AI-Powered Insurance Claim Decision Support and Advisory System

Jyoti Kanjalkar
Computer Science Engineering(Artificial Intelligence and Machine Learning)
Vishwakarma Institute of Technology
Pune, India
jyoti.kanjalkar@vit.edu

Ansh Sharma
Computer Science Engineering(Artificial Intelligence and Machine Learning)
Vishwakarma Institute of Technology
Pune, India
ansh.sharma23@vit.edu

Ninad Bodade
Computer Science Engineering(Artificial Intelligence and Machine Learning)
Vishwakarma Institute of Technology
Pune, India
ninad.bodade23@vit.edu

Shravani Divate
Computer Science Engineering(Artificial Intelligence and Machine Learning)
Vishwakarma Institute of Technology
Pune, India
shravani.divate23@vit.edu

Hitanshi Meshram
Computer Science Engineering(Artificial Intelligence and Machine Learning)
Vishwakarma Institute of Technology
Pune, India
hitanshi.meshram23@vit.edu

Saniya Nirmale
Computer Science Engineering(Artificial Intelligence and Machine Learning)
Vishwakarma Institute of Technology
Pune, India
saniya.nirmale23@vit.edu

***Abstract*—Manual insurance claims processing is often opaque, complex, and inefficient, leading to delays and limited transparency among stakeholders. This paper proposes an end-to-end AI-powered decision support and advisory system to automate insurance document analysis, claim evaluation, fraud detection, and policy recommendations. The system employs NLP, large language models, and machine learning to extract structured information from policy documents and assess claims with anomaly detection, while explainable AI provides transparent reasoning. Additionally, the platform generates risk alerts by benchmarking claim costs against hospital data. By integrating these technologies, the framework streamlines the claims workflow while ensuring decisions remain interpretable. Automated NLP can abstract key policy terms, and ML models enable real-time fraud detection and document verification. Critically, XAI provides human-understandable justifications for claim decisions, aligning with regulatory and user trust requirements. This combination of automation and transparency reduces inefficiencies and mitigates bias, ultimately fostering greater trust among policyholders and regulators.**

***Keywords*— Insurance claims processing, Natural Language Processing, Large Language Models, Machine Learning, Explainable AI, Fraud detection, Decision support, Transparency.**

I. Introduction

Insurance claim processing is traditionally labor-intensive and time-consuming, relying on manual data entry and expert review of policy documents, medical records, and billing information. This often leads to delays, inconsistencies, and customer dissatisfaction. Claimants may wait days or weeks for decisions, and lack visibility into why claims were approved or denied. At the same time, health insurance fraud remains a substantial problem, driving up costs. Recent advances in artificial intelligence (AI) offer opportunities to automate many of these tasks. Natural Language Processing (NLP) and large language models (LLMs) can parse and summarize complex insurance policies and medical reports; machine learning (ML) classifiers can learn from historical data to decide claims; and anomaly-detection algorithms can flag suspicious billing patterns. Importantly, explainable AI (XAI) techniques can reveal the rationale behind automated decisions, increasing transparency and regulatory compliance. Several insurers have begun applying AI: for example, NLP tools can extract key fields from unstructured claims forms, while chatbots handle routine inquiries.

However, much of this remains ad-hoc. A unified end-to-end system is needed to handle document ingestion, claim evaluation, fraud detection, and customer advisories in a cohesive pipeline. In this work, we propose such a system, combining state-of-the-art AI methods. The system's architecture includes modules for document processing, automated decision-making, XAI explanations, fraud screening, and policy recommendation. It leverages hospital cost data for benchmarking reimbursement and uses a Streamlit-based interface for ease of use. Key goals are faster processing, increased accuracy, full traceability of decisions, and user-friendly guidance. We ground our design in prior literature on NLP for document understanding, AI-driven claims processing, XAI, fraud detection, and recommender systems, aiming to fill gaps in transparency and automation in the insurance domain.

II. Literature Review

Flood risk prediction has evolved from manual monitoring systems to advanced computational models integrating AI, IoT, and graph theory.

