WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Secure Score: AI-Based Insurance Policy Analytics System

Ms. Sini Prabakaran, Kanishkaa.S, Karpagalakshmi A, Mathumitha B, Mithra Devi M

Sri Shakthi Institute of Engineering And Technology, Coimbatore, Tamil Nadu, India.
E-mail ID: skanishkaa9@gmail.com , karpagalakshmia7@gmail.com , bmathumitha06@gmail.com , mithradevi1903@gmail.com .

ABSTRACT

Insurance companies process a large number of claims and policy documents, making claim verification and risk assessment important parts of insurance management. Manual verification can be time-consuming, particularly when reviewers need to examine supporting documents, identify repeated claims, and investigate unusual billing patterns. To address these challenges, this project presents SecureScore, an AI-based insurance policy analytics system designed to support insurance claim analysis through machine learning and document processing.

The proposed system combines claim classification, duplicate claim detection, billing anomaly screening, document information extraction, and policy information retrieval within a unified platform. A Random Forest classification model is used to analyse claim data, while a Flask backend connects the analytical components with a React and TypeScript user interface. The system also includes document chunking, vectorization, and evidence retrieval to help users locate relevant information in policy documents.

The classification model was evaluated using a synthetic health insurance claims dataset containing approximately 19,962 records and 37 features. The experimental results achieved a ROC-AUC score of 0.885, accuracy of 0.869, precision of 0.762, recall of 0.688, and F1-score of 0.724. These results demonstrate the model's classification performance on the experimental dataset. Further evaluation using suitable real-world data is required before applying the findings to actual insurance claims. SecureScore aims to support human reviewers by organizing claim information and identifying records that may require further investigation.

Keywords: SecureScore, Insurance Analytics, Machine Learning, Random Forest, Claim Verification, Duplicate Detection, Anomaly Detection, Policy Information Retrieval.

INTRODUCTION

Insurance provides financial protection against unexpected expenses and losses. In the healthcare sector, insurance claims must be examined to ensure that the submitted information is consistent with the relevant policy conditions. This process may involve checking claim details, reviewing documents, identifying repeated submissions, and investigating unusual billing entries.

Traditional verification methods can require considerable manual effort. As the number of claims increases, reviewing every record becomes more challenging. Inconsistent information and unusual claim patterns may also require additional investigation.

Recent developments in artificial intelligence and machine learning provide opportunities to improve insurance data analysis. Classification algorithms can identify patterns in structured claim data, while document processing and information retrieval techniques can help users locate relevant information in policy documents. Anomaly detection can also highlight records that differ from expected patterns.

To address these requirements, this project proposes SecureScore, an AI-based insurance policy analytics system that combines several claim analysis functions within one application. The system uses a Random Forest classifier to analyse claim records and provides additional functionality for duplicate detection, billing anomaly screening, document extraction, and policy information retrieval.

The main objective of SecureScore is to organize claim-related information and assist reviewers in identifying records that need closer examination. The system is intended to support decision-making rather than independently approve or reject insurance claims.

1. SYSTEM ARCHITECTURE

The SecureScore architecture consists of a user interface, backend services, a machine learning model, document processing functions, and information retrieval components. These modules work together to process claim information and present analytical results through a centralized interface.

1.1 User Interface Module

The User Interface Module provides an interactive environment for accessing SecureScore's features. It is developed using React and TypeScript and presents claim information and analytical outputs in an organized format.

Users can examine classification results, risk indicators, and other information returned by the backend. The interface is designed to make the results easier to understand and helps users access the available functions from a single application.

1.2 Data Processing and Management Module

This module prepares claim information for analysis. It handles input validation and organizes the features required by the machine learning model.

Consistent data preparation is important because differences between training data and prediction inputs can affect model performance. The module also coordinates information exchange between the frontend, backend, and analytical components.

1.3 Machine Learning Module

The Machine Learning Module uses a Random Forest classifier to classify insurance claims according to patterns learned from the training dataset. Random Forest combines multiple decision trees to generate predictions based on the input features.

