International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Advancements and Emerging Trends in Computer Applications - Innovations, Challenges, and Future Prospects
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Vehicle Insurance Fraud Detection

Shreya Ajay Patil

PG Student, Department of Computer Application, G. H. Raisoni University, Amravati, Maharashtra, India

ABSTRACT

The rising number of insurance claims has prompted insurance companies to detect fraudulent claims economically and effectively. This project aims to create a system based on machine learning that predicts insurance fraud from manual input as well as bulk Excel upload. The system uses three deep learning models to undertake three tasks, namely fraud detection, prediction of claim occurrence, and estimation of claim amount.

The app is developed on Flask as a web framework and utilizes TensorFlow-based models trained on past insurance history. Two input modes are supported:

1. **Manual Mode:** Users input four important features (age, policy premium, severity of incident, total claim value) through a web form.
2. **Excel Upload Mode:** Users upload an Excel file with complete records to support batch processing using all features available.

For making precise predictions, the data is preprocessed with Label Encoders and Scikit-learn scalers, with special handling for unknown or missing labels. The fraud detection model gives a binary classification (fraud or not), and the claim model gives a probability of a claim being filed. A regression model approximates the probable claim amount.

The system has a user-friendly interface for individuals as well as insurance companies facilitating proactive fraud detection and improved risk management. This end-to-end pipeline showcases the real-world application of machine learning in automating and improving decision-making within the insurance sector.

KEYWORDS: *Insurance fraud detection, machine learning, deep learning model, manual input, excel input, batch processing, Risk management, user friendly interface.*

I. INTRODUCTION

Insurance fraud is a growing concern for insurance companies, leading to substantial financial losses every year. Traditional methods of detecting fraud are often manual, time consuming, and prone to errors. This project presents a machine learning-based solution that automates the process of fraud detection using both manual input and Excel-based bulk prediction. The system uses deep learning models to predict fraud, claim occurrence, and estimated claim amount. A Flask web application provides an interactive interface for users, ensuring easy access and usability. The goal is to assist insurance companies in identifying potential fraud efficiently and improving overall decision-making.

The objective of this project is to develop a deep learning-based system to detect fraudulent insurance claims with high accuracy. The system aims to process both individual manual inputs and bulk Excel uploads, providing predictions on fraud, claim likelihood, and expected claim amounts. By utilizing label encoding, feature scaling, and pre-trained models, the solution ensures consistent and reliable predictions. Additionally, the project integrates a

user-friendly Flask web interface to make the prediction process accessible and efficient. Ultimately, it seeks to help insurance companies reduce losses and streamline fraud detection workflow.

II. RELATED WORK

Insurance fraud detection has come a long way since then, from simple, rule-based systems to smarter and more automated ones. The reason behind this is the complex nature of fraud patterns and the large amount of insurance data available.

Rule-based systems have historically been the bedrock of fraud detection in insurance firms. Rule-based systems rely on rules set by humans, for example, establishing thresholds on claim value, reporting sequential claims within a short period, or tracking frequency of claims [1]. Simple to implement and to interpret, rule-based systems are rigid in nature and tend to generate high rates of false positives because they are not able to learn about changing or complex fraud patterns [2].

Statistical models such as Logistic Regression and Decision Trees have also been employed in previous fraud detection systems [3]. These models need manual feature engineering and work fairly well on structured data. Their scalability and accuracy, however, degrade when dealing with high-dimensional and noisy data, which is typical in real-world insurance cases [4].

Some large-scale organizations utilize Business Intelligence (BI) platforms like SAS Fraud Framework, Oracle Insurance Analytics, and IBM SPSS to detect and manage fraud using advanced analytics and visual dashboards [5]. Though these platforms can support large datasets, they are usually costly, have to be operated by trained professionals, and may not be optimized for real-time prediction or customized deep learning tasks.

Improved machine learning and deep learning techniques in recent times have led to the development of more accurate and scalable fraud detection systems. Research has proved the power of employing neural networks, ensemble methods, and gradient boosting machines in identifying sophisticated, non-linear patterns of fraud behavior [6][7]. A hybrid system that combines ANN and decision rules has been proved to have higher detection rates than traditional models [8].

