Vehicle Claim Fraud Prediction Using Machine Learning

Carina Chen
Minh DangVu
Chaw Maung
Shrishti Dewangan

Table of Contents

1. Introduction

The United States is a car-dependent country, and automobiles play an important role in life. As the demand for automobiles increases, so does the risk of auto insurance fraud. Auto insurance fraud is defined as the purposeful deception or misrepresentation of an automobile insurance claim or policy with the intent to profit financially. According to the State of Delaware [1], automobile insurance fraud takes two forms: application fraud and filing false claims. Application fraud occurs when an applicant provides false information during the application process, such as providing the incorrect location where the car is regularly garaged, failing to disclose a previous claim or accident, failing to name all eligible drivers, or naming someone who does not drive the vehicle as a principal driver in order to avoid the high premium. According to AAA Insurance Company's article "Steer Clear of Car Insurance Scams" [2], the growing number of auto insurance fraud cases costs companies at least $29 billion every year. The billions of dollars lost by insurance companies have a negative impact on policyholders; 72% of claimed fraud victims [3] believe their auto insurance premiums have increased by an average of $400 to $700 per year [4].

Combating automobile insurance fraud is both vital and challenging. Fraud detection is exceedingly difficult for insurance companies since fraudulent claims are frequently combined with legitimate ones. Soft fraud, such as faking injuries or inflating damage fees, can closely resemble true incidents. The use of counterfeit documentation, such as bills, along with the rise of new AI technologies, deepfakes, allows fraudsters to generate more convincing forgeries to identify papers, necessitating the employment of increasingly sophisticated detection algorithms. Another issue is data overload; with hundreds of car accidents per day, insurance companies must deal with massive amounts of claims data, rendering human judgment unfeasible. That is how scammers exploit gaps in this massive collection of data.

The project aims to create an accurate machine learning model for predicting automobile claim fraud by combining policyholder demographics, vehicle details, and claim-related data. The project's objectives include gathering and analyzing enormous amounts of relational data and features, finding significant patterns of fraudulent behavior, obtaining strong and valid results, and delivering interpretable outputs. The team focuses on developing and testing three machine learning models: Logistic Regression, Random Forest, and Support Vector Machines (SVM). In addition, due to the extremely unbalanced dataset, feature engineering is used in this project by adding three extra categorical features to the original dataset: Property Damage, Incident Severity, and Incident Type. The model performances are compared using stratified 10-fold cross-validation, with evaluation metrics including accuracy, precision, recall, F1-score, and ROC curve. The key finding of this project is that, despite having a low initial accuracy, model performance improved significantly after using the feature engineering technique. The Random Forest model performed the best of the three. The project will assist insurance companies in making informed decisions from a mix of fraudulent and real cases, as well as increase the accuracy of fraud prediction.

2. Related Work

There has been much work done on fraudulent detection as claims are made everyday and everywhere. An early research by Wilson [5] applied a Logistic Regression model to detect fraudulent auto insurance claims. The study focused on claims in both auto physical damage and personal injury protection cases, while also highlighting intentional fraud and opportunistic fraud. With a relatively small sample size of 98 claims, Wilson realized that claims per year and new business policies were the biggest predictors of fraud. The resulting model achieved an overall accuracy of 70.4%, performing better at identifying real claims and fraudulent ones (81.6% vs 59.2%). Despite having a fairly positive result, there are several limitations in the study. One of the key limitations is the small and limited sample size used in the study, leading to the absence of a hold-out or test sample for proper validation. Another limitation is the narrow scope of attributes, only 6 independent attributes, which fails to capture the model’s full capability across real-world scenarios; in which, results in moderate predictive power. The huge performance gap between fraudulent claims (59%) and legitimate claims (82%), while the model achieves an accuracy of 70% further proves

that the model is overfitting and has lower sensitivity to fraud claims. This is not good because identifying fraud is the model's primary goal.
In a more recent study by Jain et al. [6] in 2023 looked into fraudulent car insurance claims and developed a detection and analysis system using Random Forest classifier, as a result of the increase in machine learning and big data. The study explored a much bigger dataset with more than 15,000 records and 33 attributes, before applying preprocessing and feature selection. The model performed significantly high with an accuracy of 99.6%, surpassing baseline classifiers, including decision trees at 93.1% and k-nearest neighbors at 43.0%. The research highlighted the potential for scalable and automated detection systems that reduce loss and maintain efficiency. Similarly to the previously mentioned work, this study also fails to capture the complexity of fraud patterns, which leads to overfitting, as indicated by the unreal high accuracy rate. Another limitation is that the model is designed to rely heavily on the dataset characteristics and is not applicable to a different dataset. This study used the same dataset as our collected dataset, with 33 features, containing 15,000 instances, much less than our dataset with 150,000 instances.

3. Background

3.1. Theoretical foundation

The detection of vehicle insurance claims fraud is modeled as a monitored binary problem where each claim is discussed as a fraud. In this case, the objective of supervised learning is to acquire a function that predicts a binary output (fraud or non-fraud) using claim-related input features, including policyholder information, vehicle description, accident circumstances, amount of the claim, and previous claim history [7]. Data on insurance has become more and more voluminous and complicated, and classic rule-based systems are no longer able to ensure successful detection of fraud, which prompts the use of machine learning approaches based on data.
One of the most basic problems of insurance fraud detection is the issue of a wide imbalance in classes, as the number of fraudulent claims in most cases does not constitute a large portion of all claims. This imbalance has the potential to lead to models being biased in predicting non-fraud cases unless they are trained specifically or measured using specific evaluation metrics [9]. Moreover, the trends of fraud are not usually linear in nature and tend to change with time after fraudsters have developed to counter the detection. Such properties require the use of models that are able to learn non-linear relations that are complex, and have good generalization [6].
On this basis, the machine learning models which include Logistic Regression, Support Vector Machine and Random Forest are widely used in the insurance fraud literature. Logistic Regression is commonly applied as a model that is easy to interpret as a baseline and SVM as a model that can incorporate high-dimensional data with a margin-based framework that can predict well, and Random Forest as an ensemble model that can predict better and be more robust. Several empirical studies of high-quality auto-insurance and property insurance data have shown that these models are competitive and even better than conventional statistical methods[7][4][6].