1) Use of NLP/LLMs in Document Understanding

Insurance involves extensive documentation (policy text, bills, doctor's notes). NLP enables machines to read and structure this information. For example, Hanmante *et al.* note that NLP algorithms can *"abstract policy documents and answer customer questions in real-time,"* effectively "unscrambling insurance policies" to make them understandable. Large language models (LLMs) like BERT or GPT-4 can be fine-tuned or used with retrieval mechanisms to extract entities (e.g., diagnoses, procedure codes, costs) from unstructured text. Hybrid approaches (e.g., Retrieval-Augmented Generation, or RAG) use embeddings and vector search to fetch relevant policy clauses or medical information, then use LLMs to compose answers or summaries. Recent work even uses domain-specific LLMs and benchmarks (e.g., insurance-question answering datasets) to ensure accuracy in this context. In summary, NLP/LLMs are becoming effective at automating the parsing of claims documents and policies, reducing manual review.

2) AI in Insurance Claims Processing:

AI and ML can significantly speed up claims handling. For example, ensemble classifiers and neural networks have been applied to classify claims as approved or denied, detect needed fraud follow-up, and predict settlement amounts. Automation in claims has been shown to cut processing times dramatically. Bhattacharya *et al.* report that AI-driven automation allows *"faster claims settlements and improved customer satisfaction,"* as routine tasks are delegated to machine learners. Similarly, Rajagopal (2020) finds that AI models *"automate claim processes,"* which accelerates resolution and boosts insurer efficiency. Chatbots and virtual assistants are also used to interact with customers, answer coverage questions, and even gather missing information. According to Hanmante *et al.*, insurers currently use AI chatbots for immediate customer responses, and NLP tools to automatically extract key terms from voluminous policy documents. . These AI applications enable near-real-time claim triage and decision support, turning days-long workflows into minutes or hours. However, most implementations remain partly manual, underscoring the need for integrated pipelines.

3) Explainable AI (XAI) for Decision Transparency:

In high-stakes domains like insurance, transparency is crucial. Insurance regulators increasingly demand that automated decisions be interpretable. XAI techniques provide post-hoc explanations of model outputs. For instance, Bora *et al.* (2022) applied SHAP and LIME to interpret insurance premium prediction models, demonstrating how highlighting influential features (e.g., patient age or previous claims) allows domain experts to verify correctness. In practice, this means a user could see which factors (covered procedures, policy limits, deductibles) drove a claim decision. The literature emphasizes that XAI builds trust: Bhattacharya *et al.* note that explainability *"helps check the correctness of prediction models"* and gives users confidence by making the model's reasoning visible. Explainable frameworks are cited as critical in the insurance context to ensure fairness and regulatory compliance, as opposed to opaque "black box" AI. Our system embeds XAI by providing rule-based justifications and feature-importance scores alongside each decision, aligning with these best practices.

4) ML-based Fraud Detection in Health Insurance:

Fraudulent claims cost the industry billions annually. ML and NLP have been applied to detect anomalies and red flags. Cheekaramelli (2025) found that employing NLP models (e.g., BioBERT/ClinicalBERT) on physician notes and claim descriptions *"increases fraud detection accuracy by 30%"* and *"reduces false positives by 20%,"* compared to rule-based systems. Other work integrates AI with blockchain or wearables: Kapadiya *et al.* (2022) use AI + blockchain to flag insurance fraud, thereby *"increasing transparency and trust"* among stakeholders. In general, ensemble ML models (random forests, neural nets) trained on claim history and patient data have outperformed traditional fraud rules. These approaches inspire our fraud module, which uses supervised models to score claim risk and flags outliers (e.g., unusually high procedure costs or unlikely diagnoses), supplementing any rule-based checks. By cross-referencing hospital benchmark prices, the system can also spot inflated claims.

