Automobile Insurance Claim Fraud Detection using Random Forest and ADASYN

Dhruvang Gondalia[1], Omkar Gurav[2], Ameya Joshi[3], Aniruddha Joshi[4], Prof. Sangeetha Selvan[5]

[1,2,3,4]UG Student, Dept. of Computer Engineering, Pillai College of Engineering, New Panvel, India
[5]Assistant Professor, Dept. of Computer Engineering, Pillai College of Engineering, New Panvel, India

---***---

Abstract - *With the increasing number of fraudulent claims in the insurance industry, this issue needs to be contained. Car insurance fraud is the most common compared to all other types of fraudulent claims. Therefore, it is necessary to have a system to detect and prevent such fraud, and it is necessary to build a system to detect insurance fraud. Many fraud detection models are created using a variety of algorithms and techniques. We used a random forest as a classifier and ADASYN to balance the dataset. One Hot Encoding was used to resolve an issue of undesirable attributes during balancing the dataset. This application we created can be used by car insurers to evaluate customer claims more quickly than other traditional methods that involve manual tasks. Therefore, this application helps find out if the claim is genuine or fraud while the customer is claiming insurance. It is more accurate and free of fraud than traditional methods. Other techniques such as SVM can be used, but for this particular problem, Random Forest seems ideal because it provides significantly better accuracy than other techniques.*

Key Words: **ADASYN, SVM, Random Forest, Data Sampling, Insurance Fraud, Fraud Detection, One Hot Encoding**

1. INTRODUCTION

Insurance fraud occurs when an insurance provider, advisor, adjuster, or consumer intentionally deceives in order to obtain an illegal gain. There has been an increase in fraudulent insurance claims in recent years, particularly in the automobile insurance industry. Falsify insurance claim information, exaggerate insurance claims to represent an accident, or submit a claim form for damage or injury that has never occurred by making a false claim for car theft. That's all an example of a car insurance fraud. When insurance companies use fraud detection systems, they not only detect fraud but also save millions, if not billions, of dollars that would otherwise be paid to the person who made the fraudulent claim.

2. LITERATURE REVIEW

i. Detecting Fraudulent Insurance Claims Using Random Forests and Synthetic Minority Oversampling Technique: The author used SMOTE to balance the dataset and used Random Forest for the prediction of the claim, So SMOTE with random forest gives accuracy upto 94%. But it can be improved by using other balancing techniques like ADASYN which is grouped under over sampling technique data balancing technique.

ii. Performance comparative study of machine learning algorithms for automobile insurance fraud detection: The author showed a study comparing ten of the most frequently used machine learning algorithms for detecting fraud in insurance claims. The study shows that the Random Forest algorithm has the best performance for insurance fraud detection.

iii. Detecting Fraudulent Motor Insurance Claims Using Support Vector Machines with Adaptive Synthetic Sampling Method: They have used ADASYN for balancing the dataset where it tries to increase minority class samples by adding similar entries in it. Base model used in this project was SVM but the dataset used in it consists of only 1000 rows out of which 25% of the data consists of fraudulent claim and rest were genuine claim.

iv. Automobile Insurance Fraud Detection using Supervised Classifiers: The dataset used in this project is not available on internet the dataset consists of 11 different columns such as Gender of Policyholder, Police Report File ,Model of Car etc So for balancing the dataset the author used SMOTE to balance it and tested dataset with 3 different classifier they are Multi-Layer perceptron, Decision tree, and Random forest, Author found that Random forest is best technique for this problem statement.

v. Fraud Detection by Machine Learning: Here the author discusses different types of credit card frauds. He proposed the dataset should be in 1:1 ratio for fraud and genuine cases. And he tested different machine learning algo such as logistic regression, support vector machine, boosted trees, random forest, and neural network etc. and found random forest to be the best fit algorithm for his dataset.

3. Dataset and Parameters

The experimental dataset used in this study is provided by the user Jwilda on kaggle[6]. The dataset has 15,420 rows with 33 columns of data. Each row in the dataset has 33 attributes in total. Out of which, 32 are claim features that will help to predict the last 1 variable, called the class label. Here, FraudFound is our target variable which will contain a value, either ‘1’ or ‘0’. This variable represents whether the claim is genuine or fraud. ‘1’ would mean the claim is fraud and value ‘0’ represents a genuine claim. Here 25 out of 32 claim features are categorical and remaining 7 features are numerical. Out of 15,420 rows, 14,497 rows consist of genuine claim data and the rest 923 rows consist of fraudulent claims. So the number of genuine claims is almost 15 times more than the number of fraudulent claims. So the number of fraudulent claims is negligible compared to genuine claims. This creates a class imbalance, which will lead to a biased prediction model. In order to tackle this problem, data balancing is required.

4.1 System Architecture

The system architecture is given in Figure 1. Each block is described in this Section.

