Healthcare Provider Fraud Detection System

By: Aisha El Saadany - Habiba Amr - Omar Hatem - Ali Hesham - Abdelrahman Abualhuda

Executive Summary:

Healthcare fraud costs healthcare systems billions of dollars every year. Identifying fraudulent providers manually is slow, expensive, and error-prone.
This project develops an end-to-end machine learning pipeline that detects potentially fraudulent healthcare providers using Medicare claims data. We integrated four large datasets, created provider-level features, handled severe class imbalance (~10%), trained multiple models (Logistic Regression, Random Forest, XGBoost), and selected **XGBoost** as the best model using **Precision-Recall AUC** as the primary metric. Our evaluation and error analysis provide insights into fraud behavior and model weaknesses, helping investigators prioritize high-risk providers.

---

1. Introduction:

Healthcare fraud is a major challenge worldwide. Fraudulent billing practices—such as unnecessary procedures or inflated reimbursements—lead to significant financial losses.
Due to the huge volume of healthcare claims, manual investigation is impossible at scale.

Goal of the Project:
Build a machine learning system that automatically identifies providers with a high likelihood of committing fraud.

Objectives:

- Combine multiple Medicare datasets into a unified structure
- Engineer meaningful provider-level features
- Train multiple ML models
- Address class imbalance
- Evaluate model performance using appropriate metrics
- Analyze model errors (false positives & false negatives)

---

2. Data Description:

The project uses four datasets from the Medicare Fraud Detection Kaggle competition:

1. **Beneficiary Data**
    - Demographics, chronic conditions (e.g., diabetes, heart failure)
2. **Inpatient Claims (IP)**
    - Diagnosis codes, procedure codes, claim durations, reimbursement amounts
3. **Outpatient Claims (OP)**
    - Procedures, reimbursements, deductible amounts
4. **Fraud Labels**
    - Provider labeled as "Yes" (fraud) or "No" (non-fraud)

Key Identifiers:

- BeneID links beneficiary-level data
- Provider is the modeling unit

Target Variable:
PotentialFraud

- 1 = Fraud
- 0 = Not Fraud

---

3. Data Preprocessing

3.1 Cleaning

- Converted date fields to datetime format
- Filled missing numeric fields with 0 (medical billing assumption: no cost reported)
- Filled missing text fields with "Unknown" or "Missing"
- Standardized all categorical values
- Removed duplicate claims

3.2 Label Encoding

Yes/No → 1/0 for the target variable.

3.3 Aggregation

The final unit of analysis is **Provider**, not individual claims.
We aggregated all inpatient, outpatient, and beneficiary information per provider.
Examples of aggregated features:

- **TotalClaimAmount** = sum of all reimbursements
- **NumClaims** = number of claims submitted
- **AvgClaimDuration** = average number of days per claim
- **NumUniqueProcedures**
- **NumUniqueDiagnoses**
- **TotalDeductiblePaid**

After aggregation, the final dataset contained one row per provider.

---

4. Feature Engineering

We created additional features to help the model detect hidden fraud patterns.

4.1 Ratio Features

These capture abnormal financial behavior:

- Reimbursement-to-claim ratio
- Deductible-to-claim ratio
- Claims per beneficiary

4.2 Count Features

- Number of diagnosis codes
- Number of procedure codes
- Unique chronic condition indicators

4.3 Log Transformations

Many financial variables (like reimbursement amounts) were highly skewed.
Applying log-transforms helped normalize them.

4.4 Intensity Features

- Claims per day
- Reimbursement per day

These help identify providers who process unusually large volumes.

---

5. Modeling

This section includes all modeling decisions and trials.

5.1 Algorithms Used

We trained and compared:

- **Logistic Regression** — baseline, simple
- **Random Forest** — handles nonlinearity and interactions
- **XGBoost** — advanced gradient boosting, best for tabular data

5.2 Handling Class Imbalance

Fraud = ~10% of providers → severe imbalance.
We used:

- class_weight='balanced' (LogReg & RF)
- scale_pos_weight (XGBoost)
- Stratified train/test split (to maintain class proportions)

5.3 Hyperparameter Tuning

Used GridSearchCV and RandomizedSearchCV to tune:

- learning rate
- max depth
- number of trees
- min child weight
- regularization parameters

5.4 Best Model Selection

Models were compared using **PR-AUC**, because:

- It focuses on fraud detection performance
- It is appropriate when the positive class is rare

Best Model:

XGBoost

(highest PR-AUC and strongest recall)

---

6. Evaluation:

Evaluation used the saved tuned models from modeling.

6.1 Metrics Used

- Precision (how many flagged providers are actually fraud)
- Recall (how many fraud cases were caught)
- F1-score (balance between precision & recall)
- ROC-AUC
- **PR-AUC** (main metric)

Insert **metrics_df screenshot** here.

6.2 Confusion Matrices

Insert the confusion matrix plot here.

Findings:

- XGBoost: most true positives, fewest false negatives
- Random Forest: balanced but weaker
- Logistic Regression: weakest overall

6.3 ROC Curves

Insert ROC curve plot.

6.4 Precision–Recall Curves

Insert PR curve plot.

Why PR Curve matters:
When fraud is rare, PR-AUC shows how well the model handles the minority class.

---

7. Error Analysis :

We analyzed misclassified cases using the best model (XGBoost).

7.1 False Positives (FP)

Providers wrongly flagged as fraud.

Common characteristics:

- Extremely high reimbursement totals
- High deductible ratios
- Many claims per beneficiary

These providers *look similar to fraud* but are legitimate, causing the model to over-flag them.

7.2 False Negatives (FN)

Fraud providers predicted as non-fraud.

Common patterns:

- Claims values look “average”
- No extreme spikes in reimbursement
- Behave similarly to normal providers

These are subtle and harder to detect.

7.3 Implications

- False positives → extra investigation work
- False negatives → financial loss risk

---

8. Discussion & Insights

What We Learned

- Fraud can often be identified by extreme financial patterns
- Subtle fraud requires more advanced features
- XGBoost performs best on imbalanced tabular healthcare data

Model Limitations

- No temporal features (fraud can change over time)
- Provider-level aggregation hides individual claim behavior
- Diagnosis and procedure codes treated as simple categories

---

9. Conclusion

This project successfully built a complete fraud detection pipeline.
XGBoost was the best-performing model for identifying fraudulent healthcare providers.

Future Improvements:

- Add time-based features
- Use embeddings for diagnosis/procedure codes
- Tune decision thresholds
- Add anomaly detection as a second layer
