised learning is used to classify the dataset, and anomaly detection is used to weed out possible outliers.

Anomaly Detection: To find outliers (probably fraudulent cases), we first use anomaly detection methods. To lower the noise for the trained classifier that follows, this step pre-filters the data by eliminating outliers and keeping only inliers. At this stage, One-Class SVM and Isolation Forest are mostly utilized:

1. **Isolation Forest**: By creating random trees, this technique separates outliers, making it useful for identifying anomalies.
2. **One-Class SVM**: This approach flags the outliers as possible fraud cases and uses a high-dimensional boundary to identify the typical examples.

Supervised Classifiers: The fine-tuned dataset is fed via supervised classifiers following filtering, which make final predictions regarding the validity of the transaction. Because of their effectiveness and interpretability, we chose the following classifiers:

1. **Random Forest**: A reliable ensemble technique that requires little preprocessing to handle high-dimensional data.
2. **Naïve Bayes**: This model is a straightforward probabilistic model that works well for quick classification, particularly when dealing with structured categorical data.
3. **Decision Tree**: An easy-to-understand model that can manage intricate relationships between variables and is ideal for diverse data types.

Hybrid Model Combinations: Several combinations of anomaly detection and classification algorithms are tested to determine the most effective pairing for fraud detection. These include:

1. **Isolation Forest + Random Forest**: Isolation Forest isolates outliers, while Random Forest classifies the refined data.
2. **Isolation Forest + Naive Bayes**: A combination to examine how anomaly detection pre-filtering impacts the probabilistic Naive Bayes classifier.
3. **One-Class SVM + Decision Tree**: One-Class SVM removes outliers, allowing Decision Tree to classify data without noise.

Implementation Tools: TensorFlow handles any deep learning components (such as autoencoders), XGBoost handles sophisticated boosting methods, and scikit-learn handles model construction and assessment. Matplotlib and Seaborn are used for visualization, while Pandas and NumPy help with data manipulation.

```
# Step 1: Isolation Forest for Anomaly Detection (Unsupervised Learning)
# Initialize the Isolation Forest model for anomaly detection
isolation_forest = IsolationForest(contamination=0.035, random_state=42)

# Fit the Isolation Forest model to the data (assuming 'X' is the full feature set and 'y' the labels)
outliers = isolation_forest.fit_predict(X)  # Outliers will have label -1, inliers as 1

# Step 2: Filter based on Isolation Forest predictions and prepare the dataset for Random Forest
# Use only data points classified as inliers (value 1)
inlier_data = X[outliers == 1]
inlier_labels = y[outliers == 1]

# Splitting the inliers dataset for Random Forest (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(inlier_data, inlier_labels, test_size=0.2, random_state=42)

# Step 3: Balancing the Training Data with SMOTE (Optional, based on class imbalance)
smote = SMOTE(sampling_strategy=0.5, random_state=42)
X_train, y_train = smote.fit_resample(X_train, y_train)

# Step 4: Train the Random Forest Classifier (Supervised Learning)
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Step 5: Model Evaluation
# Making predictions on the test set
y_pred = model.predict(X_test)
y_pred_proba = model.predict_proba(X_test)[:, 1]  # Probabilities for the positive class
```

Figure 4. Implementation of Isolation Forest + Random Forest

```
# 1. Anomaly Detection with Isolation Forest
iso_forest = IsolationForest(contamination=0.035, random_state=42)  # contamination based on the class imbalance (3.5% frau
y_train_iso = iso_forest.fit_predict(X_train)  # Fit the model on the training data

# Map -1 (anomalous) to 1 (fraudulent), and 1 (normal) to 0 (non-fraudulent)
y_train_iso = [1 if i == -1 else 0 for i in y_train_iso]

# 2. Apply Naive Bayes to the pre-filtered data
naive_bayes = GaussianNB()
naive_bayes.fit(X_train, y_train_iso)  # Fit Naive Bayes using the anomaly-filtered data

# 3. Predictions
y_pred = naive_bayes.predict(X_test)  # Make predictions
y_pred_proba = naive_bayes.predict_proba(X_test)[:, 1]  # Probabilities for the positive class (fraudulent transactions)
```

Figure 5. Implementation of Isolation Forest + Naïve Bayes

6 Results and Evaluation Metrics (TBD)

The initial results of the assessment of two hybrid combination models applied to the fraud detection problem are presented in this progress report. Isolation Forest + Random Forest Classifier and Isolation Forest + Naive Bayes Classifier are the two models that have been put into practice and evaluated. These models seek to improve the detection of fraudulent transactions by combining anomaly detection and classification methods. To give a comprehensive picture of their efficacy, both models have been assessed using a range of performance indicators, such as accuracy, precision, recall, F1-score, and AUC-ROC. Two more hybrid vehicles have not yet been tested; the final report will include their findings as well as a thorough assessment. The performance results and insights obtained from the models finished thus far are highlighted in the sections that follow.