3.2. Algorithms and Foundations of Mathematics

The Logistic Regression is a probabilistic classification model that approximates the conditional probability that a claim is a fraud. Given an input feature vector x_i, the model sees how likely the probability of fraud is as a result of the logistic sigmoid function.

$$P(y = 1|x) = 1/1 + e^{-z},$$

where

$$z = w_0 + \sum_{i=1}^{n} w_i \; x_i$$

In this case, w_0 is the intercept and w_i are the learnt feature weights. Minimization of the binary cross-entropy loss is used to estimate the model parameters.

$$L = -1/N \sum_{j=1}^{N} [y_j \log(p_j) + (1 - y_j)\log(1 - p_j)],$$

y_i is the actual label of the classes and p_j is the probability predicted. Logistic Regression is a preferred model when it comes to detecting fraud because it is simple, quick to compute, and highly interpretable because the acquired coefficients measure the impact of each factor on the fraud probability [4]. Nevertheless, it is constrained by the fact that its assumption of linear separability in the log-odds space is unable to explain complex fraud behaviors that are due to nonlinear feature interactions [7].

SVM is a learning algorithm that runs on the margin; it is used to locate the best separating plane between fraudulent and valid claims. In linear model, the decision boundary would be specified by

$$w.x + b = 0,$$

w is the weight vector, and b is the bias. The optimization problem aims at minimizing the norm of weight vector such that the margin between the two classes is maximized and the constraint.

$$y_i(w.x_i + b) \geq 1.$$

In case of non-linearly separable data SVM can use some kernel functions to project the original input space into a higher-dimensional feature space. The most common one is the Radial Basis Function kernel which can be defined as.

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)$$

This allows SVM to build non-linear models. SVM has also been shown to perform well with high-dimensional datasets in insurance fraud detection when both fraudulent and legitimate claims are close enough to each other [4][9]. Although SVM has a high predictive power, it is computationally expensive for large datasets and is vulnerable to hyperparameter and kernel tuning.

Random Forest is a type of ensemble learning algorithm, which builds many decision trees on bootstrap samples of the training set and combines their prediction with majority voting. Assuming $\hat{y}^{(t)}$ to be the prediction of the t-th tree, the overall prediction is obtained as

$$\hat{y} = mode(\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots\ldots\ldots\ldots, \hat{y}^{(T)})$$

Only a random subset of features is considered to split at each decision node and this allows the reduction of correlation between individual trees and also the enhancement of generalization. The splitting of trees is usually chosen using a Gini impurity.

$$G = 1 - \sum p_i^2.$$

Random Forest models are especially effective in insurance fraud detection in that they automatically find non-linear relationships between features, are resistant to noisy and incomplete data, and are far less subject to overfitting than single decision trees. The empirical results on the real-life auto-insurance data indicate that Random Forest is the most effective and robust among the Logistic Regression and various other classifiers in terms of detection rates and strengths [7][6].
The other benefit of the Random Forest is that feature importance measures are available and help an investigator determine the most contributing factors in predicting fraud.

4. Methodology

The following diagram illustrates a five-phase workflow for developing our machine learning system and explains the major processes involved in each phase. We iteratively performed Stages 3 through 5 during our experiments and evaluated the results at each stage. However, not all processes in Stage 3 need to be repeated; some, such as feature creation and data splitting, are implemented with alternative values to train different models, compare accuracies, and evaluate the models.

In Stage 4, we trained three models: Logistic Regression, Support Vector Machine, and Random Forest. Model accuracy was then evaluated using a standard confusion matrix and receiver operating characteristic (ROC) curve.

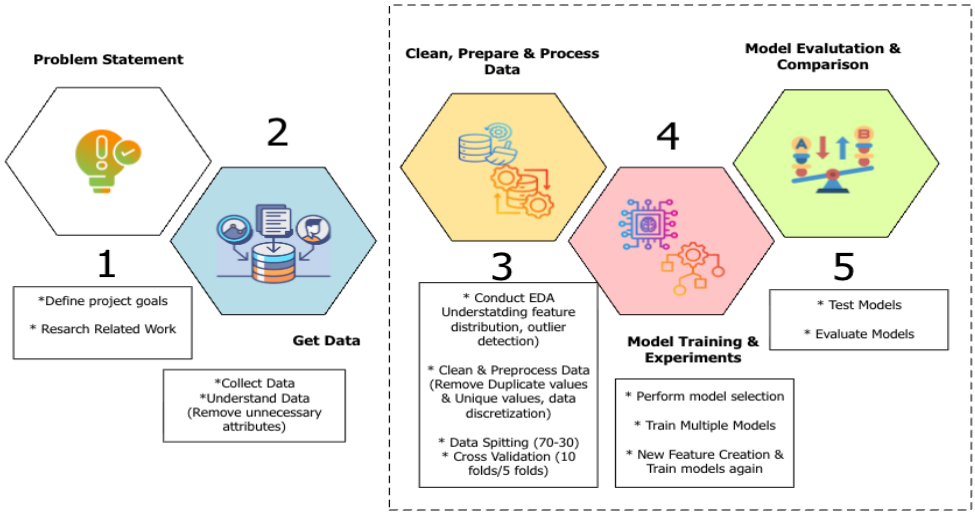


Fig 1. Work Flow Diagram of Methodology

5. Description of Data

Dataset: Kaggle dataset for vehicle claim fraud detection

The dataset "Insurance policy and fraud prediction" originated on Kaggle and contains approximately 150,000 data samples with 34 features relating to vehicle insurance policies and their characteristics. It was published by Omega SemmalaiCit - Owner and it was again updated in late 2024.
The features include demographic information about policyholders such as Age of Policy Holder, Gender, Marital Status, Change of Address, policy-related information such as Policy Type, Past Number of Claims, Fault Indicator, Number of Policy Purchased and Accident, Numbers of Cars involved in the