5) Policy Recommendation Systems:

Beyond claims, AI can advise customers on suitable insurance plans. Traditional recommendation systems (e.g., for e-commerce) have been adapted to match client profiles with policy features. Recently, retrieval-augmented generation (RAG) has been explored to retrieve and synthesize policy terms. For example, Hanmante et al. build a policy recommendation engine that uses RAG with synthetic data to suggest policies tailored to user-specified terms. In their system, users input desired coverage criteria, and the RAG model retrieves relevant policy excerpts to generate a recommendation. Other approaches use similarity ranking on structured policy databases. Our review found limited published work specifically on insurance policy recommenders, but RAG and transformer-based methods are emerging as promising techniques for generating explanations along with suggestions. We adopt a similar retrieval-based method: user requirements are vectorized and matched against a policy corpus, yielding the top policies with reasoning.

In summary, prior work across these areas suggests the feasibility and potential benefits of an integrated AI system for insurance claims. We draw on NLP for document parsing, automated decision-making, XAI for transparency, ML for fraud detection, and retrieval-based recommendation for policies. Our system synthesizes these advances into a cohesive pipeline.

III. Proposed System

A. System Architecture

The proposed system follows a modular, end-to-end pipeline. At a high level, Figure 1 (conceptual) shows that user inputs (claim forms and related documents) enter the **Document Processing** module. Extracted features feed into the **Decision Engine**, which produces an approval/denial and reimbursement amount. Concurrently, a **Fraud Detection** module assigns a risk score to the claim. The **Explainability** component intercepts the decision output to generate a human-readable explanation. The **Policy Recommendation** module can be triggered as needed (e.g. after a denial or for future planning). All user data and documents are managed in the **User Document Management** subsystem. Underlying these modules is a **Knowledge Base** containing policy texts, medical price data, and historical claims, enabling context and benchmarking. This architecture is inspired by recent multi-module AI systems in insurance support, which unify chat, document retrieval, and policy recommendation via LLMs. In our case, each module exposes a well-defined API so they can be developed and scaled independently.

B. Workflow of Modules:

Each core module operates as follows:

a) Document Processing

Incoming claim documents (e.g. PDF bills, doctor notes, images of receipts) are first run through Optical Character Recognition (OCR) (e.g. Tesseract) to obtain text. NLP pipelines then clean and parse this text. Key-value extraction and Named Entity Recognition identify relevant fields: patient name, provider, ICD or CPT codes, service dates, billed amounts, and policy identifiers. If unstructured notes contain additional context (e.g. procedure description), an LLM can be queried (with prompts or RAG) to classify or summarize that information. The output is a structured representation of the claim. For example, the module might extract:

- *Patient Age:* 45; *Policy Limit:* 1,00,000 Rs.
- *Procedure:* MRI scan of knee; *Billed Amount:* $5,000.

  These features are passed to the next stage. This approach aligns with literature showing NLP can *"extract and summarize essential information from policy documents, allowing consumers to easily comprehend their coverage terms"*, here applied to claims paperwork.

b) Claim Submission and Decision

The structured claim data (from Document Processing) and additional user profile info (existing coverage, past claims, demographics) are fed into a decision model. This model may combine rule-based logic (e.g. simple checks like coverage limits) with a trained machine learning classifier (e.g. gradient boosting or neural network) that predicts claim approval probability and reimbursable amount. The ML model is trained on historical claims data, learning patterns of accepted vs. rejected claims and typical payment calculations. If the model approves the claim, it outputs an authorized payout; if it denies, it outputs denial. Critically, the model can incorporate the **normalized cost** from our hospital dataset (see Methodology) to flag unrealistic charges. For example, if the billed amount far exceeds benchmarks, the system may adjust the payable amount or escalate to manual review. This automated decision-making reflects research showing AI can *"help identify and reduce common biases in healthcare, detect complex patterns… and ultimately lead to more informed decision-making*.