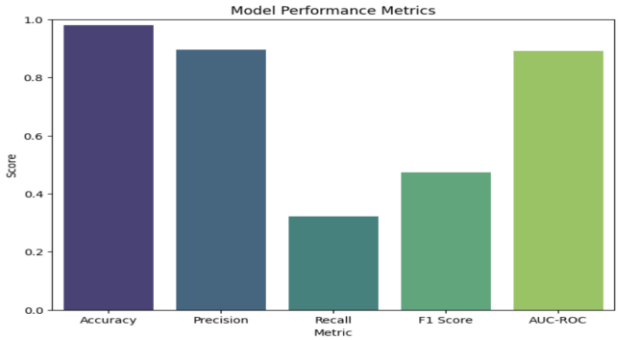

Figure 6. Metrics Result for Isolation Forest + Random Forest

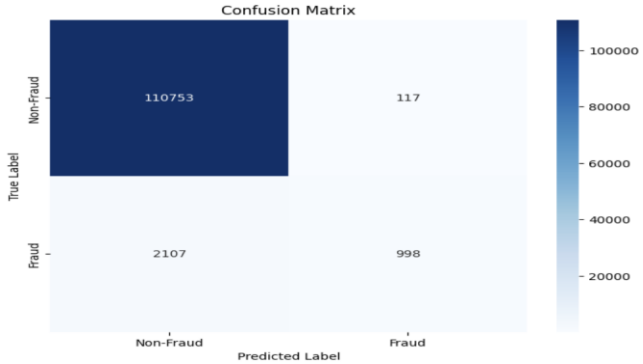

Figure 7. Confusion Matrix for Isolation Forest + Random Forest

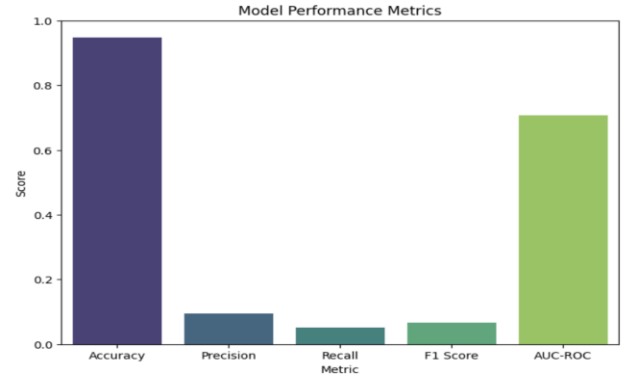

Figure 8. Metrics Result for Isolation Forest + Naïve Bayes

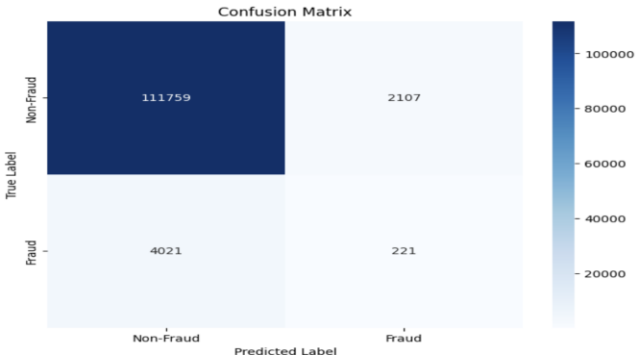

Figure 9. Confusion Matrix for Isolation Forest + Naïve Bayes

7 Summary and Discussion (TBD)

8 Conclusion (TBD)

REFERENCES

[1] Nayak, H.D., Deekshita, Anvitha, L., Shetty, A., D'Souza, D.J., Abraham, M.P. 2021. Fraud Detection in Online Transactions Using Machine Learning Approaches—A Review. In: Chiplunkar, N.N., Fukao, T. (eds) Advances in Artificial Intelligence and Data Engineering. AIDE 2019. Advances in Intelligent Systems and Computing, vol 1133. Springer, Singapore. https://doi.org/10.1007/978-981-15-3514-7_45.

[2] Farouk, M., Maged, R., Ragab, N., Salama, D., Elrashidy, O., Ghorab, N., Hany, J., Amr, A., Adel, O., Saad, K., Ali, K., and Elazab, R. 2024. Fraud_Detection_ML: Machine Learning Based on Online Payment Fraud Detection. Journal of Computing and Communication, 3, 16-131. https://doi.org/10.21608/jocc.2024.339929.

[3] Tang, P., Tang, H., Wang, W., Su, H. and Liu, Y., 2024, February. Decouple then Combine: A Simple and Effective Framework for Fraud Transaction Detection. In Asian Conference on Machine Learning (pp. 1353-1368). PMLR. https://proceedings.mlr.press/v222/tang24a.html.

[4] X. Kewei, B. Peng, Y. Jiang and T. Lu, "A Hybrid Deep Learning Model For Online Fraud Detection," 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 2021, pp. 431-434, doi: 10.1109/ICCECE51280.2021.9342110.

[5] IEEE Computational Intelligence Society, "IEEE-CIS Fraud Detection Dataset," Kaggle, 2019. [Online]. Available: https://www.kaggle.com/competitions/ieee-fraud-detection