Make	AccidentArea	Fault	PolicyType	VehicleCategory	VehiclePrice	ClaimAmount	...	Deductible	PoliceReportFiled	WitnessPresent	AgentType	NumberOfSuppliments	AddressChange-Claim	NumberOfCars	Year
Nissan	Suburban	Third Party	Collision	Sedan	6.308436e+05	20000	...	400	No	Yes	Internal	0	no change	2	2012
Honda	Suburban	Third Party	Collision	Utility	1.243086e+06	20000	...	400	No	No	External	1	4 to 8 years	3	2013
Nissan	Urban	Third Party	Collision	Utility	1.206393e+06	20000	...	300	No	No	Internal	0	1 year	3	1994
Nissan	Urban	Third Party	Collision	Sport	1.291825e+06	20000	...	300	Yes	No	Internal	1	4 to 8 years	1	1999
Toyota	Urban	Policy Holder	Liability	Sedan	7.123651e+05	20000	...	300	Yes	No	External	2	no change	2	2000

Fig 2. Overview of Dataset

accident and vehicle-related information, e.g., Vehicle Price, Vehicle Category, Make of Vehicle to predict fraud. The dataset contains both numerical and categorical features. As shown in Fig. 3, there is a fairly unbalanced distribution of values for the target, with 104,903 - (Non-Fraud) and 45,097 - (Fraud), indicating the fraud record is 30% of the overall data samples.

FraudFound	count
No	104903
Yes	45097

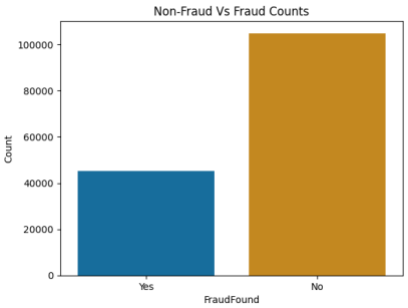


Fig 3. Non-Fraud vs Fraud

Our dataset is considerably large, so we decided to use a 70:30 hold-out approach. The team performed exploratory data analysis and preprocessing. First, we checked for null-value columns; there were none. Second, we continued identifying duplicate records and discovered that Age and Age of Policy Holder are duplicates; we removed one of them. The team checked ID attributes, such as Police Number and Rep Number, which had a large number of unique values and were unnecessary for model training. Both of them are discarded. Aside from that, the claim amount was eliminated because it had all fixed numbers. In total, we dropped 10 of 33 attributes, keeping 23 features to develop the model.

```
unique=df.nunique().to_frame('unique')
unique.columns=['count']
unique.index.name='columns'
unique.reset_index(inplace=True)
unique.sort_values(by='count',ascending=False)
```

	columns	count
11	VehiclePrice	149988
18	AgeOfPolicyHolder	50
25	Year	34
0	Month	12
17	AgeOfVehicle	10
2	DayOfWeek	7
16	PastNumberOfClaims	6
22	NumberOfSuppliments	6
13	DriverRating	5

Fig 4. Unique Features

Interestingly, we analyzed the Vehicle Prices as all are unique, yet some of them are rather close to each other. This can make model training challenging and produce inaccurate results. So, we did round up the discretization and divide them into distinct ranges. For example, the Vehicle Price of 630843.615757045 will be rounded to 600,000 in Fig. 5.

```
df['VehiclePrice'] = df['VehiclePrice'].round(decimals=-5)
print(df['VehiclePrice'].unique())
```

```
[ 600000. 1200000. 1300000.  700000. 1100000. 1000000. 1500000.  800000.
 1800000.  900000. 1400000. 1600000. 1700000.  500000.]
```

Fig 5. Vehicle Price Discretization

Similarly, the Age of Policy Holders column contains 50 unique values; therefore, we decided to apply binning discretization, grouping into five smaller age categories as shown in Fig. 6, after discussion with the professor. It will help reduce model skewness and achieve a more uniform distribution.

```
bins = [15, 25, 35, 45, 55, 65]
labels = ['16-25', '26-35', '36-45', '46-55', '56-65']
df['AgeOfPolicyHolder'] = pd.cut(df['AgeOfPolicyHolder'], bins=bins, labels=labels, right=True)

print(df['AgeOfPolicyHolder'])
```

```
0         26-35
1         26-35
2         16-25
3         46-55
4         16-25
          ...
149995    16-25
149996    56-65
149997    26-35
149998    36-45
149999    16-25
Name: AgeOfPolicyHolder, Length: 150000, dtype: category
Categories (5, object): ['16-25' < '26-35' < '36-45' < '46-55' < '56-65']
```

Fig 6. Age Binning

When moving into the feature correlation study, we have discovered that most of the attributes have a relatively uniform distribution, so it is not possible to rely on individual attributes to predict vehicle fraud detection. According to the correlation matrix, sports vehicles have a higher fraud value which leads to a higher correlation to the target value. Moreover, the analysis of the Address Change feature exhibits a slightly elevated fraud rate (34% vs 30%); the difference is marginal and lacks a significant relationship. Address change alone is not a strong predictor, but it may be valuable when combined with other features like Claim Timing or Vehicle Value.

