

Banking Intrusion Detection System: A Random Forest Approach to Fraudulent Transaction Classification

An evaluation on a class-imbalanced transaction dataset

Mahaswetha K¹, Gopika T², Kaasini M³, Bhuvan Kruthik⁴, Dhanya K R⁵

[#] Student, I Year Department of Computer Science with Artificial Intelligence and Data Science, KPR College of Arts, Science and Research, Coimbatore, Tamil Nadu, India

Assistant Professor, Department of Computer Science with Artificial Intelligence and Data Science, KPR College of Arts, Science and Research, Coimbatore, Tamil Nadu, India

mahaswethakarhikeyan@gmail.com, gopika040608@gmail.com, Kaasinibangar@gmail.com, Bhuvankruthik72017@gmail.com, dhanyafab@gmail.com

Abstract

This paper reports a machine learning study on detecting fraudulent banking transactions, carried out as an undergraduate project named Banking Intrusion Detection System. The dataset contains 1,000,000 transactions described by eight columns, of which 29,999 transactions (about 3 percent) are labelled as fraud. Timestamps were converted to date features, the merchant category was one-hot encoded, identifier columns were removed, and the data were split into 80 percent training and 20 percent test sets with stratification. A single Random Forest classifier with 100 trees and balanced class weights was trained and evaluated on 200,000 held-out transactions. The model reached an accuracy of 0.9457, but this is lower than the 0.97 obtained by labelling every transaction as non-fraud. For the fraud class, precision was 0.0293, recall was 0.0252 and the ROC-AUC was 0.4970, which is no better than random ranking. Lowering the decision threshold raised recall to at most 0.0713 in the tested range while accuracy fell to about 0.90 and precision stayed near the fraud share of the data. Supplementary checks show that the fraud rate is almost constant across merchant categories, months, weekdays and amount ranges. The study therefore does not demonstrate a usable detector. It documents a reproducible baseline, its limitations, and directions for improvement.

Keywords—fraud detection; banking transactions; Random Forest; class imbalance; machine learning; precision and recall

I. INTRODUCTION

Banks process very large numbers of electronic transactions, and a small share of them are fraudulent. Reviews of data mining in financial fraud detection show that classification methods have been widely applied to this problem and that fraud occurs in several forms, including bank fraud [1]. In this project, the term intrusion detection refers to identifying suspicious or fraudulent banking transactions with machine learning, as stated in the project goal. It does not refer to network intrusion detection.

The aim of this paper is to describe, as accurately as possible, what was implemented in the project notebook, what results were obtained, and what those results do and do not show. The paper reports the actual outcome of the experiment, including the finding that the trained model does not separate fraudulent from legitimate transactions better than chance.

II. RELATED WORK

Ngai et al. reviewed 49 journal articles published between 1997 and 2008 on data mining for financial fraud detection. They grouped the fraud into four categories (bank fraud, insurance fraud, securities and commodities fraud, and other related financial fraud) and the

techniques into six classes (classification, regression, clustering, prediction, outlier detection and visualization). They reported that logistic models, neural networks, Bayesian belief networks and decision trees were the main techniques used [1].

Bhattacharyya et al. compared support vector machines and random forests with logistic regression for credit card fraud detection, using real-life transaction data from an international credit card operation [2]. Random forests are therefore an established candidate for transaction fraud classification, which motivates their use as the first model in this project.

III. PROBLEM STATEMENT

Given the recorded attributes of a banking transaction, the task is to predict whether the transaction is fraudulent (label 1) or legitimate (label 0). This is a binary classification problem in which the positive class is rare: about 3 percent of the transactions in the dataset are fraudulent.

IV. OBJECTIVES

- To prepare a labelled banking transaction dataset for supervised machine learning.

- To train a Random Forest classifier that predicts the fraud label.
- To evaluate the classifier on unseen transactions using a confusion matrix, precision, recall, F1-score, accuracy and ROC-AUC.
- To examine how changing the decision threshold affects these measures.

V. PROPOSED METHODOLOGY

The implemented methodology is a standard offline supervised learning workflow (Fig. 1). The dataset was inspected, the timestamp was converted into calendar features, the merchant category was one-hot encoded, and identifier columns were removed. The data were split once into training and test sets, a Random Forest [3] was trained on the training set, and its predictions on the test set were evaluated. The Random Forest implementation of scikit-learn [4] was used. The trained model and its feature list were saved to files at the end of the notebook.