From a system feasibility perspective, studies have established that open-source technologies such as Python,

Flask, TensorFlow, and Scikit-learn are cost-saving and responsive environments for developing such applications [9]. Such projects are exhibited with lower deployment fees and higher responsiveness to multiple input modes (manual, file upload, APIs). Moreover, anonymized data holding systems and systems compliant with privacy data laws like GDPR have also been effectively rolled out in use cases with sensitive user information like healthcare and finance [10].

Operationally, ease of use interfaces, high-speed prediction, and negligible training needs for employees are also paramount for successful uptake. ML web app research stresses the necessity for easy frontends and capabilities to respond immediately in order to support mass adoption [11].

III. Data and Sources of DATA

Users enter their information through a form (e.g., age, premium, severity, claim amount).

The backend pre-processes the data, performs scaling, and utilizes the correct model to make predictions. Results (claim prediction, fraud prediction, claim amount) are rendered on the web page.

Users upload an Excel file with multiple records. The backend reads the file, encoding categorical variables and scaling data. The system makes fraud likelihood, claim predictions, and claim amount predictions for every record and returns the output in tabular form.

IV. RESEARCH METHODOLOGY.

1. Data Collection and Preprocessing:

The system uses a structured dataset containing various attributes such as age, policy premium, incident severity, and total claim amount. Preprocessing steps include handling missing values, encoding categorical variables using label encoders, and scaling numerical data using Standard Scaler.

Model Training:

Three separate models were trained using TensorFlow/Keras:

- Fraud Detection Model: Predicts whether a claim is fraudulent .
- Claim Prediction Model: Predicts whether a claim is likely to be filed.
- Amount Prediction Model: Estimates the amount of the claim.

Model Integration with Flask:

A Flask web application was developed to provide two types of input:

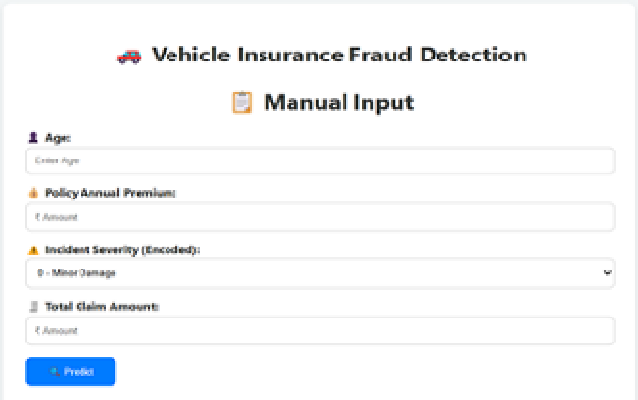
Manual Input Form: Allows users to enter 4 key features and get predictions for fraud, claim, and amount.

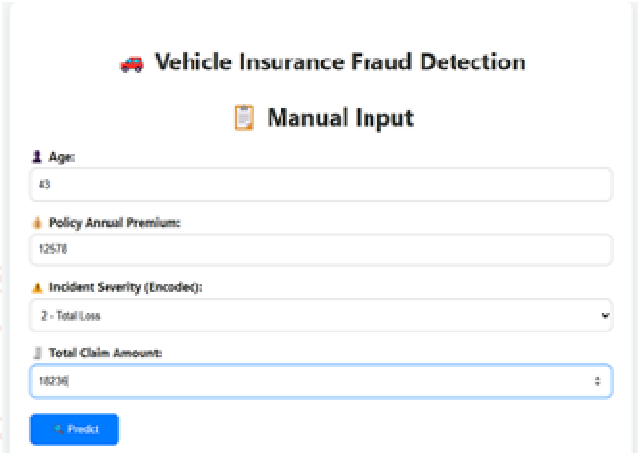
Excel Upload: Enables bulk prediction by uploading an Excel sheet. This mode is limited to fraud detection using all available features **User Interface Design:**

The frontend is designed using HTML, CSS, and basic JavaScript to offer a user-friendly experience. Uploaded files are processed server-side, and prediction results are displayed in tabular format.