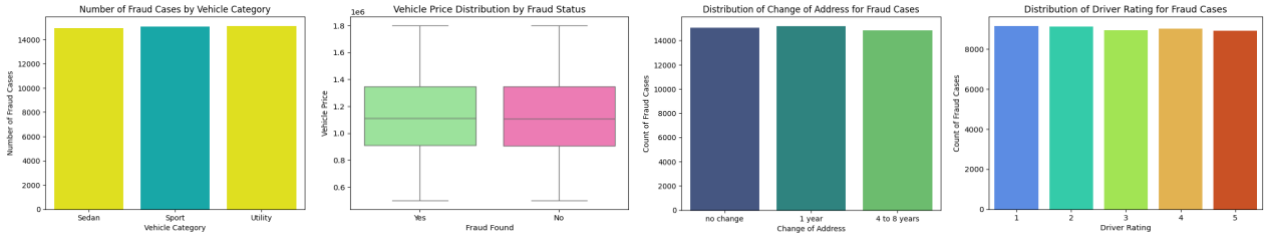


Fig 7. Feature Analysis

Additionally, the following correlation matrix reveals that there are only notable relationships between Vehicle Category and Vehicle Price is 0.33. Most features show very weak correlation (even close to 0.00) to fraud prediction output (Fraud Found). For example, claim-related features (Past Number of Claims, Number of Cars involved) don't strongly correlate with accident details. The original dataset showed no meaningful correlation between input and target variables. We addressed this issue through feature engineering, applying domain knowledge to create new features. Adding too many features may increase model complexity and hurt model performance so we started with a simple approach and selected three highly correlated features: Incident Severity,

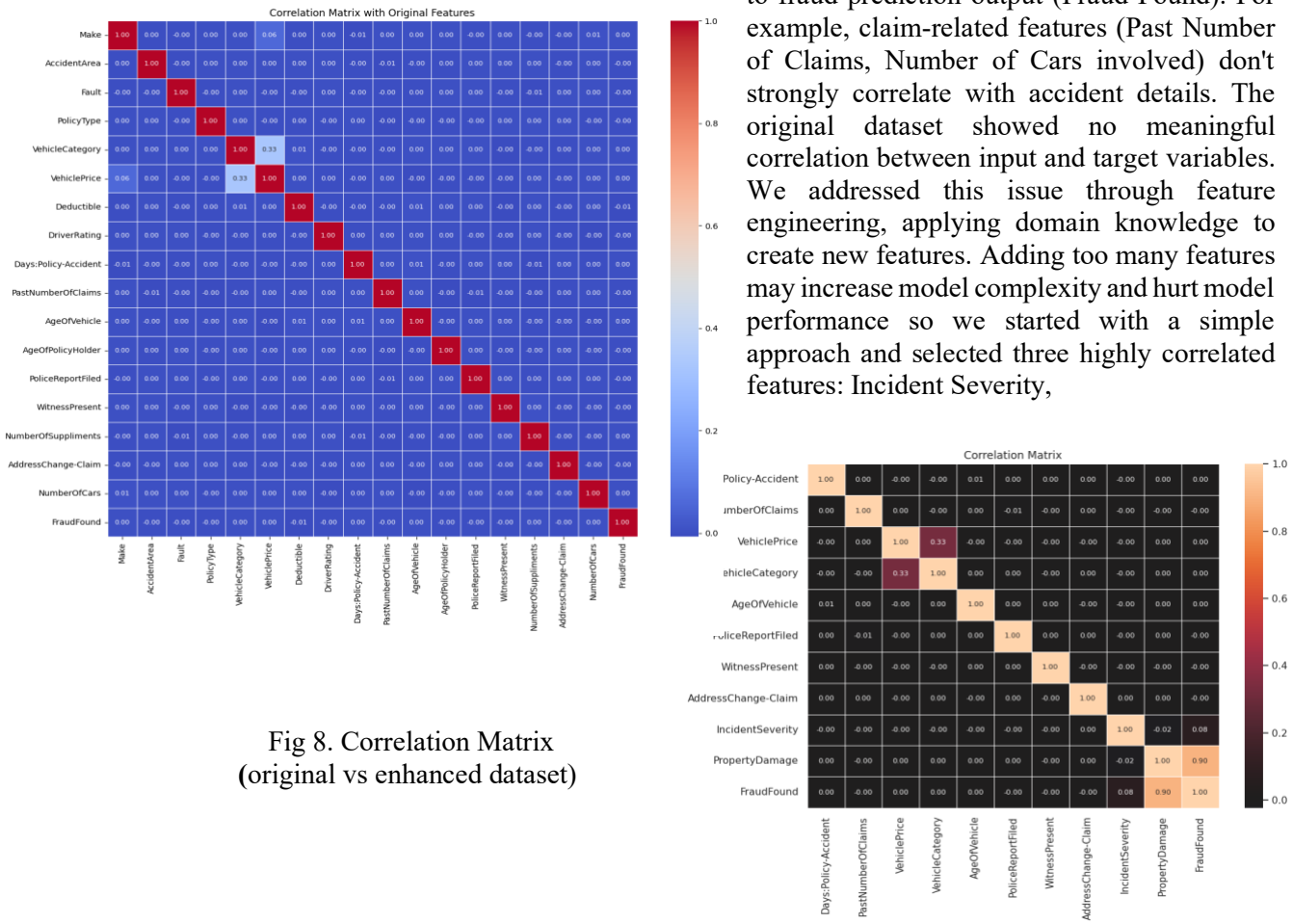


Fig 8. Correlation Matrix
(original vs enhanced dataset)

Incident Type and Property Damage. These features demonstrate a strong correlation with our target variable - fraud found as shown in Fig. 8.