The model is trained using prepared claim data, while a separate validation dataset is used to select the classification threshold. The trained model then produces predictions for unseen records.

The output helps identify claims that may require additional review. However, a positive prediction does not prove fraudulent activity, and the result should be interpreted alongside supporting information.

1.4 Duplicate Claim and Billing Anomaly Detection Module

This module provides additional screening functions for identifying potentially repeated claims and unusual billing records.

The duplicate detection component examines claim information for repeated or overlapping details. The billing anomaly component highlights records that differ from expected patterns according to the implemented checks.

These functions can help reviewers identify records that deserve closer examination. However, similar claims may be legitimate, and unusual billing values do not automatically indicate fraud. Further verification is necessary before reaching a conclusion.

1.5 Document Processing and Policy Retrieval Module

Insurance policies contain coverage details, conditions, and restrictions that may be relevant to claim assessment. Finding specific information in lengthy documents can require time and effort.

SecureScore includes document information extraction and retrieval functionality to help users locate relevant policy content. The implementation uses document chunking, vectorization, and evidence retrieval to search processed document text.

The retrieved passages can assist users in finding supporting information. Important details should still be verified against the original policy document.

1.6 Backend Services Module

The backend is developed using Flask and connects the frontend with the analytical components. It receives requests, coordinates the required processing, and returns results to the user interface.

The backend supports the integration of claim classification, document processing, and other implemented analytical functions. Separating backend services from the frontend helps organize the application and supports independent testing of its components.

2. LITERATURE REVIEW

Machine learning methods are increasingly used to analyse structured datasets and identify patterns that may be difficult to recognize through manual examination alone. Classification algorithms such as decision trees, Random Forest, and logistic regression can learn from labelled examples and generate predictions for new records.

Random Forest is an ensemble learning method that combines the outputs of multiple decision trees. Its ability to model relationships among several input features makes it suitable for exploring classification problems. However, its performance depends on data quality, class distribution, feature selection, and the evaluation procedure.

Anomaly detection provides a complementary approach by identifying records that differ from expected patterns. In insurance analysis, unusual records may deserve further investigation, although an anomaly does not necessarily indicate fraudulent behaviour.

Document processing and information retrieval also have an important role in insurance claim assessment. Text extraction and retrieval methods can help users locate relevant policy information without manually examining an entire document.

SecureScore combines these approaches in one application. Its main contribution is the integration of machine learning-based claim classification with duplicate detection, billing anomaly screening, document information extraction, and policy information retrieval. The classification model has been evaluated experimentally, while the remaining components require separate testing to establish their effectiveness.

3. METHODOLOGY

The development of SecureScore follows a structured process involving requirement analysis, data preparation, model training, application integration, and evaluation.

3.1 Requirement Analysis

The system was designed to support the following functions:

- Classification of insurance claims using machine learning.
- Identification of potentially duplicated claim records.
- Screening of unusual billing information.
- Extraction of information from supported documents.
- Retrieval of relevant policy information.
- Presentation of analytical results through a user interface.

3.2 Dataset Preparation

The classification model uses a synthetic health insurance claims dataset containing approximately 19,962 records and 37 features. The dataset contains 4,940 positive records and 15,022 negative records.

The records are divided into training, validation, and test subsets using stratified splitting. The training subset is used to fit the model, the validation subset supports threshold selection, and the test subset is used for final evaluation.

Since the dataset is synthetic, the results demonstrate performance within the experimental environment and cannot automatically be generalized to actual insurance claims.

3.3 Model Training and Evaluation

A Random Forest classifier is trained using the prepared dataset. The model learns patterns from the input features and produces a classification score for each record.

The classification threshold is selected using the validation subset, with F1-score used as the selection criterion. The selected threshold for the reported experiment is 0.35.

The model is evaluated using accuracy, precision, recall, F1-score, ROC-AUC, and average precision. These metrics provide information about overall classification performance and the model's ability to identify positive records.