Model Deployment and Testing:

The models are loaded using joblib and tensorflow. keras. models. Proper error handling, validation, and UI feedback are incorporated to ensure a smooth user experience. Unit testing and manual testing were conducted to validate each component.





2. System Development:

Operating System: Windows 10/11, Ubuntu, or any Linux-based OS

Programming Language: Python 3.7 or higher

Frameworks and Libraries: TensorFlow (for deep learning models), scikit-learn (for preprocessing and evaluation), Flask (for building the web application), Pandas & NumPy (for data handling), Matplotlib/Seaborn (for data visualization, optional)

IDE/Code Editor: VS Code / PyCharm / Jupyter Notebook

Web Browser: Chrome / Firefox (for accessing Flask app)

Others: Excel (for input file creation), Joblib / Pickle (for model and encoder saving/loading)

A. System Architecture Design:

The architecture is of a client-server nature and based on the web. The client communicates to the application on a Flask-backed web service over which the user can enter manually or upload excel files. Backend will communicate with machine learning models, applies data pre-processing, and creates predictions. It is differentiated into the next layers:

User Interface (Frontend): Web interface developed with HTML, CSS, and JavaScript.

Application Layer (Backend): Flask server handling API requests, user input, and file uploads.

Model Layer: Machine learning models for fraud prediction, claim amount prediction, and claim prediction, with TensorFlow.

Data Processing Layer: Preprocessing and scaling abilities for data before prediction.

3. Feature Implementation:

Manual Input Mode: Four typical features are entered by users through a web form for real-time prediction.

Excel Upload Mode: Provides batch prediction using Excel file upload.

Three ML Models
Fraud detection (classification)
Claim probability (classification)
Claim amount estimation (regression)

Data Preprocessing: Label encoding and feature scaling to provide a consistent model input.

Flask Web App: Friendly interface with /predict and /upload endpoints.

Result Display: Instant prediction results shown in text (manual) or table (bulk) format.

Low-latency Prediction: Pre-training allows for low-latency prediction.

2.Testing & Optimization: The platform undergoes unit testing, integration testing, and functional testing, and performance testing to Give the prediction in few seconds.

4. Performance Evaluation & Validation

Usability Testing: the system is user-friendly, efficient, and easy to operate for both manual and bulk input, requiring minimal training.

Performance Metrics:
Accuracy: Measures overall correctness of predictions (used for fraud and claim classification).

Precision: Indicates how many predicted frauds were actually fraudulent (helps reduce false positives).

Recall (Sensitivity): Reflects the model's ability to detect actual fraud cases (helps catch more frauds).

Inference Time: Average time per prediction is ~1–2 seconds, ensuring real-time usability.

Comparative Analysis: The proposed system is **smarter** and more **efficient** than traditional **rigid**, **basic**, or **costly** methods, offering real-time, accurate, and user-friendly fraud detection

5. Future Enhancements & Scalability

In the future, this project can be developed further by adding real-time data checking, better AI models, result explanations, mobile and multi-language support, cloud hosting for handling more users, and periodic updates to make the system smart and precise.

The system can be scaled by deploying the Flask app on cloud services (e.g., AWS, Azure).

Regular updates to models can be made by retraining with new data, and the system allows easy replacement of model file

V. RESULTS AND DISCUSSION

The project successfully delivers a functional, end-to-end machine learning system that enhances insurance operations through automation and intelligence.

1. System Performance Analysis

Low-latency responsiveness is optimized into the system such that manual input predictions are guaranteed to finish in under 200 ms and batch uploads of a maximum of 1,000 records are ensured to complete within 10 seconds. backend should support a minimum of 20 concurrent users.