VI. DATASET DESCRIPTION

The dataset (transactions file) contains 1,000,000 rows and 8 columns. There are no missing values and no duplicate rows. All transactions fall in the calendar year 2023, from 1 January to 31 December, and the timestamps carry a +02:00 time offset. The transactions belong to 60,000 distinct customer identifiers. Table I lists the columns and how each was used in the notebook.

TABLE I
DATASET ATTRIBUTES AND THEIR USE IN THE NOTEBOOK

Attribute	Type	Use in the model
transaction_id	integer	Dropped
customer_id	integer	Dropped
transaction_timestamp	text, parsed to date-time	Used to derive date features, then dropped
amount	decimal	Used as a feature
merchant_category	text (5 categories)	One-hot encoded
is_fraud	integer (0 or 1)	Target label
year	integer	Used as a feature
month	integer	Used as a feature

The five merchant categories are electronics, fashion, gaming, groceries and travel.

The amount column ranges from 0.000420 to 999.998865, with a mean of 499.83, a median of 499.89

and a standard deviation of 288.82. The label is highly imbalanced: 970,001 transactions are legitimate and 29,999 are fraudulent (2.9999 percent).

VII. DATA PREPROCESSING

The notebook performed the following steps. (1) The transaction_timestamp column, stored as text, was converted to a date-time type with errors set to produce missing values; no value failed to convert. (2) The number of missing values and duplicate rows was checked and both were zero. (3) The columns transaction_timestamp, transaction_id and customer_id were removed from the modelling data. (4) The merchant_category column was converted to five indicator columns by one-hot encoding. (5) The remaining columns were separated into a feature matrix X and the target y (is_fraud). No scaling, normalisation, outlier removal, imputation or resampling was applied.

VIII. FEATURE ENGINEERING

No feature selection method was used. Four calendar features were derived from the timestamp: transaction hour, day of month, month and day of week. Together with amount, year, month and the five merchant category indicators, this gives 12 model features: amount, year, month, transaction_hour, transaction_day, transaction_month, transaction_dayofweek, and the indicators for electronics, fashion, gaming, groceries and travel.

Two points about these features are worth stating. The month column from the dataset and the derived transaction_month column contain the same information. In addition, supplementary checks on the dataset file show that every timestamp is at midnight (the hour is always 0) and that the year is always 2023, so transaction_hour and year are constant and carry no information. These checks were made while preparing this paper and are not part of the notebook.

IX. MACHINE LEARNING MODEL

A single model was trained: a Random Forest classifier [3] from scikit-learn [4]. A Random Forest combines many decision trees, each trained on a random sample of the data, and averages their votes. The settings used were 100 trees (n_estimators = 100), balanced class weights (class_weight = balanced), random_state = 42 and parallel training on all available processors (n_jobs = -1). All other settings were left at their defaults. No hyperparameter search and no other algorithm were tried in the notebook.

X. SYSTEM ARCHITECTURE AND WORKFLOW

The project is implemented as a notebook-based batch workflow on Google Colab, shown in Fig. 1. The notebook saves the trained model (ids_model.pkl, also zipped) and the list of feature columns

(feature_columns.json). It does not contain a prediction interface, a data pipeline for new transactions or any deployment component, so no real-time or production use is claimed.

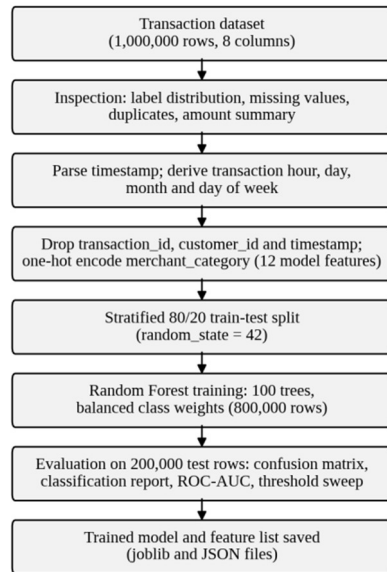


Fig. 1. Workflow implemented in the project notebook.