3.4 Technology Stack

The system uses the following technologies:

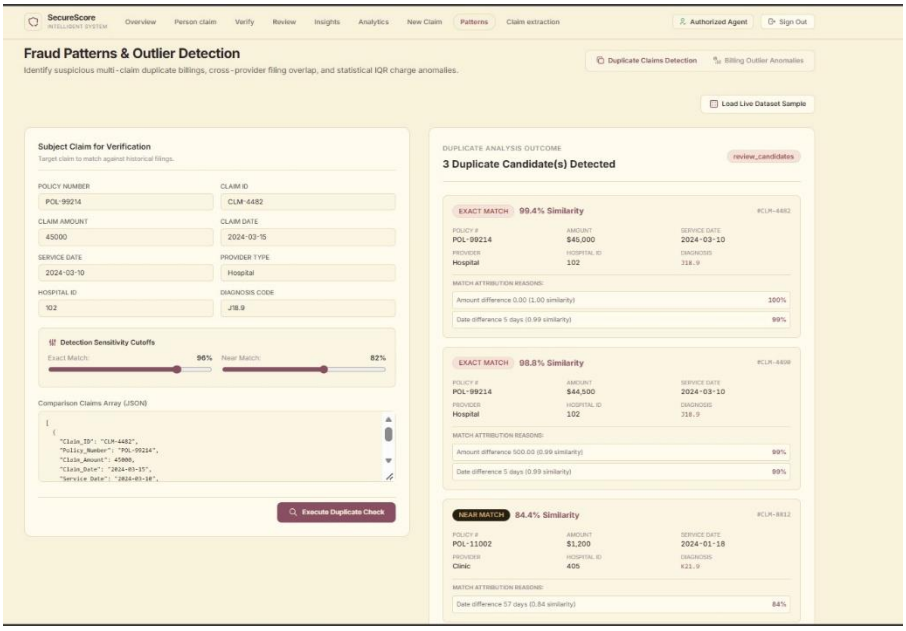
- **Python:** Data processing and machine learning.
- **Scikit-learn:** Model training and evaluation.
- **Flask:** Backend services and API handling.
- **React and TypeScript:** Frontend development.
- **Document processing and vectorization:** Preparation and retrieval of policy information.

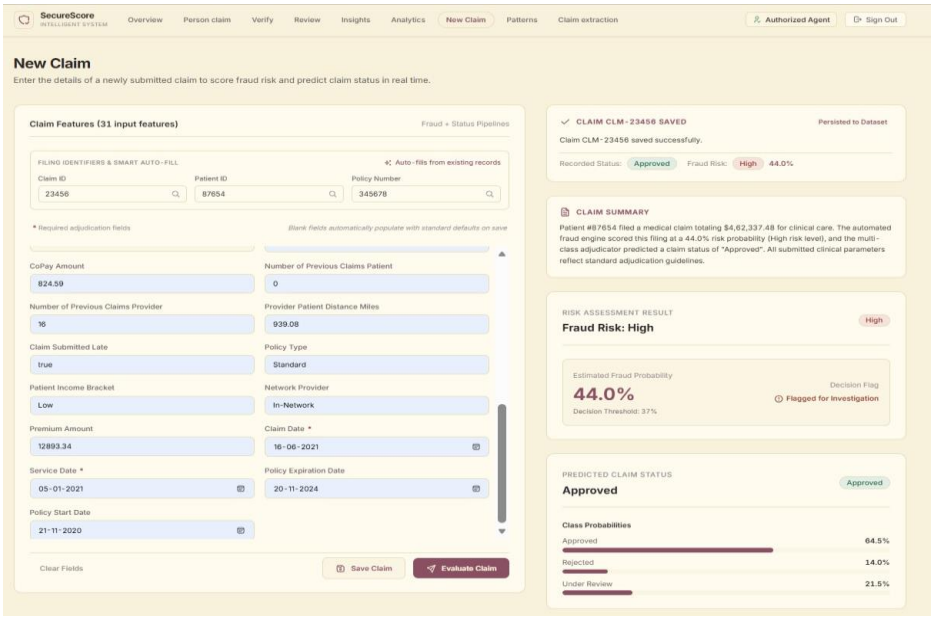
- **Git and GitHub:** Source code management and version control.