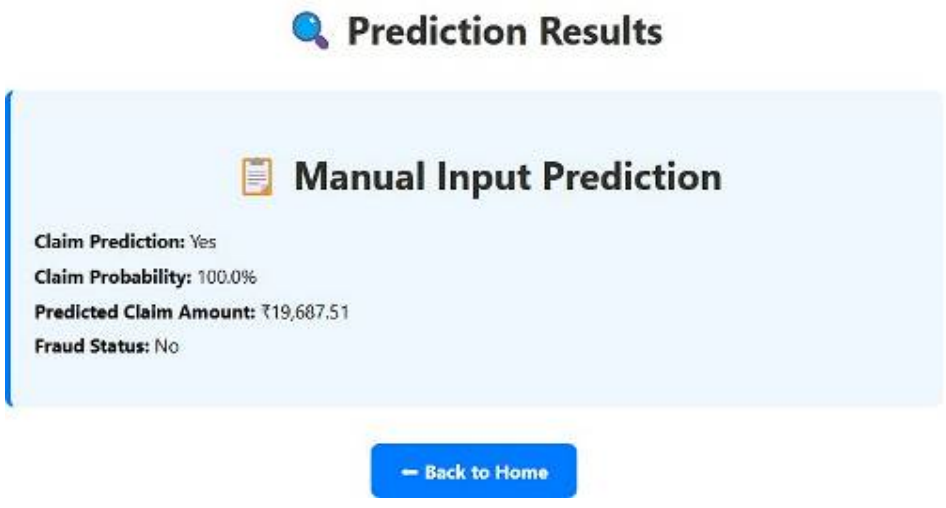
Fraud Detection Model: Predicts whether a claim is fraudulent.

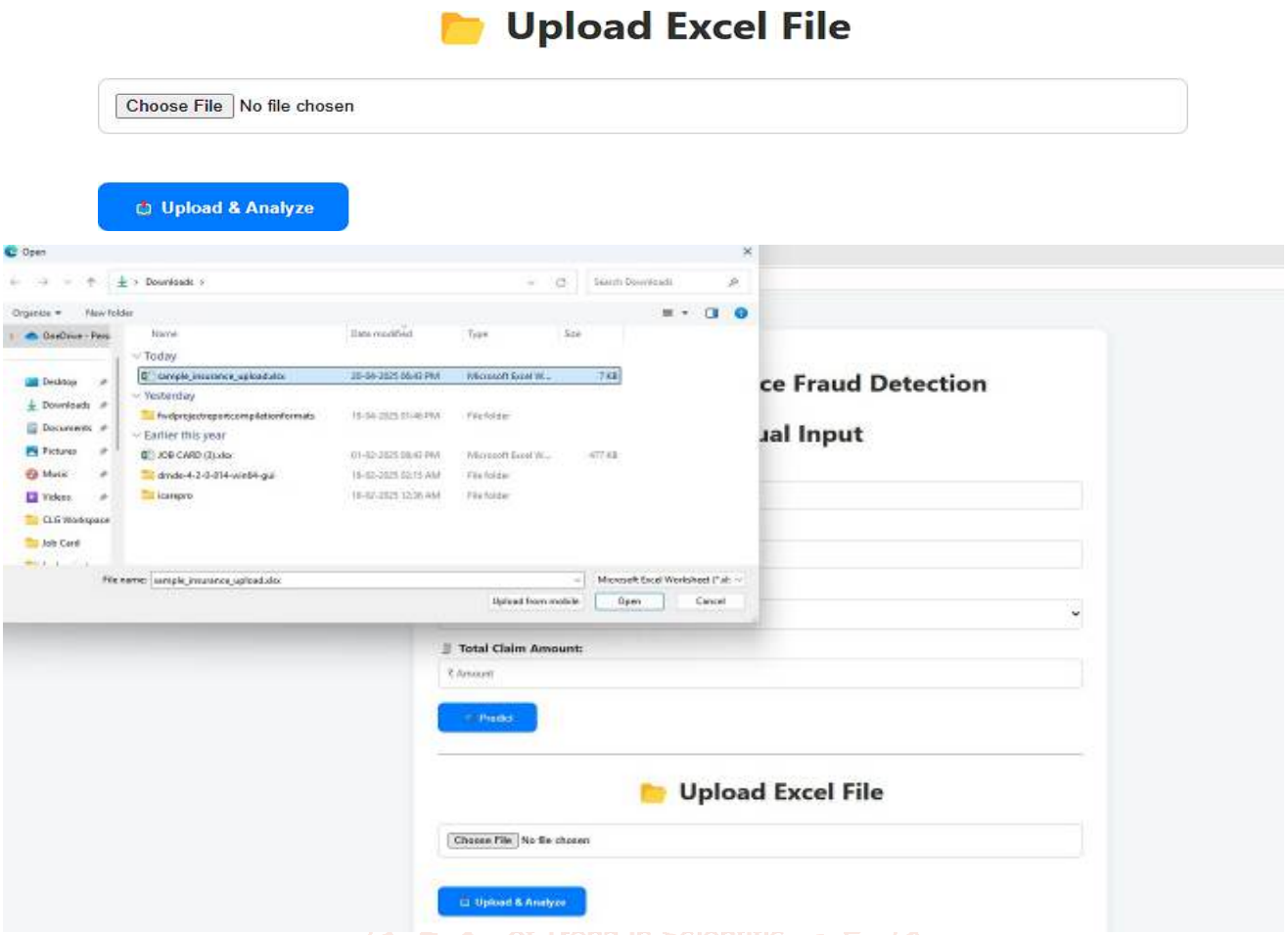
Claim Prediction Model: Predicts whether a claim is likely to be filed.