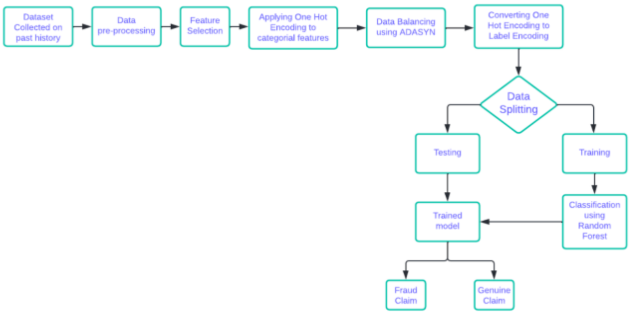

Dataset Collected on past history
Data pre-processing
Feature Selection
Applying One Hot Encoding to categorial features
Data Balancing using ADASYN
Converting One Hot Encoding to Label Encoding
Data Splitting
Testing
Training
Trained model
Classification using Random Forest
Fraud Claim
Genuine Claim

Fig.1 Proposed system architecture

A. Data preprocessing: Here we checked our data for any missing values, redundant data, duplicates or null values so we removed those rows from the training dataset. Also, we transformed the categorical data into numeric data by using label encoding and a few columns with One Hot Encoding. Along with that, few columns consist of a range value like 10-20 so we replaced these values with mean of their extreme values. We also maintained a dictionary to get back categorical value from label encoded value.

B. Feature Selection: Based on the literature survey we made we have selected an important column from the dataset. We also removed some of the unwanted columns like Policy number which consists of random values for each insurance claim and does not affect policy claims.

C. Applying One Hot encoding: One hot encoding is one of the techniques to represent a categorical feature. Here we set a new binary variable for each unique value in a categorical feature. It is one of the most preferred techniques when it comes to training categorical data. But its disadvantage is that the number of columns is equal to the number of unique values in the column of the categorical dataset. We have used One Hot encoding because directly using ADASYN created undesirable values for some attributes for e.g., Age is a whole number value which was a fractional value in the dataset generated by ADASYN.

D. Data Balancing using ADASYN: To train a model for such classification where the number element in one class is less than the other class in such a situation our model will make biased predictions where we see our model to be more tend towards the majority class. So, ADASYN is one of the data balancing techniques which tries to increase the number of minority class samples.

E. One hot encoding to Label Encoding: Once we generated random samples in one hot encoding but the issue here is that the number of columns increased from 33 to 105 which is a very tedious task to handle such a huge data column. So we have converted it back to categorical data so we will get a balanced dataset with valid inputs for the model. But again we cannot provide categorical data to train the model so we converted these categorical features to numeric labels.

F. Data Splitting: For training the model we will split the dataset in two parts for example 25% of data for testing and remaining 75% of data for training.

G. Training: We have used a few machine learning algorithms, which trains the data set aside for training the dataset into a model which will classify any new input case as fraud or not fraud. The algorithms which we used are Support Vector Machine (SVM), Naive Bayes, AdaBoost and Random Forest.

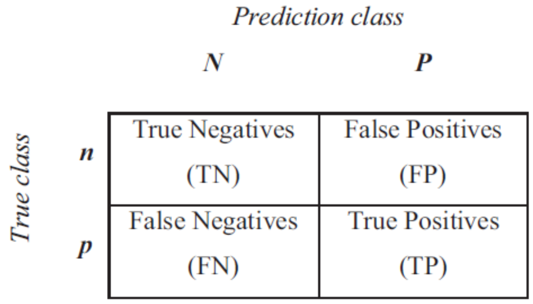
H. Testing: Remaining of splitted data is used for testing the model. Output of this will help us to evaluate our model using different evaluation metrics.

I. Trained Model: Once we are done with Testing our model. So, we finally create our model and test with different train-test split ratio and different randomness in our dataset. And we find the best configuration model for our problem statement. Now this model is ready to give us a prediction whether a new insurance claim is fraud or genuine.

5. Performance Analysis

A. Evaluation Criteria:

There are different evaluation metrics for evaluation of our model, few of the popular metrics are accuracy, precision and recall. These metrics are calculated using a confusion matrix which is prepared in the Testing phase of our project. A confusion matrix consists of 4 different values: True positive, True Negative, False Positive and False Negative. These are calculated as the number of cases classified as genuine and they are actually genuine, these claims are called as True Positive. Similarly, if a claim is fraudulent and it is classified as fraudulent then it is called as true negative. These are the two values which show both positive and negative classes which are correctly classified.



		Prediction class	
		N	P
True class	n	True Negatives (TN)	False Positives (FP)
	p	False Negatives (FN)	True Positives (TP)

Fig.2 Confusion matrix

Based on the above matrix one can evaluate his model by finding different values such as accuracy, precision and recall as

$$\textit{Recall} = \frac{TP}{TP + FN}$$

Fig.3 Formula for Recall

The above equation can be explained by saying that, from all the positive classes, what percentage correct we predicted.

$$\textit{Precision} = \frac{TP}{TP + FP}$$

Fig.4 Formula for Precision

The above formula can be explained by showing how many of all the classes that were predicted to be positive are actually positive.