6. Experimental details

The aim of the experiment was to evaluate how feature engineering and cross-validation impact fraud detection performance. The model was trained on structured vehicle insurance claim data comprising policyholder attributes, vehicle characteristics, accident-related data, and past claim behavior. The target variable was binary: fraudulent (1) or legitimate (0). Initially, the experiments were conducted with the original dataset. Then, feature engineering was performed with three additional features based on domain knowledge of insurance fraud patterns. These new features significantly improved the model's predictive performance.

Three algorithms were selected for the experiments. Logistic Regression served as the baseline model due to its computational efficiency and interpretable relationship between input features and the probability of fraud. Support Vector Machine was chosen for its ability to generalize well with high-dimensional data while avoiding overfitting. Random Forest was selected for its strong performance with non-linear relationships, robustness to noisy data, and the ability to model complex feature interactions commonly found in insurance claim datasets.

6.1. Experiment Setup

1. Logistic Regression

Initially, the dataset was split into a 70:30 training/testing set, which resulted in 46% accuracy. This indicates that the model's performance is not good as it wrongly predicted 54% of the dataset, so a stratified 10-fold cross-validation algorithm is used. A Logistic Regression model is trained with L2 regularization to prevent overfitting, and the class weight parameter is set to balance to address the imbalance between the fraudulent and legitimate claim class in the dataset. To ensure that the optimization algorithm converges, a maximum number of iterations is set to 1000. The model's performance is evaluated using stratified 10-fold cross-validation, to maintain the similar distribution of class values in each fold, avoiding bias in predictions. The dataset is shuffled before splitting to ensure that no order is introduced. Cross-validation is used to provide unbiased predictions for all samples. The performance metrics for this model include accuracy, precision, recall, F1-score and confusion matrix. These metrics capture different aspects of the classification quality.

The training and testing itinerary remains the same when using the dataset with feature engineering. With four additional features, the enhanced dataset is split using stratified 10-fold cross-validation to ensure consistency. These new features allow the model to capture underlying relationships in the data better. L2 regularization is once again used to reduce overfitting.

2. Support Vector Machines (SVM)

Similarly, in the first part of the experiment, SVM training started with a 70:30 hold-out approach for the original dataset. SVM is primarily designed to work with numerical data; therefore, a one-hot encoding method is used to transform all categorical data such as Policy Type, Vehicle Category, Incident Severity and StandardScalar function is applied to standardize numerical features such as Vehicle Price, Number of Past Claims before feeding into the model to ensure all features are fairly contributed to training data and preventing data leakage between train and test sets.

Since there is a larger number of training samples, a Linear SVC classifier was trained to find the optimal hyperplane to separate fraud vs non-fraud classes. In order to mitigate class imbalance, the class weight was balanced by shifting the decision boundary to the minority (fraud) class and improving its recall and F1-score. An SVM hold-out approach gives overall 50% accuracy with the original dataset; therefore,

moving forward, the model's performance is evaluated using stratified 10-fold cross-validation, to maintain a similar distribution of class values in each fold, avoiding bias in predictions.
Interestingly, SVM classifier provides nearly the same accuracy for both hold-out and stratified 10-fold cross-validation with the original dataset. The training and testing itinerary remains the same with an enhanced dataset after adding three meaningful and more relevant features for vehicle fraud prediction.
Although with these new features, the model shows improvement, other non-linear approaches, such as Kernel SVM with C hyperparameters, were planned to test during the experiment.

3. Random Forest

A hold-out validation plan was used on the original dataset in the first step of experimentation and then feature engineering was done. The data were divided into training and test parts in which the model was trained on the training set and tested on the unknown test set. Based on this baseline dataset, the Random Forest model was also able to attain an accuracy of around 70%, which means that predictive performance was moderate and that improvement was still necessary, either by the addition of feature engineering.
The second phase of experimentation involved three novel engineered features added to the dataset in order to improve the behavior patterns of fraudsters. The Random Forest model was retrained after feature augmentation and the same hold-out validation strategy was used. The accuracy of the model increased significantly with the supplemented feature set to about 95% proving that domain-based feature engineering was critical in the enhancement of the performance of the fraud detection.
To further guarantee the strength and external validity of the model, stratified 10-fold cross-validation was used on the final feature-engineered dataset. To maintain the original class distribution of the fraudulent and non-fraudulent claims in all folds, stratification was applied. In this validation process, the dataset was divided into ten equal subsets and in each run, 9 folds were employed in training and one fold in testing. To have a consistent and unbiased measure of performance of the models, the performance measures were averaged over all 10 folds.
The stratified 10-fold cross-validation results proved the good generalization potential of the Random Forest classifier, which still provided the overall accuracy near 95%. This showed that it was not the data leakage or overfitting that increased the performance, but rather it was the meaningful feature engineering and good model training that did it.

6.2. Report Experimental Results and Analysis

1. Logistic Regression

This Logistic Regression Model achieved an accuracy of 50%, precision of 0.30, recall of .496, and an F1-score of 0.37 with stratified 10-fold cross-validation on the original dataset. It classifies 75,903 claims correctly, while wrongly classifying 74,097 claims out of 150,000 claims. The confusion matrix reveals that the model correctly classifies 52,591 legitimate claims (true positive), and 22,382 fraudulent claims are misclassified as legitimate (false positive). However, it also misclassifies 51,982 legitimate claims as fraudulent (false negative) and only correctly classifies 22,553 fraudulent claims (true negative). This shows that while the model successfully identifies about half of the fraudulent claims, it also gives an equal amount of false alarms, reflecting the recall of 0.49 and precision of 0.30.
During the second experiment with the enhanced dataset, the model reached a performance with 93.32% accuracy, 0.87 precision, 0.98 recall and 0.93 F1-score with stratified 10-fold cross-validation. Out of 150,000 claims, it successfully classified 142,983 claims, leaving 7017 incorrectly classified. The confusion matrix shows that the model correctly classified 98,387 legitimate claims with only 501 misclassified (false positive), and correctly identified 44,596 claims as fraudulent with 6,516 misclassified fraudulent claims (false negative). This experiment with the enhanced dataset shows a jump in performance, especially when the predictions were nearly random with the original dataset.