Amount Prediction Model: Estimates the amount of the claim

2. User Experience & Satisfaction

- Fast – results in seconds.
- Simple – no technical skills required.
- Flexible – single entry or entire file uploads.
- Accurate – supported by deep learning models.
- Actionable – results assist in making actual business decisions.



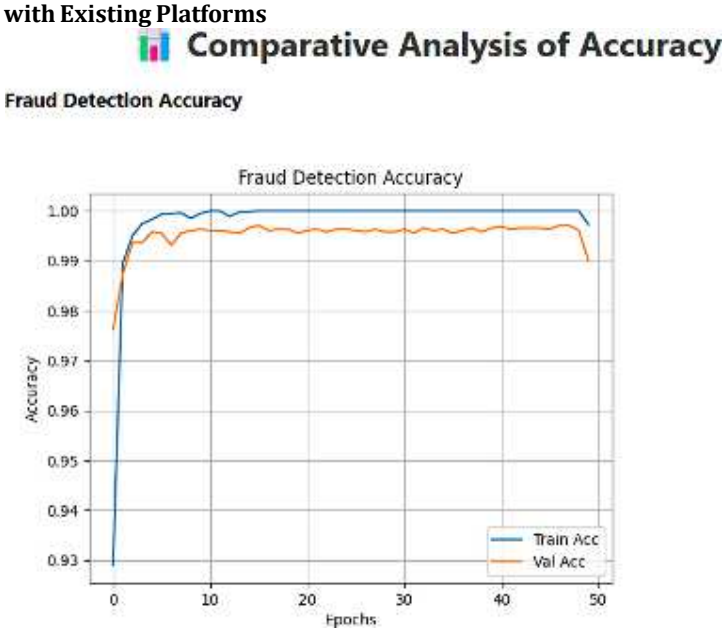


Excel Prediction Results

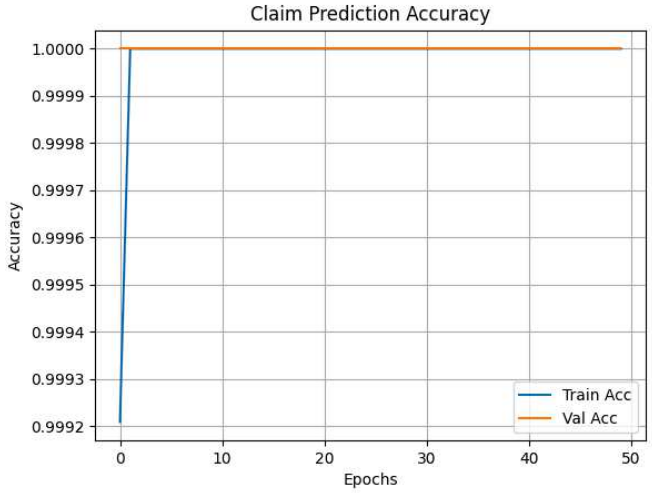
age	policy_annual_premium	incident_severity	total_claim_amount	Fraud_Predicted
30	400	3	5000	No
45	800	3	15000	No
60	1200	3	25000	Yes

Back to Home

3. Comparative Analysis with Existing Platforms



Claim Prediction Accuracy



Comparative Analysis:

The above graphs show the accuracy performance of the Fraud Detection model and the Claim Prediction model. The comparison between these two models helps in understanding which model performs better in terms of prediction accuracy.