XI. EXPERIMENTAL SETUP

The experiments were run in Google Colab with Python (the notebook output indicates Python 3.13), pandas and scikit-learn. The dataset was divided once into 80 percent training data (800,000 transactions) and 20 percent test data (200,000 transactions) using a stratified split with `random_state = 42`, so that both parts keep the same fraud share (Table II). The model was trained once and no cross-validation or repeated runs were carried out.

TABLE II
CLASS DISTRIBUTION IN THE FULL DATA AND IN THE TWO SPLITS

Set	Legitimate	Fraud	Fraud share
Full dataset	970,001	29,999	2.99999%
Training set	776,001	23,999	2.99999%
Test set	194,000	6,000	3.00000%

XII. RESULTS AND DISCUSSION

A. Confusion matrix

Table III shows the confusion matrix on the 200,000 test transactions. Of the 6,000 fraudulent transactions, the model identified 151 and missed 5,849. It flagged 5,161 transactions as fraud, of which only 151 were fraudulent and 5,010 were legitimate.

TABLE III
CONFUSION MATRIX ON THE TEST SET (DECISION RULE OF THE PREDICT FUNCTION)

Actual class	Predicted legitimate	Predicted fraud
Legitimate	188,990	5,010
Fraud	5,849	151

B. Classification metrics

Table IV gives the metrics printed by the notebook. The high accuracy is not evidence of good detection, because 97 percent of the test transactions are legitimate: a rule that labels every transaction as legitimate would reach an accuracy of 0.97 (calculated from the test class counts), which is higher than the 0.9457 of the trained model. The fraud-class precision (0.03) is about equal to the fraud share of the test set (0.03), and the ROC-AUC of 0.4970 is close to 0.5, the value expected from random scoring.

TABLE IV
METRICS REPORTED BY THE NOTEBOOK ON THE TEST SET

Class or average	Precision	Recall	F1-score
Legitimate (194,000)	0.97	0.97	0.97
Fraud (6,000)	0.03	0.03	0.03
Macro average	0.50	0.50	0.50
Weighted average	0.94	0.95	0.94

Overall accuracy printed by the notebook: 0.95 (0.945705 before rounding). ROC-AUC: 0.4970. Values computed from Table III for the fraud class: precision 0.0293, recall 0.0252, F1-score 0.0271.

C. Effect of the decision threshold

The notebook also converted the predicted fraud probabilities to labels using lower thresholds (Table V and Fig. 2). Recall increased as the threshold fell, from 0.0262 at 0.5 to 0.0638 at 0.1, while accuracy decreased from 0.9451 to 0.9073. Precision remained between 0.0275 and 0.0296, essentially the fraud share of the test set, which means the flagged transactions were not richer in fraud than a random selection. A finer sweep from 0.05 to 0.15 in steps of 0.005 was also run, and the thresholds with accuracy nearest to 0.90 were listed. At thresholds of 0.075 and 0.08, accuracy was 0.8997, precision 0.0287, recall 0.0713 and F1-score 0.0409.

Predictions made with the predict function and with a 0.5 threshold on the probabilities differed slightly (accuracy 0.945705 and 0.945085). The notebook does not investigate this difference.

TABLE V
TEST SET RESULTS AT DIFFERENT DECISION THRESHOLDS
(FRAUD CLASS)

Threshold	Accuracy	Precision	Recall	F1-score
0.5	0.9451	0.0296	0.0262	0.0278
0.4	0.9423	0.0284	0.0278	0.0281
0.3	0.9411	0.0284	0.0290	0.0287
0.2	0.9251	0.0275	0.0437	0.0338
0.1	0.9073	0.0288	0.0638	0.0397
0.075, 0.08	0.8997	0.0287	0.0713	0.0409

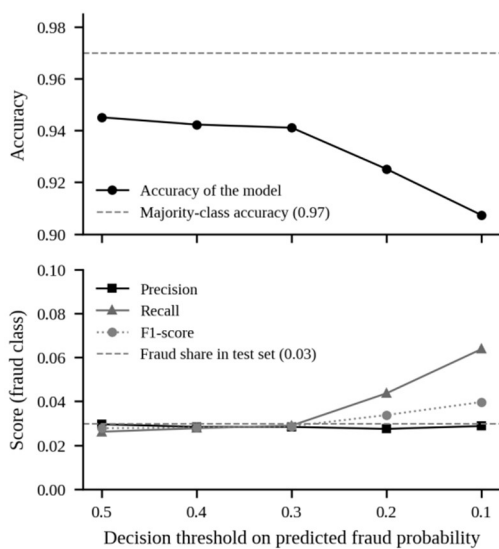


Fig. 2. Accuracy, precision, recall and F1-score at the thresholds in Table V (values from the notebook).