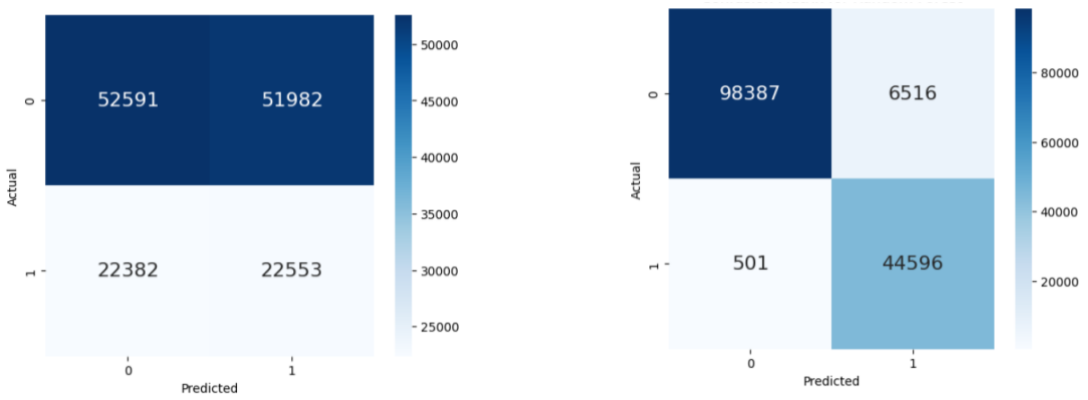


Fig 9. Logistic Regression Confusion Matrix Comparison (original vs enhanced dataset)

2. Support Vector Machines (SVM)

Our SVM model achieved an accuracy of 49%, precision of 0.299, recall of 0.505, and an F1-score of 0.376 with stratified 10-fold cross-validation on the original dataset. The model correctly classifies approximately 50% of the total cases, which indicates the model failed to learn the meaningful patterns from the original dataset. With precision is about 30%, which means that when the model makes a positive prediction, 7 out of 10 predictions are incorrect (false positives). The model identifies only half of the actual fraud cases while missing the other half for detection. Therefore, the model's overall performance is weak, no better than random guessing.

After feature engineering with three domain-related attributes, such as Incident Severity, Incident Type, and Property Damage, the second experiment was conducted using the enhanced dataset. The results revealed dramatic improvement, with accuracy increasing to 91.57%, precision of 0.841, recall of 0.951, and F1-score of 0.895 with stratified 10-fold cross-validation. This demonstrates that the domain-specific features are highly predictive of the target label.

All experiments were conducted using the Linear SVC classifier. Although additional experiments have been planned to continue Kernel SVM with different C hyperparameters, the training time was extremely long, and the team decided to discontinue training, which is impractical with our larger dataset and 10-fold cross-validation within the available time constraints.

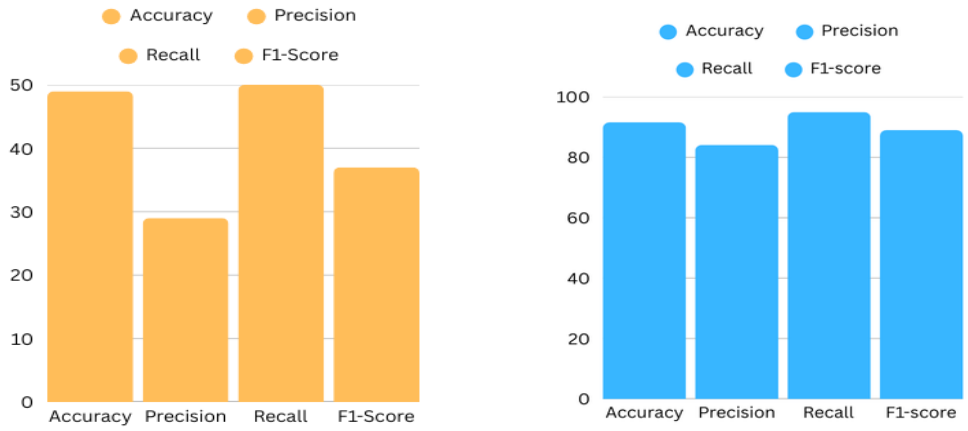


Fig 10. Support Vector Machine Performance Matrix Comparison (original vs enhanced dataset)

3. Random Forest

Random Forest model obtained an overall average accuracy of 69.935%. The mean precision, recall, and F1-score were noted as 0.8734, 0.9871 and 0.9268, respectively. High recall value shows that the model was very effective at identifying the majority class whereas the high F1-score shows that the model was well balanced in terms of precision and recall. Nevertheless, the total confusion table reveals that all the predictions favored the legitimate category and there were no cases of fraudsters predicted. It confirms that despite the overall accuracy that seems to be reasonable, the model continues to be imperfect in the area of effective minority class (fraud) identification because of extremely high class imbalance.

