AI-Based Health Insurance Claim Fraud Detection System: A Research Paper

KAVYA SREEKUMAR
Good Shepherd College
of Engineering and Technology

ATHIRA S
Good Shepherd College
of Engineering and Technology

SARAVANAN
Good Shepherd College
of Engineering and Technology

ABSTRACT

The rapid growth of the healthcare insurance industry has led to an increase in fraudulent claims, resulting in significant financial losses. This paper presents an AI-Based Health Insurance Claim Fraud Detection System developed using Python, Django, and machine learning techniques. The system aims to identify fraudulent claims accurately by analyzing patient details, treatment information, and billing patterns.

The fraud detection module utilizes Optical Character Recognition (OCR) to extract text from medical bills and a Logistic Regression model to calculate a fraud probability score. A Smart Audit engine further enhances the system by applying automated business rules, such as policy velocity checks. Results show that the system improves auditing efficiency and reduces manual effort

KEYWORDS

Artificial Intelligence, Machine Learning, Fraud Detection, OCR, Health Insurance, Logistic Regression, Smart Audit.

1. INTRODUCTION

The healthcare insurance industry faces significant challenges due to rising fraudulent medical claims, which cause financial losses and raise premium costs. Traditional auditing relies on manual review processes that are time-consuming and prone to human error. This project introduces a system combining OCR technology and machine learning to automate the initial analysis phase, allowing human auditors to focus on high-risk cases

2. LITERATURE REVIEW

The literature review explores the evolution of artificial intelligence in insurance fraud detection, moving from traditional manual methods to advanced machine learning and deep learning approaches.

2.1 RELATED WORKS

The proposed system is built upon the foundational research of several key studies in the financial anomaly detection domain:

Decision Tree Based Fraud Detection: Kumar et al. (2019) used decision trees to classify claims, finding them highly interpretable for non-technical users but prone to overfitting on unseen data.

Ensemble Approaches (Random Forest & Logistic Regression): Patel et al. (2020) compared multiple techniques and determined that while ensemble methods like Random Forest offer high performance, they require significant computational resources and large labeled datasets.

Support Vector Machines (SVM): Sharma et al. (2021) utilized SVMs to separate fraudulent and non-fraudulent claims by finding optimal decision boundaries in high-dimensional space. However, the training phase was noted to be slow for large datasets.

Deep Learning Models: Wang et al. (2022) introduced neural networks to identify complex non-linear patterns. While highly accurate, this approach proved impractical for many organizations due to the immense data science infrastructure required.

Hybrid Models: Singh et al. (2023) proposed combining multiple algorithms to maximize reliability. Despite superior performance metrics, the complexity of such models makes them difficult to maintain and explain to stakeholders.

2.2 RESEARCH GAPS AND PROPOSED APPROACH

Existing research highlights a significant trade-off between model complexity and practical deployment. High-accuracy deep learning or hybrid models often demand excessive computational power and massive training datasets that are difficult to obtain due to privacy regulations.

The proposed system addresses these gaps by implementing a lightweight Logistic Regression model. This choice provides a computationally efficient solution that, when combined with OCR-based data extraction and automated business rules (Smart Audit), offers a deployable and scalable alternative for modern insurance providers.

3. METHODOLOGY

The methodology of the AI-Based Health Insurance Claim Fraud Detection

System utilizes a systematic, modular approach to bridge the gap between document processing and automated risk assessment. The system architecture is designed to ensure a seamless data flow from initial document upload to the final classification decision.

3.1 SYSTEM ARCHITECTURE

The system is organized into three primary layers to maintain scalability and performance:

- Input Module: This layer allows insurance officers to upload medical bill images (JPG, JPEG, or PNG) via a web-based Django interface.
- Processing Module: This core analytical layer handles OCR extraction, feature engineering, and machine learning inference.
- Output Module: This layer presents the final audit results, including the fraud probability percentage, risk level, and recommended action (Approve, Reject, or Investigate).

3.2 FRAUD DETECTION PIPELINE

The analytical process follows a step-by-step pipeline to transform unstructured images into actionable data:

- Image Acquisition and Preprocessing: Uploaded images are preprocessed using the Pillow library to enhance contrast and reduce noise, which improves OCR accuracy.