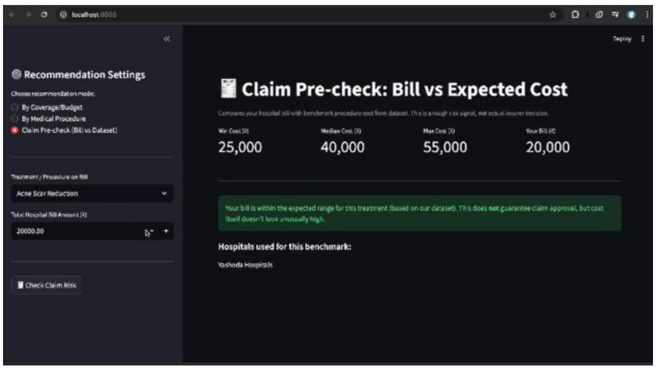

Fig 1. Claim Pre-Check

c) Explainable AI:

After the decision is made, this module generates an explanation for the user or claims adjuster. Techniques include:

(1) **Feature Attribution:** computing SHAP or LIME values to rank input factors by importance. For instance, "Procedure cost ($X)" and "Coverage type (HMO)" may be top contributors.

(2) **Decision Rules Trace:** if simple rules triggered a flag (e.g. "Exceeded annual limit"), cite them explicitly.

(3) **Natural Language Summary:** use a template or LLM to produce a natural-language justification, e.g. "Claim denied: the procedure exceeds policy coverage by $Y." According to Bora *et al.*, such explanations help domain experts verify outcomes. By presenting these insights, the system ensures *transparency* — users see *why* the claim was handled a certain way, addressing a common pain point in AI insurance applications.

d) Fraud Detection

In parallel or as part of decision logic, the claim is evaluated by an anti-fraud model. This module uses supervised ML (e.g. random forest, XGBoost) or unsupervised techniques (autoencoders for anomaly detection) trained on known fraudulent vs. genuine claims. Features include unusual billing patterns, mismatched coding frequencies, past fraud scores of providers, etc. NLP features from the claim description (identified in Document Processing) are also input. As Cheekaramelli (2025) demonstrated, NLP on claim narratives significantly boosts fraud detection accuracy. If the model predicts high fraud risk, the system flags the claim for manual review or automatic rejection based on thresholds. This reduces false approvals of fraud and enhances cost-control. The output (a risk score or binary flag) is included in the final decision record.

e) Policy Recommendation:

If a claim is denied or partially approved, the user may benefit from advice on other policies or coverage supplements. This module takes the user's profile (age, health history, claim needs) and retrieves suitable policies from a database of offerings. We implement a retrieval-augmented recommendation: user requirements are converted into an embedding, and nearest-neighbor search (e.g. FAISS with sentence-transformer embeddings) finds top-matching policies. A small LLM (or rule engine) can then summarize why a recommended policy fits the user's needs. Hanmante *et al.* report that leveraging RAG and synthetic policy data enables *customized policy recommendations* effectively. Similarly, our system generates a ranked list of plans (with coverage amounts and premiums) explaining each choice, aiding users in selecting better coverage.

f) User Document Management:

All uploaded documents and generated data are stored securely (e.g. in an encrypted database or storage bucket). Users can log in to view past claims, approvals, explanations, and recommended policies. The document management subsystem indexes policy documents and claim records for quick retrieval. It also supports versioning (if policies update) and audit trails (tracking who accessed or modified records).

This persistent storage ensures continuity of service and compliance with data retention regulations.

C. Integration and Data Flow

In operation, these modules interact via APIs or internal function calls. For example, after Document Processing extracts fields, it invokes the Decision Engine; the Decision Engine calls the Fraud module and then the XAI module. The Policy Recommendation engine may be triggered by user request or certain decision outcomes. This modular design allows independent scaling (e.g., deploying the NLP service separately from the ML inference engine). Throughout, data exchanges occur over secure channels and use common data schemas (JSON structures) to represent claims, explanations, and recommendations. The overall architecture mirrors multi-agent AI frameworks where a central orchestrator delegates tasks to specialized components.

D. Tech Stack

1. Programming Language: Python (due to rich ML and NLP libraries).

2. Document Processing: Tesseract OCR (via pytesseract), PDF libraries (PyMuPDF), and NLP (spaCy or HuggingFace Transformers). Pre-trained models (BERT, BioBERT) may be fine-tuned on insurance/medical data.

3. Machine Learning: scikit-learn for classical models, XGBoost/LightGBM for gradient boosting, PyTorch/TensorFlow for deep networks.