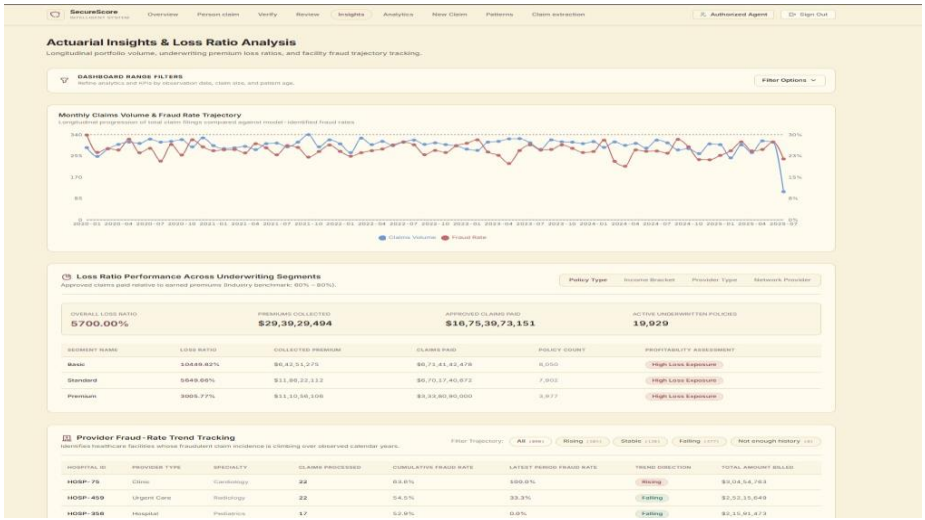
3.5 Integration and Testing

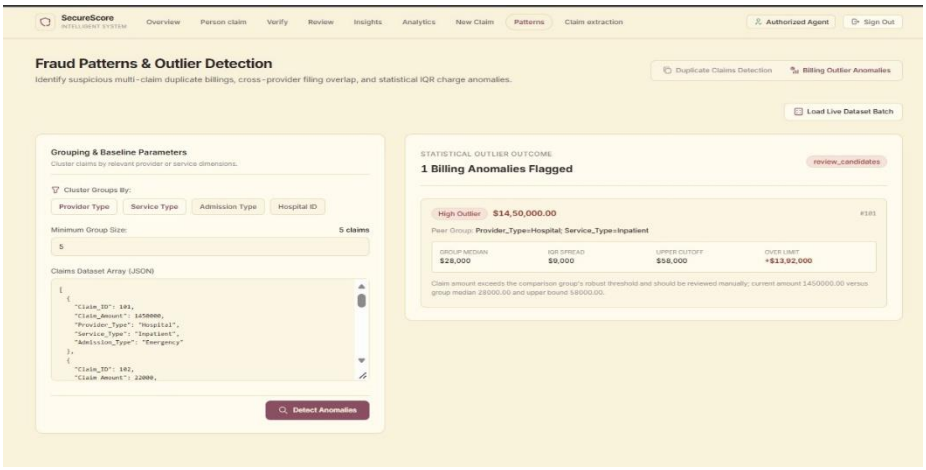
The backend connects the analytical components with the frontend. Requests are processed by the relevant function, and the resulting information is returned to the interface.