- OCR Extraction: The system uses Tesseract or EasyOCR to convert image pixels into machine-readable text strings.
- Feature Engineering: Raw text is parsed using Regular Expressions

  (Regex) to extract specific variables: Patient Age, Claim Amount, Hospital Type, Treatment Duration, and Gender.

- Data Normalization: Categorical variables are label-encoded, and numerical features are scaled using a StandardScaler to prepare them for the model.
- Fraud Scoring: A trained Logistic Regression model processes the normalized features to output a probability score between 0% and 100%.

3.3 SMART AUDIT DECISION ENGINE

To provide a more comprehensive assessment than AI alone, the system integrates a Smart Audit Engine that combines the ML score with predefined business rules:

- Policy Velocity Check: The engine queries the MySQL database to count claims submitted by a policy within the last 365 days. If a policy has three or more claims in a year, it is flagged for immediate rejection.
- Policy Age Check: Claims made within the first 15 days of a new policy are automatically sent for manual investigation to prevent "immediate claim" fraud patterns.

- Decision Matrix:

  Approve: AI Score < 40%, clean history, and policy > 15 days old.

  Investigate: AI Score between 40%–75% or policy < 15 days old.

  Reject: AI Score > 75% or 3rd claim within one year.

3.4 HUMAN-IN-THE-LOOP (HITL)

Recognizing the limitations of automated systems, the methodology includes an Admin Audit Interface. Claims flagged as "Investigate" are routed to a dashboard where human auditors can review the OCR evidence and apply a manual override, ensuring accountability through a permanent audit trail.

Mathematical Representation

The linear combination of input features:

$$z = \beta_0 + \beta_1(\text{Age}) + \beta_2(\text{Amount}) + \beta_3(\text{Hospital Type}) + \beta_4(\text{Duration}) + \ldots$$

The Sigmoid Function maps z to a probability:

$$P(\text{Fraud}) = 1 / (1 + e^{(-z)})$$

Where β (Beta) values represent the model weights learned during training. A higher β for a feature means that feature has a stronger influence on the fraud prediction.

4. IMPLEMENTATION

The implementation phase involves converting the proposed system design into a functional software application. The AI-Based Health Insurance Claim Fraud Detection System is built using Python and the Django web framework, integrating document processing and machine learning workflows.

4.1 DEVELOPMENT ENVIRONMENT

The system leverages a robust stack of open-source technologies to handle web management, data extraction, and predictive modeling:

- Web Framework: Django (Python-based) manages the server logic, user authentication, and the admin panel.
- Database: MySQL stores claim records, policy history, and audit trails.
- OCR Engines: Tesseract or EasyOCR are used to convert medical bill pixels into machine-readable text.
- Machine Learning: Scikit-Learn provides the Logistic Regression implementation and data normalization tools.

4.2 FRAUD DETECTION MODULE

When an insurance officer uploads a medical bill image through the Django frontend, the module initiates a multi-step analysis pipeline:

1. Image Preprocessing: The Pillow library reduces noise and enhances contrast to improve text recognition accuracy.
2. Text Extraction: The OCR engine extracts raw text, which is then logged to the database as evidence for audit review.
3. Data Structuring: Regular Expressions (Regex) scan the raw text for specific patterns matching fields like Age, Claim Amount, and Hospital Type.
4. Inference: Extracted features are passed to a pre-trained Logistic Regression model (stored as a .pkl file) to return a Fraud Probability score.

4.3 SMART AUDIT AND ADMIN INTERFACE

The implementation ensures that AI decisions are supplemented by business logic and human oversight:

- Business Rules Engine: The system queries the MySQL database for "Policy Velocity," checking if the same policy has submitted three or more claims within 365 days.
- Admin Dashboard: Developed using HTML5, CSS3, and JavaScript, the dashboard allows authorized auditors to view claims flagged for investigation.
- Override Mechanism: Administrators can manually approve or reject a claim. This action requires a mandatory justification note, which is recorded in an Audit Trail table for accountability.