4. Explainability: SHAP and LIME libraries for feature attribution; custom templates for rule-logging.

5. Database: PostgreSQL or MongoDB for structured data; Elasticsearch or similar for fast text retrieval.

6. Retrieval & Embeddings: Sentence Transformers (e.g. all-MiniLM-L6-v2) to embed text, FAISS for nearest-neighbor search in policy recommendation.

7. Web Framework: Streamlit for the front end; FastAPI or Flask for hosting the ML pipeline. Docker containers orchestrate services.

8. Cloud/Deployment: AWS or Azure cloud (EC2, S3) can host the services and data.

9. Other: Pandas, NumPy for data handling; LangChain or similar toolkits for LLM orchestration if using a chain-of-thought retrieval.

This stack ensures quick integration of models and libraries. We follow common software engineering practices: version control (Git), CI/CD for model updates, and logging for audits.

IV. Methodology

A. Hospital Cost Scraping and Benchmarking:

To ground claim evaluations in real-world prices, we compile a database of procedure costs. We scrape publicly available sources: hospital websites that publish price lists (often required by transparency laws), government datasets (e.g. Medicare/All-Payer datasets), and consumer sites (like FAIR Health). Python tools (requests, BeautifulSoup) extract items (procedure code, description, cost). This raw data is cleaned and merged; for the same procedure across providers, we record statistics (mean, median, std deviation). If online data is sparse, we may solicit costs via APIs or web portals. The result is a “master price list” for common services (e.g. chest X-ray, appendectomy) by region or hospital class. These benchmarks allow the system to detect outliers in billed amounts and to suggest fair reimbursement.

B. Cost Normalization and Dataset Building:

Collected costs are normalized to a consistent scale. For example, costs from different years are adjusted for inflation; costs from different cities are adjusted for local cost-of-living indices. We convert all costs to a standard currency (e.g. USD or INR). We may train a regression model (or simple normalization factors) that predicts expected cost of a procedure given features (hospital rating, location, etc.). The normalized dataset then serves two purposes: (1) At decision time, comparing the claim’s billed amount to expected range (flagging excessively high charges); (2) Training input for the decision model (the “true” claim value). We store this data in a structured database (SQL or NoSQL).

C. Integration with Insurance Decision System:

The scraped and normalized cost data is integrated into the claim decision logic. During training, historical claims paired with actual paid amounts help tune the decision model’s thresholds. At inference, if a claim’s cost significantly exceeds the normalized benchmark (beyond a tolerance range), the decision logic can reduce the payout or recommend further verification. This benchmarking step ensures cost fairness. For instance, if a knee surgery normally costs $8,000 but the claim is $12,000, the system may only pay up to $9,000 or require proof. Integration is achieved by embedding cost lookups into the Decision Engine’s feature set.

D. Streamlit-based Frontend with Decision Logic:

The user interface is built with Streamlit for rapid development of interactive dashboards. Users (claimants or agents) can upload claim documents through the browser. Streamlit callbacks handle the submission: the documents are sent to the back-end pipeline, and the returned decision and explanations are displayed on-screen. The interface presents: (1) a form showing extracted claim details (for user confirmation), (2) the claim status (Approved/Denied) and calculated payout, (3) an explanation panel (text and possibly charts of feature importances), (4) fraud alert (if any), and (5) recommended policies list. We also build an administrative view for underwriters to see all details and override decisions if needed. Streamlit is paired with a lightweight API (e.g. FastAPI) that hosts the ML models and NLP services. This setup allows the entire logic (data processing and model inference) to be called from the UI seamlessly. Using Streamlit ensures the front end is intuitive (drag-and-drop for docs, buttons for actions) and can run on web or mobile..

V. Results and Discussion

We anticipate that the AI-powered system will deliver significant improvements in claims processing. Based on prior studies, we expect up to 30% faster decision times and notable reduction in manual workload
carijournals.org
pmc.ncbi.nlm.nih.gov
. Claims that previously took days can be resolved in near-real-time, as textual extraction and inference are automated. Accuracy is expected to improve as well: Cheekaramelli et al. report that NLP-enhanced fraud detection "increases fraud detection accuracy by 30%"; we expect our fraud module to similarly flag more true fraud cases (and reduce false positives). The integration of normalized cost data should also reduce overpayment errors.

