HEALTHCARE FRAUD DETECTION SYSTEM

A Machine Learning Based Approach for Detecting Fraudulent Healthcare Claims

PROJECT REPORT

Date: December 15, 2025

Generated on 15-12-2025

TABLE OF CONTENTS

1. ABSTRACT

Healthcare fraud is a significant challenge in the medical insurance industry, resulting in billions of dollars in losses annually. This project presents a comprehensive Healthcare Fraud Detection System that leverages Machine Learning techniques to identify potentially fraudulent healthcare claims in real-time.

The system employs a two-layer detection approach: (1) Rule-based detection for obvious fraud patterns, and (2) A Gradient Boosting Classifier trained on provider-level aggregated features. The model achieves 94.82% accuracy with a ROC-AUC score of 0.9683.

Key features include disease-specific pricing comparison, provider type benchmarking, GST calculation, and a modern web-based dashboard for real-time claim analysis. The system processes claims instantly and provides detailed risk assessments with explanations.

This project demonstrates the practical application of machine learning in healthcare fraud detection, providing a scalable solution that can be adapted for various healthcare systems.

Keywords

Healthcare Fraud, Machine Learning, Gradient Boosting, Fraud Detection, Claims Analysis, ICD Codes, Real-time Prediction

2. PROBLEM STATEMENT

2.1 Background

Healthcare fraud is one of the most significant challenges facing the healthcare industry worldwide. According to the National Health Care Anti-Fraud Association (NHCAA), healthcare fraud costs the US healthcare system approximately $68 billion to $230 billion annually. In India, with the growing adoption of insurance schemes like Ayushman Bharat, fraud detection has become increasingly critical.

Healthcare fraud takes many forms, including:

- Billing for services not rendered
- Upcoding - Billing for more expensive procedures than performed
- Unbundling - Billing separately for services that should be billed together
- Duplicate billing - Submitting multiple claims for the same service
- Phantom billing - Billing for non-existent patients
- Kickbacks - Receiving payments for patient referrals

2.2 T

Traditional fraud detection methods rely heavily on manual auditing, which is:

1. Time-consuming: Manual review of millions of claims is impractical
2. Expensive: Requires large teams of trained auditors
3. Reactive: Fraud is often detected months after occurrence
4. Inconsistent: Human reviewers may miss subtle patterns

The challenge is to develop an automated, real-time system that can:
- Process claims instantly as they are submitted
- Identify suspicious patterns with high accuracy
- Minimize false positives (legitimate claims flagged as fraud)
- Provide explainable decisions for auditor review

2.3 Problem Definition

Given a healthcare claim with attributes such as provider information, diagnosis codes, claim amount, patient demographics, and medical procedures, the system must:

1. Predict whether the claim is potentially fraudulent (binary classification)
2. Provide a fraud probability score (0-100%)
3. Classify the risk level (Low, Medium, High, Critical)
4. Compare the claim amount against expected costs for the specific disease
5. Identify which specific rules or patterns triggered the fraud flag
6. Present results through an intuitive web interface

3. PROJECT OBJECTIVES

3.1 Primary Objectives

- Develop a machine learning model to detect healthcare fraud with >90% accuracy
- Create a real-time web-based dashboard for claim analysis
- Implement disease-specific pricing comparisons using ICD-9 and ICD-10 codes
- Build a scalable backend API for integration with existing systems

3.2 Secondary Objectives

- Implement rule-based fraud detection for obvious patterns
- Calculate GST and total billing amounts
- Support multiple provider types (Government, Clinic, Private)
- Provide detailed explanations for fraud predictions
- Generate comprehensive reports for audit purposes

3.3 Scope

The project scope includes:
- Training on Medicare claims data (558,211 claims from 5,410 providers)
- Supporting ICD-9 and ICD-10 diagnosis code lookup
- Real-time prediction API with <100ms response time
- Web dashboard with statistics, analytics, and claim analysis
- Batch processing capability for historical claims

Out of scope:
- Integration with live insurance systems (demo environment only)
- Patient identity verification
- Provider credential verification

4. LITERATURE REVIEW

4.1 Traditional Approaches

Early fraud detection systems relied on rule-based approaches and expert systems. These systems used predefined rules based on domain knowledge, such as:
- Claims exceeding certain thresholds
- Unusual billing patterns
- Known fraudulent provider characteristics

While effective for known fraud patterns, these systems struggle with novel fraud schemes and require constant manual updates.