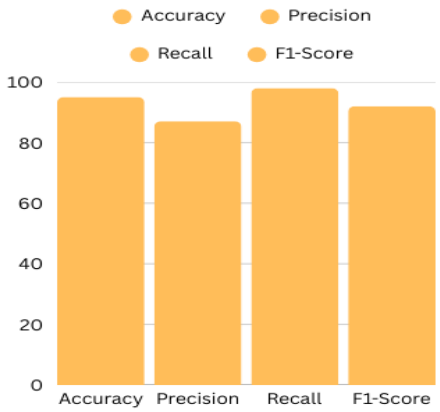


Fig 11. Performance metrics of Random Forest with original dataset

6.3. Model Comparison

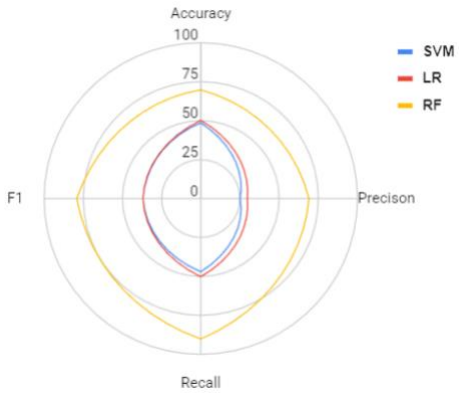
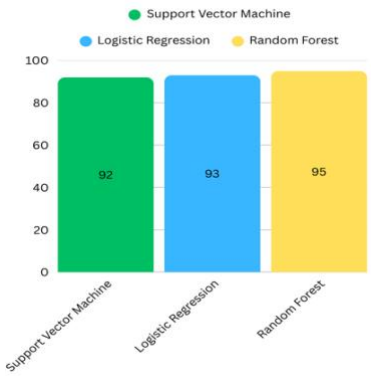


Fig 12: SVM, LG, RF accuracy, precision, recall and F1-score

Fig. 12 compares the performance of all three models with the original dataset. Random Forest's performance on the original dataset is the best out of the three. It performed better on all four of the evaluation metrics, especially with recall of 0.9. This shows that the Random Forest algorithm is good at identifying legitimate claims that satisfied our goal for this project.



When we experimented with the enhanced dataset, the overall performance of all models improved, but Random Forest continued to achieve the highest accuracy among the models. Its accuracy, at 95%, remained the strongest, reinforcing that it is the most effective model for our tasks, both before and after data enhancement.

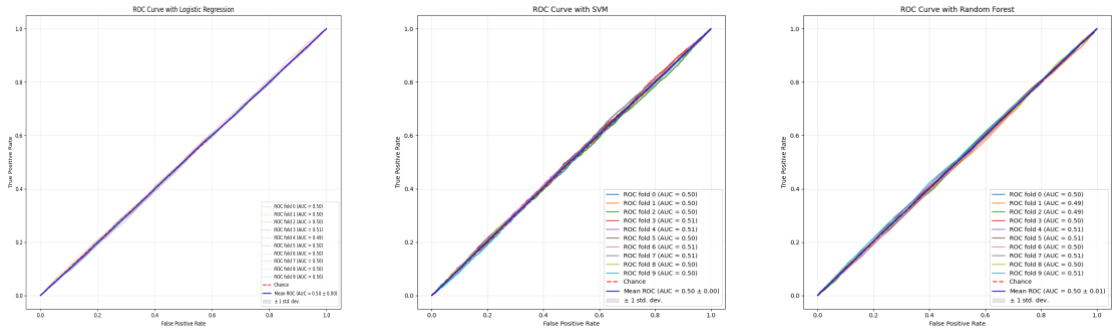


Fig 13. ROC curves of all three models with the original dataset

All three of the ROC curves with the original dataset follow the same upward trajectory across the graph, indicating that they achieve similarly strong TP and FP trade-offs. This suggests that despite the differences in accuracy and other evaluation metrics, they all exhibit similar behavior in terms of ROC performance.

6.4. Best Performing Model

Random Forest was found as the most successful model in vehicle insurance claim fraud detection with both the original dataset and the improved dataset with newly developed features and was always superior to Support Vector Machine and Logistic Regression models in their accuracy, consistency, and predictability strength. This is mainly because Random Forest can model complex non-linear correlations that are involved in the patterns of fraud, which Logistic Regression cannot model as it is linear and SVM cannot model as it is expensive to compute on large data with high dimensions. Also, to a great extent, Random Forest is resistant to noise and overfitting since it integrates several decision trees with the help of ensemble learning and random selection of features. It also manages imbalanced data much better, having a better recall of fraudulent cases than compared to the other models which tend to be biased to the majority class. Moreover, the Random Forest will automatically detect the most informative features in both the original and engineered variables, which in turn, lets this classification technique take full advantage of fraud-specific patterns, including the time of claims, policy risk, and previous claim behavior, and will unquestionably make it more reliable and accurate in this study.

7. Discussion of Our Experiments

7.1. Contributions of Our Work

Carina contributes to the team by participating in data collecting, building and training SVM models with Chaw, and documenting. Chaw carried out data collection, exploratory data analysis (EDA), feature engineering, and SVM training. Shrishti developed and trained a Random Forest model, researched and compiled algorithm theories, and interpreted the model outputs. Minh's responsibilities include training and developing a Logistic Regression model, gathering previous related studies, and arranging the presentation slides. At the end of the semester, everyone took part in the project presentation. Following the workflow discussed in section 4, we scheduled a checkpoint at each milestone to track our progress and to meet the project goal.