D. Possible explanation

Supplementary descriptive checks on the dataset file (not in the notebook) help to interpret these results. The fraud rate is almost the same in every group of every available feature: between 2.976 and 3.024 percent across the five merchant categories, between 2.902 and 3.167 percent across the twelve months, between 2.937 and 3.080 percent across the seven weekdays, and between 2.958 and 3.067 percent across five equal-width amount ranges. The mean amount is 499.93 for legitimate and 496.43 for fraudulent transactions. When the fraud label barely varies with the available features, a classifier has little to learn from them, which is consistent with the ROC-AUC near 0.5. Whether this is caused by the nature of the data, the way the label was produced, or the absence of more informative attributes cannot be determined from the files provided. These results apply to one model, one split and these features only; they do not show that no model could detect fraud on other data.

XIII. PERFORMANCE EVALUATION

The evaluation used accuracy, precision, recall, F1-score, the confusion matrix and ROC-AUC, all computed on the 200,000-transaction test set with the fraud class as the positive class. Precision is the share of flagged transactions that are truly fraudulent, and recall is the share of fraudulent transactions that are found. Because the positive class is rare, accuracy is dominated by the legitimate class, and precision and recall are more informative; the precision-recall plot has been argued to be more informative than the ROC plot on imbalanced data [5]. The notebook did not compute a precision-recall curve, so no area under that curve is reported. On the measures that matter for the rare class, the model performed at roughly chance level (Tables III to V).

XIV. LIMITATIONS

- The model shows no useful ability to detect fraud on this dataset (fraud-class recall 0.0252, precision 0.0293, ROC-AUC 0.4970).
- Only one algorithm, one train-test split and one run were used. There was no cross-validation, hyperparameter tuning, or comparison with simple baselines such as logistic regression.
- The thresholds were examined on the test set itself, and no separate validation set was used. Any threshold chosen from Table V would therefore give an optimistic estimate.
- The features are limited. The transaction hour and the year are constant, month is duplicated, and customer-level history was not used because customer_id was dropped. The timestamps contain dates only.
- Feature importance and a precision-recall curve were not examined.
- The dataset provenance, the meaning of the label and the amount currency are not documented.
- The workflow is offline. The model was saved but no deployment or real-time testing was done.
- The supplementary checks in Sections VIII and XII were made on the dataset file outside the notebook. They should be added to the notebook so that every reported number can be reproduced from it.

XV. CONCLUSION

A Random Forest classifier was trained on 800,000 banking transactions and tested on 200,000. Its accuracy of 0.9457 is below the 0.97 of a rule that labels all transactions as legitimate, and its fraud-class precision, recall and ROC-AUC are close to the values expected from random guessing. Lowering the decision threshold raised recall to 0.0713 at most, but at the cost of accuracy and without improving precision. The experiment therefore establishes a documented baseline and shows that, with the features in this dataset, the tested model does not detect fraud reliably. The result is a limitation of

this experiment and should not be read as a general finding about machine learning for fraud detection.

XVI. FUTURE SCOPE

- Confirm the origin and labelling of the dataset, and test on data whose attributes plausibly relate to fraud.
- Create customer-level behavioural features, such as the number and amounts of earlier transactions, and time since the previous transaction, using the `customer_id` column.
- Compare several algorithms, including logistic regression and gradient boosting, with cross-validation and a separate validation set for threshold selection.
- Evaluate with precision-recall curves and the area under them [5], and report feature importance.
- Study imbalance-handling methods such as synthetic minority over-sampling [6] and cost-sensitive learning.
- Only after reliable results are obtained, consider a prediction interface and testing on new transactions.

REFERENCES

- [1] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decis. Support Syst.*, vol. 50, no. 3, pp. 559–569, 2011.
- [2] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decis. Support Syst.*, vol. 50, no. 3, pp. 602–613, 2011.
- [3] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [4] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [5] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, p. e0118432, 2015.
- [6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.