4.2 Machine Learning Approaches

Modern fraud detection leverages various ML techniques:

Supervised Learning: Random Forest, Gradient Boosting, Neural Networks trained on labeled fraud/non-fraud examples. These achieve high accuracy but require labeled training data.

Unsupervised Learning: Clustering and anomaly detection to identify unusual patterns without labeled data. Useful for discovering new fraud schemes.

Deep Learning: LSTM and Transformer models for sequence analysis of claim patterns over time.

4.3 Challenges in Healthcare Fraud Detection

Key challenges identified in literature include:

1. Class Imbalance: Fraudulent claims are rare (<5% typically), causing models to be biased toward predicting non-fraud.

2. Evolving Patterns: Fraudsters constantly adapt their techniques to evade detection.

3. Feature Engineering: Raw claims data requires significant preprocessing and domain expertise.

4. Interpretability: Black-box models are difficult to explain to auditors and regulators.

5. Real-time Processing: Systems must handle high volumes with low latency.

5. SYSTEM ARCHITECTURE

5.1 High-Level Architecture

The system follows a three-tier architecture:

1. Presentation Layer (Frontend)
   - React.js web application
   - Recharts for data visualization
   - Tailwind CSS for styling

2. Application Layer (Backend)
   - FastAPI REST API server
   - ML model inference engine
   - Rule-based fraud detection
   - ICD code lookup service

3. Data Layer
   - SQLite database for claims storage
   - CSV files for disease pricing
   - Pickle file for trained ML model

5.2 Data Flow

```
User Input (Web Form)
       |
       v
Frontend (React.js)
       |
       | POST /predict
       v
Backend API (FastAPI)
       |
       +---> Disease Price Lookup
       +---> Rule-Based Detection
       +---> ML Model Prediction
       +---> GST Calculation
       |
       v
JSON Response
       |
       v
Frontend Display
```

5.3 Component Diagram

Key components and their responsibilities:

1. main.py: FastAPI application, API endpoints, request handling
2. database.py: SQLAlchemy models, database connection
3. icd_lookup.py: ICD-9 and ICD-10 code name lookup
4. train_model.py: ML model training pipeline
5. model.pkl: Serialized trained model and scaler
6. App.jsx: React frontend components
7. disease_prices.csv: Expected prices per diagnosis code

6. TECHNOLOGY STACK

6.1 Backend Technologies

Technology	Version	Purpose
Python	3.10+	Primary programming language
FastAPI	0.100+	REST API framework
SQLAlchemy	2.0+	Database ORM
Pandas	2.0+	Data manipulation
Scikit-learn	1.3+	Machine learning
Joblib	1.3+	Model serialization
Uvicorn	0.23+	ASGI server

6.2 Frontend Technologies

Technology	Version	Purpose
React	18.2+	UI framework
Vite	5.0+	Build tool
Axios	1.5+	HTTP client
Recharts	2.8+	Charts and graphs
Tailwind CSS	3.3+	Styling framework
Lucide React	0.290+	Icon library

6.3 Database

SQLite: Lightweight, file-based database suitable for demonstration and development. Contains 558,211 claims records with provider, patient, and claim information.

For production deployment, PostgreSQL or MySQL would be recommended for better concurrency and scalability.

7. DATASET DESCRIPTION

7.1 Data Source

The dataset is derived from the "Healthcare Provider Fraud Detection Analysis" dataset available on Kaggle, originally sourced from US Medicare claims data (2008-2010).

This dataset was chosen because:

- Large scale: 558,211 claims for statistical significance
- Real-world patterns: Based on actual Medicare billing data
- Labeled data: Contains known fraud/non-fraud labels
- Multiple features: Rich attribute set for ML training

7.2 Dataset Statistics

Attribute	Value
Total Claims	558,211
Unique Providers	5,410
Unique Patients	138,556
Fraudulent Providers	~38%
Inpatient Claims	40,474 (7.3%)
Outpatient Claims	517,737 (92.7%)
Unique Diagnosis Codes	6,016 (with 10+ claims)

7.3 Feature Description

Key features used in the model:

Feature	Type	Description
provider_id	String	Unique provider identifier
diagnosis_code	String	ICD-9 diagnosis code
claim_type	Categorical	Inpatient or Outpatient
amount	Float	Claim amount in currency
num_diagnoses	Integer	Number of diagnosis codes
num_procedures	Integer	Number of procedures
length_of_stay	Integer	Days in hospital
patient_age	Integer	Patient age in years
chronic_conditions	Integer	Number of chronic conditions
is_fraud	Boolean	Fraud label (target)

8. MACHINE LEARNING MODEL

8.1 Model Selection

After evaluating multiple algorithms, Gradient Boosting Classifier was selected as the primary model due to its:

1. High accuracy on tabular data
2. Resistance to overfitting with proper tuning
3. Built-in feature importance ranking
4. Interpretability compared to deep learning
5. Efficient inference time for real-time prediction

8.2 Feature Engineering

A key insight that improved accuracy from 62% to 95% was provider-level aggregation. Instead of training on individual claims, we aggregate claims by provider to create 28 features:

- total_claims: Count of claims per provider
- amount_mean, amount_sum, amount_std: Financial metrics
- claims_per_patient: Average claims per unique patient
- revenue_per_patient: Average revenue per patient
- num_diagnoses_mean: Average diagnoses per claim (upcoding indicator)
- inpatient_ratio: Percentage of inpatient claims

8.3 Model Training

Training Configuration:

```
GradientBoostingClassifier(
    n_estimators=200,      # Number of trees
    max_depth=6,           # Maximum tree depth
    learning_rate=0.1,     # Step size shrinkage
    min_samples_split=10,  # Minimum samples to split
    min_samples_leaf=5,    # Minimum samples in leaf
    subsample=0.8,         # Fraction of samples per tree
    random_state=42        # Reproducibility
)
```

8.4 Model Performance

Metric	Value
Training Accuracy	100.00%
Testing Accuracy	94.82%
ROC-AUC Score	0.9683
Precision (Fraud)	91.4%

Recall (Fraud)	95.8%
F1-Score	93.5%
Cross-Validation Mean	94.14% (+/- 1.43%)

8.5 Confusion Matrix

```
                    Predicted
                  Non-Fraud  Fraud
Actual Non-Fraud    620       38     <- False Positives
       Fraud         18      406     <- True Positives
                     ^
                False Negatives

True Negatives (TN):  620 legitimate correctly identified
False Positives (FP): 38 legitimate wrongly flagged
False Negatives (FN): 18 fraud cases missed
True Positives (TP):  406 fraud correctly caught
```

8.6 Feature Importance

Top 5 most important features for fraud prediction:

1. amount_sum (14.2%) - Total revenue is the strongest indicator
2. revenue_per_patient (11.8%) - High extraction per patient
3. amount_mean (9.5%) - Average claim amount
4. claims_per_patient (8.7%) - Repeat billing patterns
5. total_claims (7.6%) - Volume of claims submitted

9. IMPLEMENTATION DETAILS

9.1 Rule-Based Detection

Before ML prediction, claims are checked against rule-based filters:

Rule 1: Disease-Specific Pricing
- Compare claim amount against expected price for the specific diagnosis
- Flag if amount > 2.5x expected (Suspicious zone)

Rule 2: Excessive Diagnoses (Upcoding)
- Flag claims with more than 15 diagnosis codes

Rule 3: Invalid Patient Age
- Flag if age > 120 or age < 0

Rule 4: Age-Condition Mismatch
- Flag young patients (<30) with many chronic conditions (>5)

Rule 5: Inpatient Without Stay
- Flag inpatient claims with 0 length of stay

9.2 Disease-Specific Pricing

The system maintains expected prices for 6,016 diagnosis codes:

```
Provider Multipliers:
  Government: 0.7x (subsidized)
  Clinic:     1.0x (standard)
  Private:    1.8x (premium)

Price Zones:
  Normal:     <= 1.5x expected
  Elevated:   1.5x to 2.5x expected
  Suspicious: > 2.5x expected
```

9.3 API Endpoints

Endpoint	Method	Description
/stats	GET	Get dataset statistics
/claims	GET	List recent claims
/predict	POST	Analyze a claim for fraud
/health	GET	API health check

9.4 GST Calculation

All claims include 18% GST calculation (Indian healthcare standard):

Base Amount: Original claim amount
GST Amount: Base Amount x 0.18
Total with GST: Base Amount + GST Amount