4.4 TESTING AND DEBUGGING

To ensure system stability, the application underwent rigorous testing:

- Unit Testing: Each module (OCR, ML model, Business Rules) was tested separately to verify intended functions.
- Integration Testing: The complete pipeline—from image upload to final classification—was verified for connectivity and API response.
- Functional Testing: Scenarios such as policy velocity violations and highrisk fraud indicators were simulated to ensure the correct trigger of "Reject" or "Investigate" statuses.

5. RESULTS AND DISCUSSION

The AI-Based Health Insurance Claim Fraud Detection System was evaluated based on its ability to process medical documents, calculate risk scores, and manage the administrative audit workflow. The results demonstrate that the system effectively integrates document analysis with real-time decision support.

5.1 SYSTEM RESULTS

The system successfully executed its core functions throughout the testing phase:

- Text Extraction: The OCR module accurately converted pixel data from medical bills into machine-readable strings for most standard document formats.
- Risk Classification: The Logistic Regression model correctly categorized claims into Low, Medium, and High risk based on expected probability thresholds.
- Business Rule Enforcement: The Smart Audit engine successfully identified and flagged policy velocity violations, such as three or more claims within a single year.
- Audit Trail Accountability: Every administrative override was documented with a mandatory reason, ensuring a permanent and transparent record of human decisions.

5.2 PERFORMANCE ANALYSIS

The system's performance was analyzed across several technical and usability metrics:

- Processing Speed: The entire pipeline—from document upload to final classification display—completes within seconds for standard-quality images.
- System Stability: The Django-based backend remained stable while handling concurrent users and simultaneous claim submissions.
- User Interface (UI) Clarity: The animated fraud probability gauge and risk-level indicators improved the interpretability of AI findings for nontechnical insurance staff.

5.3 DISCUSSION

The implementation highlights the significant potential for AI to reduce the manual workload in insurance auditing. While the system provides a robust first line of defense, several critical observations were noted during development:

- OCR Dependencies: The accuracy of the fraud score is heavily dependent on the quality of the uploaded document. Low-resolution or handwritten bills resulted in reduced extraction accuracy.

- Human-in-the-Loop Necessity: Because AI systems can make errors, the "Investigate" status and manual override dashboard are essential for maintaining the final decision's accuracy and legal defensibility.
- Scalability: Combining lightweight models with automated business rules allows the system to scale effectively without the high computational costs associated with deep learning.

This system represents a meaningful advancement toward intelligent healthcare financial management by protecting both providers and policyholders from the financial damage caused by fraudulent claims.

REFERENCES

The following references were used during the development and study of the project:

- A. Kumar et al., Insurance Fraud Detection Using Decision Tree Algorithm, International Journal of Computer Applications, Vol. 180, No. 28, pp. 15-19, 2019.
- S. Patel et al., Machine Learning Approaches for Insurance Claim Fraud Detection, Journal of Information Security and Applications, Vol. 52, pp. 102460, 2020.
- R. Sharma et al., Insurance Fraud Detection Using Support Vector Machine, International Journal of Advanced Research in Computer Science, Vol. 12, No. 3, pp. 45-51, 2021.
- L. Wang et al., Deep Learning Based Insurance Fraud Detection System,IEEE Transactions on Neural Networks and Learning Systems, Vol. 33, No. 8, pp. 3218-3229, 2022.
- M. Singh et al., Hybrid Model for Insurance Fraud Detection, Expert Systems with Applications, Vol. 215, pp. 119360, 2023.
- Django Software Foundation. (2024). Django Web Framework Documentation. Available at: https://www.djangoproject.com
- Python Software Foundation. (2024). Python Documentation. Available at: https://www.python.org
- Scikit-Learn Documentation. (2024). Machine Learning in Python. Available at: https://scikit-learn.org
- Tesseract OCR Documentation. (2024). Open Source OCR Engine. Available at: https://github.com/tesseract-ocr
- MySQL Documentation. (2024). MySQL 8.0 Reference Manual. Available at: https://dev.mysql.com/doc
- Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, pp. 2825-2830.
- McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas and NumPy. O'Reilly Media.