7.2. Limitations of Our Work

There are several constraints that limited our project performance and analysis. First, the dataset is highly imbalanced, with significantly higher legitimate claims than fraud claims, making it difficult for the models to learn minority class patterns and increasing overfitting. Although class balancing techniques were used

during training, the imbalance still limits overall performance. Second, the time required for training on some models is too long, which restricts us from exploring other training methods, such as 80:20 hold-out or repeated hold-out. Particularly, the SVM model was computationally complex and took a very long time to complete a single cross-validation fold. As a result, not all available training methods or parameter configurations could be explored. Another limitation is that, due to the overall time constraint, we were unable to perform an exhaustive search of alternative algorithms, parameter tuning methods and other feature engineering approaches that may have further improved the model's performance.

7.3. Possible Future Works

In future work, we aim to refine our feature engineering techniques to improve predictive performance for weakly correlated features. We plan to investigate feature interactions by combining features, such as Vehicle Price, Past Number of Claims, and Claim Amount. Furthermore, we will collect additional domain-specific features, including Total Claim Amount, Injury Claim, Time of the Accident, Insured Hobbies, Insured Occupation, and geographical area, to deepen our understanding of the domain. Additionally, we will apply NLP analysis of claim descriptions to extract sentiment, complexity, and inconsistency patterns. Finally, we intend to use tree-based models (Random Forest, XGBoost) to validate our feature selection and assess multicollinearity among the selected features.

For model development, we will implement advanced modeling techniques that capture non-linear patterns and feature interactions. Specifically, we will explore XGBoost for gradient-boosting capabilities and Neural Networks, which can learn complex, non-linear relationships to capture diverse patterns. To address the class imbalance in fraud records, we will evaluate SMOTE and cost-sensitive classification methods to enhance the model's predictive accuracy and operational efficiency.

8. Conclusion

As the number of automobile insurance fraud cases increased, this project developed a vehicle claim fraud prediction system utilizing machine learning to assist insurance companies in making data-driven decisions and better spotting suspicious fraud patterns. In total, three types of machine learning algorithms were developed and trained: Logistic Regression, Random Forest, and SVM. The correlations between each feature were exceptionally low throughout the data preprocessing phase due to the dataset's major imbalance. The team resolved this issue by including three new features in the dataset. The model's performance was evaluated using the stratified 10-fold cross-validation approach; Support Vector Machines scored 50% accuracy, Logistic Regression scored 54%, and Random Forest performed best with 70% accuracy. After applying the feature engineering technique, all three models achieved greater than 90% accuracy.

The team also evaluated Random Forest using the updated dataset in terms of other metrics; as a result, precision increased from 0.69 to 0.87, recall increased from 0.9 to 0.99, and F1-score increased from 0.79 to 0.93. These findings show that feature engineering significantly boosted feature correlations and overall model performance. In the future, the team plans to incorporate hyperparameter fine-tuning into the training process to generate a more accurate evaluation of performance and reduce overfitting. Furthermore, the team will explore advanced feature engineering techniques such as feature interactions, NLP-based analysis of claim descriptions and advanced modelling algorithms such as XGBoost and Deep Learning could provide further improvements in fraud prediction accuracy.

9. References

[1] "Auto Insurance Fraud - Delaware Department of Insurance - State of Delaware," *Delaware Department of Insurance - State of Delaware*, May 14, 2021. https://insurance.delaware.gov/reportfraud/auto-insurance/

[2] A. Staff, "Steer Clear of Car Insurance Scams," *Your AAA Network*, Feb. 20, 2024. https://magazine.northeast.aaa.com/daily/insurance/auto-insurance/auto-insurance-fraud-affect-everyone/

[3] A. Hurst, "Despite Blind Spots About Insurance Fraud, Nearly 1 in 3 People Believe They've Been a Victim," *ValuePenguin*, Aug. 23, 2021. https://www.valuepenguin.com/one-third-of-consumers-believe-they-have-been-insurance-fraud-victims

[4] Alrais, A. I. (2022). *Fraudulent insurance claims detection using machine learning* (Master's thesis). Rochester Institute of Technology. https://repository.rit.edu

[5] "CoverHound® | Insurance Learning Center | Distinguishing Between Hard And Soft Insurance Fraud," *www.coverhound.com*. https://www.coverhound.com/insurance-learning-center/distinguishing-between-hard-and-soft-insurance-fraud

[6] Khalil, A. A., Subudhi, S. P., & Panigrahi, B. K. (2025). Enhancing insurance fraud detection accuracy with machine learning: Automobile insurance claims case study. *Journal of Economic Dynamics and Control*.

[7] Severino, M. K., & Peng, Y. (2021). Machine learning algorithms for fraud prediction in property insurance: Empirical evidence using real-world microdata. *Machine Learning with Applications, 5*, 100074. https://doi.org/10.1016/j.mlwa.2021.100074

[8] Srijit Panja. (2022, November 3). *Step by step Data Science execution — Car Insurance Fraud Detection Task (Example)*. Medium. https://medium.com/@srijitpanja/step-by-step-data-science-execution-car-insurance-fraud-detection-task-example-9855d306a4c9

[9] Wang, C., Nie, C., & Liu, Y. (2025). Evaluating supervised learning models for fraud detection: A comparative study of classical and deep architectures on imbalanced transaction data. *arXiv Preprint*.

[10] Zhang, Z. (2024). *UNIVERSITY OF CALIFORNIA Los Angeles Fraud Detection in Vehicle Insurance Claims using Machine Learning*. https://escholarship.org/content/qt0jx1h48j/qt0jx1h48j_noSplash_84e3c87ba8f0762a50da53fd36439a93.pdf?t=secjt7

[11] scikit-learn, "1.4. Support Vector Machines," *Scikit-learn.org*, 2018. https://scikit-learn.org/stable/modules/svm.html

[12] A. Navlani, "Scikit-learn SVM Tutorial with Python (Support Vector Machines)," *www.datacamp.com*, Dec. 2019. https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python