10. RESULTS AND EVALUATION

10.1 System Performance

Metric	Result
API Response Time	<100ms
Model Load Time	<2 seconds
Claims Processed	558,211
Providers Analyzed	5,410
ICD Codes Supported	162,611 (ICD-9 + ICD-10)

10.2 Test Cases

Comprehensive testing was performed with 10 test cases:

1. ICD-9 Hypertension (4019) - PASSED
2. ICD-9 Acute Respiratory Failure (51881) - PASSED
3. V-Code Heart Assist (V4321) - PASSED
4. ICD-10 COPD (J44.1) - PASSED
5. ICD-10 Fracture (S72001A) - PASSED
6. Government Hospital Low Amount - PASSED
7. Suspicious High Amount Claim - PASSED
8. Young Patient (Age 25) - PASSED
9. Many Chronic Conditions (11) - PASSED
10. Unknown Diagnosis Code - PASSED

All 10 tests passed successfully.

10.3 Key Findings

- Provider-level aggregation significantly improves accuracy (62% to 95%)
- Top fraud indicators: total revenue, revenue per patient, claim volume
- Disease-specific pricing catches overcharging effectively
- Rule-based detection catches obvious fraud patterns before ML analysis

11. SCREENSHOTS

The web dashboard provides an intuitive interface for fraud detection:

1. Dashboard Tab: Shows overall statistics including total claims, fraud percentage, and key metrics with visual charts.

2. Analyze Claim Tab: Interactive form to input claim details and receive instant fraud risk assessment with detailed breakdown.

3. Analytics Tab: Data visualization including fraud distribution pie chart, claim trends, and provider statistics.

4. Dataset Explorer Tab: Browse and search through the claims database with filtering capabilities.

Key UI Features:
- Dark theme for reduced eye strain
- Responsive design for various screen sizes
- Real-time updates without page refresh
- Color-coded risk levels (Green=Low, Yellow=Medium, Orange=High, Red=Critical)
- Detailed tooltips and explanations

12. FUTURE ENHANCEMENTS

12.1 Short-term Improvements

- Add user authentication and role-based access control
- Implement batch upload functionality for multiple claims
- Add email notifications for high-risk claims
- Create PDF export for audit reports

12.2 Medium-term Improvements

- Integrate with live insurance claim systems via APIs
- Add temporal analysis for detecting patterns over time
- Implement provider network analysis for collusion detection
- Add natural language explanation generation using LLMs

12.3 Long-term Vision

- Deep learning models for sequential claim analysis
- Real-time streaming with Apache Kafka
- Graph neural networks for provider relationship analysis
- Federated learning for privacy-preserving model training

13. CONCLUSION

This project successfully demonstrates a comprehensive Healthcare Fraud Detection System that combines rule-based and machine learning approaches to identify potentially fraudulent claims.

Key Achievements:

1. High Accuracy: The Gradient Boosting model achieves 94.82% accuracy with a ROC-AUC of 0.9683, effectively distinguishing between fraudulent and legitimate providers.

2. Real-time Processing: The system provides instant fraud risk assessment with average response times under 100 milliseconds.

3. Explainable Results: Each prediction includes detailed explanations, price zone classifications, and rule violations, making it suitable for auditor review.

4. Comprehensive Coverage: Support for 162,611 ICD codes (both ICD-9 and ICD-10), disease-specific pricing for 6,016 diagnoses, and three provider type benchmarks.

5. Modern Architecture: Clean separation of concerns with FastAPI backend, React frontend, and SQLite database, making it easy to extend and maintain.

The project validates that machine learning can be effectively applied to healthcare fraud detection, providing a scalable foundation that can be adapted for real-world deployment with appropriate data security and regulatory compliance measures.

14. REFERENCES

[1] Kaggle Healthcare Provider Fraud Detection Dataset
https://www.kaggle.com/datasets/rohitrox/healthcare-provider-fraud-detection-analysis

[2] National Health Care Anti-Fraud Association (NHCAA)
https://www.nhcaa.org/

[3] ICD-9-CM Diagnosis Codes
https://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes

[4] ICD-10-CM Diagnosis Codes
https://www.cms.gov/Medicare/Coding/ICD10

[5] Scikit-learn: Machine Learning in Python
https://scikit-learn.org/

[6] FastAPI Framework
https://fastapi.tiangolo.com/

[7] React.js
https://reactjs.org/

[8] Gradient Boosting Classifier
Friedman, J.H. (2001). Greedy function approximation: a gradient boosting machine.