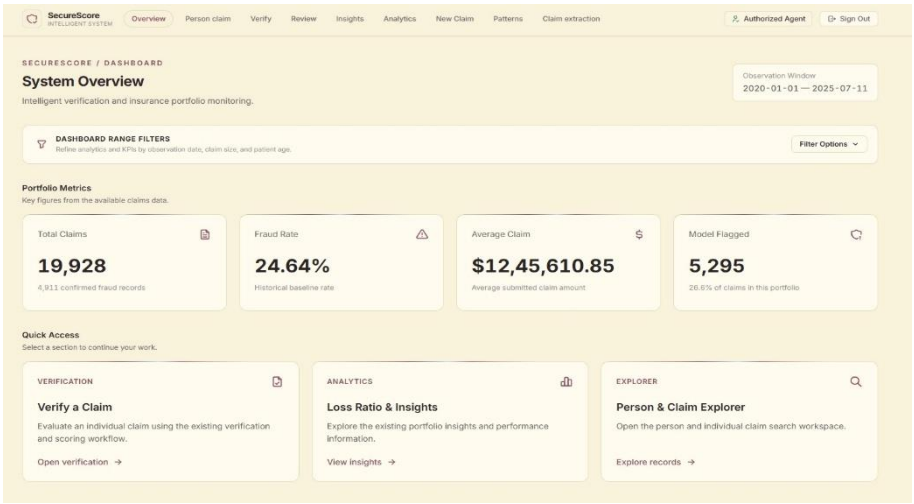
Testing should cover model predictions, input validation, duplicate detection, billing screening, document extraction, policy retrieval, and interface behaviour. Each component should be evaluated separately to identify errors and determine whether it performs its intended function.











4. RESULT ANALYSIS

The Random Forest classifier was evaluated using a held-out test subset containing 3,993 records. The selected threshold was 0.35.

The following results were obtained during the reported experimental run.

Metric	Result
Test records	3,993
ROC-AUC	0.885

Metric	Result
Average Precision	0.812
Accuracy	0.869
Precision	0.762
Recall	0.688
F1-score	0.724

The model achieved an accuracy of 86.9%, indicating that most records in the test subset were classified correctly. The ROC-AUC score of 0.885 indicates that the model could distinguish between the two classes to a useful degree within the experimental dataset.

The precision of 0.762 shows that approximately 76.2% of the records predicted as positive were positive according to the dataset labels. The recall of 0.688 indicates that the model identified approximately 68.8% of the actual positive records.

The reported confusion matrix is shown below.

Actual Class	Predicted Negative	Predicted Positive
Negative	2,793	212
Positive	308	680

The results show that the model made both false-positive and false-negative predictions. Therefore, its output should be treated as a screening indicator rather than definitive evidence of fraud.

These metrics evaluate the classification model only. Separate experiments are needed to measure the accuracy of duplicate detection, billing anomaly screening, document extraction, and policy information retrieval.

5. CONCLUSION AND FUTURE WORK

SecureScore presents an integrated approach to insurance policy analytics by combining machine learning-based claim classification, duplicate detection, billing anomaly screening, document information extraction, and policy information retrieval.

The application uses Flask for backend processing and React with TypeScript for its user interface. The Random Forest classifier was evaluated using a synthetic health insurance claims dataset and achieved a ROC-AUC score of 0.885, accuracy of 0.869, precision of 0.762, recall of 0.688, and F1-score of 0.724.

These findings demonstrate the model's performance within the experimental dataset. However, the results do not establish how accurately the system would perform on real insurance claims. Further testing is necessary, and the system should support human review rather than make consequential decisions independently.

Future enhancements include:

- Evaluating the model using suitable real-world insurance data.
- Comparing different machine learning algorithms.
- Improving the detection of duplicate and unusual claim records.
- Evaluating document extraction and policy retrieval accuracy.
- Improving the presentation of supporting evidence.
- Strengthening data protection and access controls.
- Examining model performance across relevant groups.
- Conducting usability testing with representative reviewers.

These improvements would help determine the reliability and practical usefulness of SecureScore in insurance claim assessment.

REFERENCES

1. Breiman, L. “Random Forests.” *Machine Learning*, vol. 45, pp. 5–32, 2001.

2. Saito, T., and Rehmsmeier, M. “The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets.” *PLOS ONE*, vol. 10, no. 3, 2015.
3. Lundberg, S. M., and Lee, S.-I. “A Unified Approach to Interpreting Model Predictions.” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
4. Pedregosa, F., et al. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
5. Flask Project. *Flask Documentation*.
6. React Team. *React Documentation*.