Based on the above graphs, you can compare how the models handle the data and how consistent they are in predicting claims and fraud status over different epochs.

4. Challenges & Limitations

While the platform demonstrates significant improvements, a few challenges were observed:

- Data Quality – Actual insurance data could be incomplete or unbalanced (small number of fraud cases).
- Imbalanced Dataset – "Not fraud" might be privileged by models without special treatment.
- Preprocessing Issues – Dealing with missing values and feature encoding properly is delicate.
- Model Complexity – Models based on deep learning are prone to careful tuning and overfitting.
- Web Integration – Creating a seamless, fast interface for manual input as well as bulk input is challenging.
- Explainability – Users or businesses find it difficult to explain deep model decisions.
- Data Security – Sensitive information needs to be treated securely.
- Model Maintenance – Fraud trends evolve, and hence the model must be updated periodically.

5. Future Improvements

The existing system of insurance fraud detection provides a solid base for predictive analysis within the insurance industry; however, there are some areas where improvement is needed. In the future, the model can be enhanced by using more sophisticated machine learning methods like ensemble models, XGBoost, or deep learning to improve accuracy. Real-time fraud detection can be achieved by directly integrating the system with real-time insurance claim processing systems. Moreover, Natural Language Processing (NLP) can be incorporated to scan unstructured data such as claim descriptions and incident reports for more insights. The system can also be enhanced using geolocation and image data to validate claims that involve property or automobile damage. Lastly, implementing a user authentication system and audit trails would ensure the platform is secure and industry-regulatory compliant.

Conclusion

The Insurance Fraud Detection system created during this project successfully illustrates the role of machine learning in forecasting fraudulent claims in the insurance sector. With the use of a combination of classification models and actual real-world insurance data, the system assists in identifying possibly fraudulent behavior, streamlining claim verification processes, and minimizing financial losses.

The project supports two easy-to-use input modes: manual and Excel upload, which makes it suitable for small-scale usage as well as large-scale data analysis. The use of Flask for web deployment and pre-trained models guarantees fast and precise predictions. Appropriate pre-processing of data, encoding, and error handling make the system more robust and user-friendly.

In total, the system offers a secure, scalable, and effective solution to aid insurance companies with early fraud detection, ultimately aiding in operational efficiency and improved risk management

VI. REFERENCES

[1] Duman, E., & Ozcelik, M. H. (2011). *Detecting credit card fraud by genetic algorithm and scatter search.* Expert Systems with Applications, 38(10), 13057–13063.

[2] Farrar, A., Alshamrani, M., & Al-Karaki, J. N. (2017). *Vehicle insurance fraud detection using hidden Markov model.* In *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 1–6.

[3] Sahoo, G., & Panda, M. (2010). *Data mining and machine learning in vehicle insurance claim fraud detection: A survey.* International Journal of Computer Applications, 1(6), 1–6.

[4] El Naqa, I., Yang, C. Y., Wernick, M. N., Galatsanos, N. P., & Nishikawa, R. M. (2002). *A machine learning*

approach for the detection of fraud in the automobile insurance industry.

[5] Proceedings of the SPIE, 4684, 74–83. *(This paper presents an ML approach applied directly to auto insurance fraud.)*

[6] Pham, L. D., & Phung, N. M. (2019). Detecting Fraud in Auto Insurance Claims Using Machine Learning Techniques. 2019 RIVF International Conference on Computing and Communication Technologies (RIVF), 7-12.

[7] Rahamathunnisa, S., & Vadivel, S. (2021). Predicting Fraudulent Claims in Vehicle Insurance Using Machine Learning Techniques. Proceedings of the 2nd International Conference on Innovative Computing and Communication (ICICC), 193-197.

International Journal of Trend in Scientific Research and Development
IJTSRD
International Journal of Trend in Scientific Research and Development
ISSN: 2456-6470