Accuracy= TP+TN/(TP+TN+FP+FN)

Accuracy is calculated as a percentage of how many entries we correctly classified as correct to the total number of entries.

B. Experimental Results

After oversampling the minority class in the dataset using ADASYN the number of rows were increased to 28,628 out of which 14410 are of fraud claims and 14208 are genuine claims.

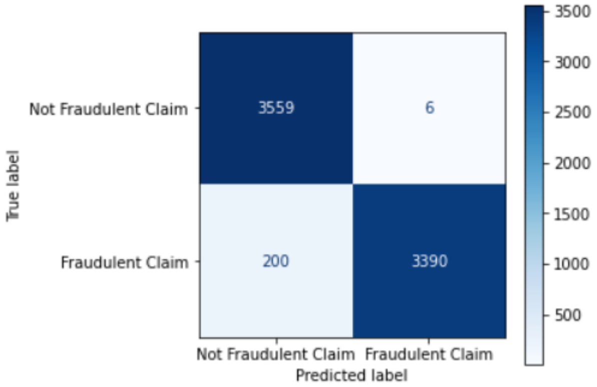


Fig.6 Confusion matrix for Random Forest with ADASYN

For testing our model, we gave 7155 rows which is 25% of the total balanced dataset. Out of these we got True Positive: 3390, True Negatives: 3559, False Positives: 6 these are the claims which are classified as fraud but labeled as genuine False Negatives: 200 these are the claims which are classified as genuine but they are labeled as fraud.

Table 1: Comparison of various classifiers on balanced dataset

Performance Metrics (in %)	Support Vector Machine (SVM)	Naive Bayes	AdaBoost	Random Forest
Accuracy	62.4	88.5	95.8	97.1
Sensitivity (or recall value)	84.5	91.1	92.7	94.4
Precision	58.7	86.5	98	99.8

In the above table results of various classifiers are given. Random Forest has performed better than other classifiers in all three metrics Accuracy, Sensitivity and Precision. SVM did not perform much well for this dataset. AdaBoost and Naive Bayes gave pretty good accuracy but not better than random forest.

ACKNOWLEDGMENT

We would like to extend our deepest gratitude to our Project Guide, Prof. Sangeetha Selvan, who guided us and provided us with her valuable knowledge and suggestions on this project and helped us improve our project beyond our limits.

We would also like to express our heartfelt thanks to our Head of Department, Dr. Sharvari S. Govilkar, for providing us with a platform where we can try to work on developing projects and demonstrate the practical applications of our academic curriculum.

We would like to express our gratitude to our Principal, Dr. Sandeep Joshi, who gave us a golden opportunity to do this wonderful project on the topic of ‘Automobile Insurance Claim Fraud Detection using Random Forest and ADASYN’, which has also helped us in doing a lot of research and learning their implementation.

REFERENCES

[1] S. Harjai, S. K. Khatri and G. Singh, "Detecting Fraudulent Insurance Claims Using Random Forests and Synthetic Minority Oversampling Technique," 2019 4th International Conference on Information Systems and Computer Networks (ISCON), 2019, pp. 123-128, doi: 10.1109/ISCON47742.2019.9036162.

[2] B. Itri, Y. Mohamed, Q. Mohammed and B. Omar, "Performance comparative study of machine learning algorithms for automobile insurance fraud detection," 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), 2019, pp. 1-4, doi: 10.1109/ICDS47004.2019.8942277.

[3] C. Muranda, A. Ali and T. Shongwe, "Detecting Fraudulent Motor Insurance Claims Using Support Vector Machines with Adaptive Synthetic Sampling Method," 2020 61st International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), 2020, pp. 1-5, doi: 10.1109/ITMS51158.2020.9259322.

[4] I. M. Nur Prasasti, A. Dhini and E. Laoh, "Automobile Insurance Fraud Detection using Supervised Classifiers," 2020 International Workshop on Big Data and Information Security (IWBIS), 2020, pp. 47-52, doi: 10.1109/IWBIS50925.2020.9255426.

[5] Y. Wei, Y. Qi, Q. Ma, Z. Liu, C. Shen and C. Fang, "Fraud Detection by Machine Learning," 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2020, pp. 101-115, doi: 10.1109/MLBDBI51377.2020.00025.

[6] Jwilda, “Classifying Fraud by Decision Trees”, Kaggle, Available: https://www.kaggle.com/code/jwilda3/classifying-fraud-by-decision-trees/data , [Accessed 30-Sept-2021]

BIOGRAPHIES

Dhruvang Gondalia

Omkar Gurav

Ameya Joshi

Aniruddha Joshi