On transparency, the XAI component should enhance trust. Literature suggests that explanations "help check correctness" and build user confidence. By presenting clear reasons (e.g. "Coverage limit exceeded" or "Pre-existing condition not covered"), customers will understand outcomes better, leading to higher satisfaction and fewer appeals. Our policy recommendation feature adds user value: by comparing profile to policy features, we offer customized advice, potentially increasing customer retention and satisfaction.

We also expect business benefits. Automated claims reduce labor costs and cycle time, aligning with industry reports of up to 80–90% reduction in processing time for certain tasks when using AI. Transparent decisions can lower legal risk and improve compliance with regulations. Overall, the system should transform the customer experience from opaque and slow to transparent and efficient, aligning with the goal of a "more efficient, data-driven, and customer-centric" insurance model.

VI. Conclusion

This work proposes a comprehensive AI-driven system for insurance claim adjudication and advice. By unifying NLP-based document understanding, automated decision logic, explainable AI, fraud detection,

and policy recommendations, the system addresses key pain points in claims processing. Automation promises faster turnarounds and consistency, while explainability and policy suggestions improve transparency and customer service. The methodology – from hospital cost scraping to Streamlit front-end – demonstrates how modern AI tools can be integrated in practice. Future work may involve deploying the system in pilot trials, expanding the knowledge base (e.g. integrating live hospital APIs), and adding continual learning to models as new claim data arrives. Ultimately, the system aims to shift the insurance process toward one that is **transparent, proactive, and user-friendly**, benefiting both insurers and insured

VII. Acknowledgement

The authors wish to thank the faculty of the Department of Computer Science Engineering at Vishwakarma Institute of Technology for their invaluable guidance and support throughout this project. We are also grateful for the insightful suggestions from the anonymous reviewers, which have significantly improved the quality of this paper.

References

[1] M. J. R. Cheekaramelli, "Using Natural Language Processing (NLP) to Identify Fraudulent Healthcare Claims," *International Journal of Computing and Engineering*, vol. 7, no. 3, 2025.

[2] S. Hanmante, S. Patil, and A. K. Shahade, "A multi-module AI system for intelligent health insurance support using Retrieval Augmented Generation," *Scientific Reports*, 2025.

[3] S. Bhattacharya, G. Castignani, L. Masello, and B. Sheehan, "AI revolution in insurance: bridging research and reality," *Frontiers in Artificial Intelligence*, vol. 8, 2025.

[4] S. Bora, S. Sahu and S. Goel, "AI and Explainable Machine Learning for Insurance Premium Prediction," *Machine Learning with Applications*, vol. 11, 2023.

[5] H. Kapadiya et al., "AI and Blockchain framework for Insurance Fraud Prevention," *IEEE Int. Conf. on Data and Communication Systems*, 2022.

[6] A. Rajagopal, "Automation of Claim Processing Using Artificial Intelligence," *Journal of Insurance and Financial Technology*, 2020.

[7] J. Gu, H. Li, and P. Chen, "Document understanding for enterprise policy analysis using transformer-based NLP models," *IEEE Access*, vol. 10, 2022.

[8] Y. Zhang and Q. Wu, "Using BERT for legal document clause classification," *ACM SIGAI*, 2021.

[9] Z. Cao et al., "Hybrid intelligent claim approval using deep learning and rule-based audit," *Proc. IEEE Big Data*, 2023.

[10] M. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You? Explaining Predictions of Any Classifier," *Proc. ACM SIGKDD*, 2016.

[11] S. Lundberg and S. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[12] P. Wu and J. Wang, "Anomaly detection in insurance claims using unsupervised deep autoencoders," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 2, 2022.

[13] FAIR Health, "Benchmarking Medical Procedure Costs in India and U.S.," FAIR Health Public Datasets, 2024.

[14] National Health Authority, India, "Ayushman Bharat — Treatment Cost Packages," Public Release, 2024.

[15] Surya Hospitals & Yashoda Hospitals, "Price Transparency and Procedure Cost Listings," Public Domain Web Data, accessed 2024.